Universidade de Brasília (UnB)
Faculdade de Economia, Administração, Ciências Contábeis e Gestão de Políticas Públicas
Departamento de Administração
Curso de Graduação em Administração

Genilson Salustiano da Silva

Prevenção de irregularidades em contratos públicos: uma análise utilizando técnicas de *machine learning*

Brasília - DF
2022

Genilson Salustiano da Silva

Prevenção de irregularidades em contratos públicos: uma análise utilizando técnicas de *machine learning*

Trabalho de conclusão de curso apresentado ao Departamento de Administração da Universidade de Brasília, como requisito parcial para a obtenção do título de Bacharel em Administração.

Orientador: Prof. Tit. Herbert Kimura

Brasília - DF

2022

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

S586p

Silva, Genilson Salustiano da
Prevenção de irregularidades em contratos públicos: uma análise utilizando técnicas de Machine Learning / Genilson Salustiano da Silva; orientador Herbert Kimura. -- Brasília, 2022.
90 p.

Monografia (Graduação - Bacharelado em Administração) -- Universidade de Brasília, 2022.

1. Inadimplência (Finanças). 2. Fraude. 3. Inteligência artificial. 4. Administração pública. I. Kimura, Herbert, orient. II. Título.

Genilson Salustiano da Silva

Prevenção de irregularidades em contratos públicos: uma análise utilizando técnicas de *machine learning*

Trabalho de conclusão de curso apresentado ao Departamento de Administração da Universidade de Brasília, como requisito parcial para a obtenção do título de Bacharel em Administração.

Data da aprovação: 25/02/2022

Herbert Kimura - Orientador
Doutor em Administração
Professor Titular do Departamento de Administração da UnB

Vinicius Amorim Sobreiro - Membro 1
Doutor em Engenharia de Produção Mecânica
Professor Associado do Departamento de Administração da UnB

Olinda Maria Gomes Lesses - Membro 2
Mestre em Ciências Empresariais
Coordenadora do Curso de Graduação em Administração da UnB

A Deus pelo o dom da vida e depois, não por acaso, aos meus pais, que, com exemplos, até hoje, me ensinam valores fundamentais, dentre os quais: honestidade, integridade, responsabilidade e respeito.

Agradecimentos

Agradeço à minha família, minha referência, em especial, aos meus pais, Miguel Salustiano da Silva e Maria Pinheiro da Silva, e à minha irmã, Eliane, pelo amor e apoio à realização deste trabalho;

Agradeço imensamente ao Prof. Dr. Herbert Kimura por ter aceitado o desafio de me orientar neste trabalho, pela paciência, objetividade, dedicação, recomendações, em suma, por ter me orientado de maneira séria e comprometida;

Agradeço à equipe do Prof. Kimura, em especial, ao Cristiano e ao César, pela colaboração com o pré-processamento e tratamento dos dados;

Agradeço à Universidade de Brasília (UnB) pelo seu comprometimento com a construção de uma sociedade mais justa, democrática e inclusiva, oferecendo uma formação acadêmica de qualidade para alunos de diferentes realidades socioeconômicas;

Agradeço à minha companheira, Sandrina, e à minha filha postiça, Júlia (minha enteada), pela companhia, carinho e por entenderem o motivo de minha ausência durante a realização deste trabalho;

Agradeço a todos os amigos, que sempre estiveram presentes nos momentos difíceis, dentre outros tantos que, de alguma forma, contribuíram para que este trabalho fosse possível;

Em síntese, sou muito grato pela experiência motivadora, inspiradora e instigante obtida durante a orientação deste trabalho. O meu orientador sempre esteve disponível para sanar as dúvidas eventuais surgidas durante todo o processo de orientação. Dito isto, assumo inteira responsabilidade pelos erros que ainda persistem neste trabalho.

"O aprendizado de máquina nos permite construir soluções de software que excedem a compreensão humana e nos mostra como a inteligência artificial pode inervar todos os setores."

(JURVETSON, 2016, n.p.)

Resumo

Técnicas de *machine learning* (aprendizado de máquina), embora largamente empregadas como suporte à gestão de risco de crédito, em instituições financeiras e em empresas de seguro, suas aplicações vêm se tornando cada vez mais bem-sucedidas no setor público. O objetivo deste trabalho foi demostrar como a inteligência artificial, por meio de algoritmos de *machine learning*, pode ser útil na redução da inadimplência nas aquisições do governo federal brasileiro e na prevençao de irregularidades em contratos públicos. Diferentes técnicas de classificação, atualmente disponíveis, foram aplicadas, dentre as quais, técnicas tradicionais de análise estatística (*discriminant analysis - DA* e *logistic regression - LR*) e técnicas de *machine learning* (*k-nearest neighbors - KNN*, *bayesian networks - BN*, *support vector machine - SVM*, *decision trees - DT*, *random forests - RF*, *bagging - BG*, *gradient boosting - GB* e *neural networks - NN*). Os métodos tradicionais, *DA* e *LR*, foram utilizados como parâmetro comparativo na análise de desempenho de cada algoritmo testado. Na amostra de teste, os métodos *ensemble learning* (aprendizagem de conjunto) *RF*, *GB* e *BG* apresentaram os melhores resultados, com taxas de erro tipo II de 4,42%, 5,65% e 6,64% e acurácia de 90,95%, 90,21% e 90,82%, respectivamente. Os resultados mostraram que as técnicas, *ensemble*, avaliadas são importantes ferramentas de apoio à decisão que podem ser empregadas na priorização da atividade, fiscalização de contratos, sendo valiosas na alocação adequada de fiscais de contratos e na prevenção de irregularidades e ilegalidades em contratos públicos.

Palavras-chave: contratos públicos; inadimplência; detecção de fraudes; tomada de decisão; inteligência artificial.

Abstract

Machine learning techniques, although widely used to support credit risk management, in financial institutions and insurance companies, their applications are becoming increasingly successful in the public sector. The objective of this work was to demonstrate how artificial intelligence, through machine learning algorithms, can be useful in reducing defaults in Brazilian federal government acquisitions and in preventing irregularities in public contracts. Different classification techniques, currently available, were applied, among which, traditional techniques of statistical analysis (discriminant analysis - DA and logistic regression - LR) and machine learning techniques (k-nearest neighbors - KNN, bayesian networks - BN, support vector machine - SVM, decision trees - DT, random forests - RF, bagging - BG, gradient boosting - GB and neural networks - NN). The traditional methods, DA and LR, were used as a comparative parameter in the performance analysis of each tested algorithm. In the test sample, the *RF*, *GB* and *BG* ensemble learning methods showed the best results, with type II error rates of 4.42%, 5.65% and 6.64% and accuracy of 90.95%, 90.21% and 90.82%, respectively. The results showed that the techniques, ensemble, evaluated are important decision support tools that can be used in the prioritization of the activity, inspection of contracts, being valuable in the adequate allocation of contract inspectors and in the prevention of irregularities and illegalities in public contracts.

keyword: public contracts; default; fraud detection; decision making; artificial intelligence.

Lista de Figuras

1 Etapas básicas de um processo *KDD*. 16

2 Ciência da computação e sua relação com *big data*. 17

3 Fluxo de trabalho de um sistema de aprendizagem supervisionada. 19

4 Fases do processo de contratação pública. 28

5 Distribuição do *status* de risco dos fornecedores no conjunto de dados. 42

6 Estrutura de priorização de fiscalização de contrato (EPFC). 49

7 Gráficos *ROC* com possíveis desempenhos de um classificador. 52

8 Gráficos *ROC* com o desempenho dos modelos analisados. 56

9 Performance dos classificadores *RF*, *GB*, *BG*, *DT*, *DA* e *KNN*. 59

10 Curvas *ROC* e erros de classificação dos modelos *DA* e *RF*. 61

Lista de Tabelas

1 Classificação para fraudes financeiras com base no *FBI*. 33

2 Emprego da análise de dados na prevenção de fraudes em contratos públicos. . 40

3 Composição do conjunto de dados utilizado nos modelos de classificação. . . . 43

4 Variáveis do conjunto de dados utilizadas nos modelos. 44

5 Matriz de confusão. 50

6 Métricas utilizadas na avaliação dos modelos. 55

7 Desempenho dos modelos nas amostras de treino e de teste. 57

8 Taxas de erro tipo I e tipo II dos modelos *GB*, *BG*, *DA* e *KNN*. 59

Lista de Abreviaturas e Siglas

AUC	*Area under the ROC curve*
BG	*Bagging*
BN	*Bayesian Network*
CEIS	Cadastro de Empresas Inidôneas e Suspensas
CGU	Controladoria-Geral da União
CNAE	Classificação Nacional de Atividades Econômicas
CNAE_Sec	Classificação Nacional de Atividades Econômicas Secundárias
ComprasNet	Portal de Compras do Governo Federal
DA	*Discriminant Analysis*
DISD	*Data Intensive Scientific Discovery*
DT	*Decision Tree*
FBI	*Federal Bureau of Investigation*
GB	*Gradient Boosting*
IBM	*International Business Machines Corporation*
KDD	*Knowledge Discovery in Database*
KNN	*K-Nearest Neighbors*
LASSO	*Least Absolute Shrinkage and Selection Operator*
LR	*Logistic Regression*
NN	*Neural Network*
RAIS	Relação Anual de Informações Sociais
RF	*Random Forest*
RFB	Receita Federal do Brasil
ROC	*Receiver Operating Characteristic*
SIAFI	Sistema Integrado de Administração Financeira
SIASG	Sistema Integrado de Administração de Serviços Gerais
SICONV	Sistema de Convênios
SVM	*Support Vector Machine*
TSE	Tribunal Superior Eleitoral
UnB	Universidade de Brasília

Sumário

1 Introdução **12**

2 Referencial Teórico **15**

2.1 *KDD*, *data mining*, *machine learning* e problemas de *big data* 15

2.1.1 Mineração de dados e *KDD* 15

2.1.2 *Machine learning* 16

2.1.3 Problemas de *big data* 20

2.2 Inadimplemento contratual em compras do setor público 21

2.2.1 Risco de crédito 22

2.2.2 Contratação pública 26

2.3 Emprego da análise de dados na prevenção de fraudes 32

2.3.1 Prevenção de fraudes financeiras 32

2.3.2 Prevenção de fraudes em contratos públicos 35

3 Método **41**

3.1 Dados 41

3.1.1 Coleta de dados 42

3.1.2 Balanceamento artificial de dados 43

3.1.3 Variáveis dependentes e independentes 43

3.2 Técnicas utilizadas 45

3.3 Estrutura de priorização baseada em riscos 48

3.4 Procedimentos de avaliação e validação 49

3.4.1 Matriz de confusão 50

3.4.2 Curva *ROC* e área sob a curva *ROC* 52

3.4.3 Custos de classificação incorreta associados aos erros tipo I e tipo II . . 54

4 Resultados **56**

4.1 Análise dos resultados obtidos 56

4.1.1 Custos de classificação incorreta dos modelos *GB*, *BG*, *DA* e *KNN* . . . 59

4.1.2 Análise comparativa entre os modelos *RF* e *DA* 60

4.2 Implantação 61

5 Considerações Finais **63**

Referências **71**

Apêndice A - Variávieis preditoras e componentes de riscos **73**

Apêndice B - Resultados com dados de treino (código-fonte em *python*) **74**

Apêndice C - Resultados com dados de teste (código-fonte em *python*) **82**

1 Introdução

Contratos públicos são arranjos legais, acordados mutualmente entre o setor público e o setor privado, que impõem um conjunto de termos e condições para o alcance de objetivos de interesse público. Esses instrumentos, segundo Bloomfield et al. (2019) e Brown, Potoski e Van Slyke (2010), visam principalmente à defesa jurídica dos limites das partes, a organização das atividades dentro da estrutura contratual e o delineamento das relações entre essas partes, aqui designadas setor público e setor privado, em termos dos respectivos direitos e obrigações.

Nas últimas décadas, com os programas nacionais de governo eletrônico e a consequente necessidade de mais eficiência e transparência ativa no setor público, Concha et al. (2012), Klabi, Mellouli e Rekik (2018) e Mostafa e El-Masry (2013), o processo de contratação pública passou por intensas inovações. Como exemplo, merece destaque a regulamentação do pregão eletrônico e a implantação do portal de compras do governo federal brasileiro, o *ComprasNet* (RALHA; SILVA, 2012). Contudo, apesar dos avanços, a contratação pública ainda está sujeita a resultados indesejáveis como improbidade, quebra de contrato e ineficiência contratual, Gallego, Rivero e Martınez (2020), dentre outros resultados como inadimplência, fraude e corrupção, Dávid-Barrett e Fazekas (2020), Fazekas e Kocsis (2017) e Berru et al. (2019).

Métodos de inteligência artificial têm sido utilizados na automatização de processos de auditoria, gerando respostas em tempo hábil e reduzindo tarefas manuais, na indústria e no governo, Phua et al. (2010), por isso, podem ser empregados como importantes ferramentas no gerenciamento de contratos públicos. Ao contrário dos métodos tradicionais, demorados, caros e impraticáveis na era do *big data*, a inteligência artificial, por meio de técnicas de classificação, pode ser utilizada para automatizar tarefas no processo de contratação pública e reduzir o esforço manual do auditor (gestor e fiscal de contrato) nesse ambiente rico em dados.

Ao fazer a classificação preliminar de instâncias, alocando dados em classes predefinidas, essas técnicas permitem ao auditor direcionar sua atenção investigativa para os casos relevantes (CARNEIRO et al., 2020). Portanto, identificar, de forma automática, quais fornecedores têm maior probabilidade de estar envolvidos em atividades suspeitas reduz, significativamente, gastos do setor público, permitindo que os custos, da auditoria e do acompanhamento do contrato, sejam sempre menores que o ganho potencial gerado por esses instrumentos de fiscalização.

A realização de procedimentos contínuos e permanentes de gerenciamento de riscos e

controle preventivo no processo de contratação pública, Brasil (2021), previnem irregularidades devastadoras, dentre as quais, consequências econômicas oriundas da corrupção (MARTINEZ-PLUMED et al., 2018). Diversos tipos de crimes como os desvendados na *operação lava jato*, por exemplo, podem ser evitados, não permitindo o desvio de recursos públicos, reduzindo o risco de intenções de investimento privado em diversas áreas e viabilizando decisões polícias importantes (LIMA; DELEN, 2020).

Nesse contexto, a proposta apresentada neste trabalho pode ser usada no setor público para monitorar a propensão de fornecedores à inadimplência e à irregularidade contratual. Determinar quais fornecedores têm maior probabilidade de inadimplir ou de estar envolvidos em atividade irregular ou ilegal previamente à fase gestão de contrato, etapa final do processo de contratação pública, é essencial, pois o gerenciamento de contrato é realizado por mão de obra especializada, muitas vezes, escassa e cara, que não consegue fiscalizar, com eficiência, todos os contratos públicos.

No presente estudo, sobre prevenção de irregularidades em contratos públicos, dados de fornecedores do governo federal brasileiro foram reunidos e analisados com o propósito de responder as seguintes questões de pesquisa: como a inteligência artificial pode ser utilizada para reduzir a inadimplência nas aquisições do governo federal brasileiro e previnir irregularidades em contratos públicos? Quais algoritmos e métodos de análise de dados podem ser utilizados?

Para responder a estas perguntas, uma estrutura de priorização de fiscalização de contrato, baseada no risco de inadimplência corporativa, foi desenvolvida e aplicada no processo de contratação pública. Trata-se de um método de classificação, utilizado na triagem de fornecedores do governo federal, que tem como objetivo a identificação de empresas com risco elevado de inadimplência e a otimização da fiscalização de contratos públicos. A referida estrutura pode ser útil na prevenção de irregularidades e pode possibilitar maior eficiência e eficácia na execução dos contratos públicos.

Portanto, o objetivo principal desta pesquisa é demostrar como a inteligência artificial, por meio de algoritmo de *machine learning*, pode ser útil na redução da inadimplência nas aquisições do governo federal brasileiro e prevenir irregularidades em contratos públicos.

Esse objetivo foi dividido nos seguintes objetivos específicos: definir e fundamentar o método mais adequado ao problema de pesquisa; investigar as irregularidades contratuais e apresentar possíveis estratégias de prevenção baseadas em técnicas de análise de dados; sele-

cionar, analisar e estruturar os dados; avaliar diferentes técnicas estatísticas de classificação, apresentar uma estrutura de priorização de fiscalização de contrato (EPFC) e empregar diferentes métricas de validação de modelos; e implementar os diferentes modelos de classificação, gerados na etapa anterior, na *EPFC* e realizar a comparação dos resultados.

Utilizando método quantitativo, este trabalho empregou técnicas estatísticas de classificação na triagem de fornecedores do governo federal brasileiro. O estudo explorou um problema em que a variável dependente, binária, *Y* (*status*), expressa a situação de risco em que o fornecedor se encontra (*adimplente*, $\{0\}$; ou *inadimplente*, $\{1\}$) e as variáveis independentes, X_1, X_2,..., X_n, expressam características do fornecedor, relacionadas à credibilidade e à reputação corporativa. Diferentes técnicas de classificação, atualmente disponíveis, foram aplicadas (Seção 3.2), dentre as quais, técnicas tradicionais de análise estatística e técnicas de *machine learning*.

O restante deste trabalho está estruturado da seguinte forma: Seção 2, revisão da literatura fundamentando o método, análise da inadimplência em contratos públicos, com base na qualidade de crédito e na reputação do fornecedor, e apresentação de casos e soluções baseadas em dados; Seção 3, método particularizando a amostra de dados, as técnicas, a estrutura de priorização de fiscalização de contrato (EPFC) e as métricas de validação de modelos empregadas; Seção 4, apresentação dos resultados obtidos, análise comparativa de desempenho e implementação da *EPFC*; e Seção 5, contendo as considerações finais.

2 Referencial Teórico

A fundamentação teórica utilizada neste trabalho de pesquisa tem como base os assuntos, descritos nas seções a seguir: 2.1 definições e particularidades da atividade de descoberta de conhecimento em base de dados (KDD), *data mining*, *machine learning* e problemas de *big data*; 2.2 inadimplemento contratual em compras do setor público; e 2.3 emprego da análise de dados na prevenção de fraudes.

2.1 *KDD*, *data mining*, *machine learning* e problemas de *big data*

Nesta seção, cabe destacar de início a diferença existente entre *knowledge discovery in databases* - *KDD* (descoberta de conhecimento em bases de dados), *data mining* (mineração de dados) e *machine learning* (aprendizado de máquina), muitas vezes, vistos na literatura como abordagens equivalentes.

2.1.1 Mineração de dados e *KDD*

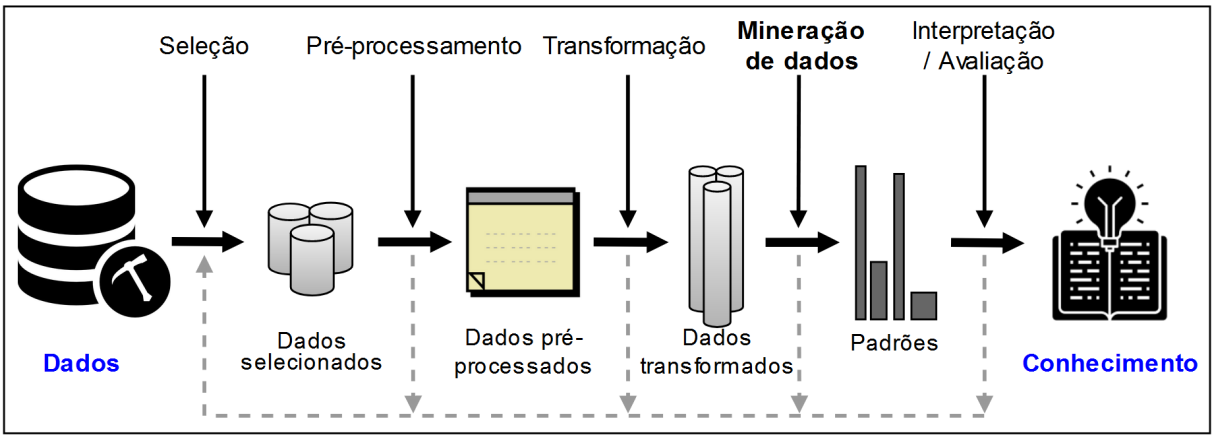
Segundo Fayyad, Piatetsky-Shapiro e Smyth (1996), mineração de dados é uma etapa do processo *KDD* e tem como objetivo a análise, preditiva e descritiva, de dados a partir da utilização de algoritmos específicos para a extração de padrões nos dados em análise. Já o termo *KDD*, diz respeito ao processo geral de descoberta de conhecimento, no qual a etapa mineração de dados faz parte. Em outras palavras, de acordo com Fayyad, Piatetsky-Shapiro e Smyth (op. cit.), o termo "*KDD* refere-se ao processo global de descobrir conhecimento útil a partir de dados e mineração de dados refere-se a um determinado passo nesse processo".

Para Kum, Duncan e Stewart (2009), *KDD* é um ramo da ciência da computação que tenta produzir uma abordagem integrada para extrair informações importantes de dados, compatibilizando conhecimentos de bancos de dados, *machine learning*, inteligência artificial, sistemas baseados em conhecimento, recuperação de informações, estatística, reconhecimento de padrões e visualização.

Conforme demonstrado na Figura 1, o processo *KDD* é composto por múltiplas etapas (seleção, pré-processamento, transformação, mineração de dados e interpretação/avaliação) e, destas, a tarefa mineração de dados é a que mais se destaca, pois é nesta etapa onde os algoritmos de *machine learning* identificam padrões nos dados em análise. A definição mais aceita de *KDD* é dada por Fayyad, Piatetsky-Shapiro e Smyth (1996, p. 40-41), isto é, "processo não trivial

que identifica padrões válidos, novos, potencialmente úteis e, finalmente, compreensíveis nos dados".

Figura 1: Etapas básicas de um processo *KDD*.



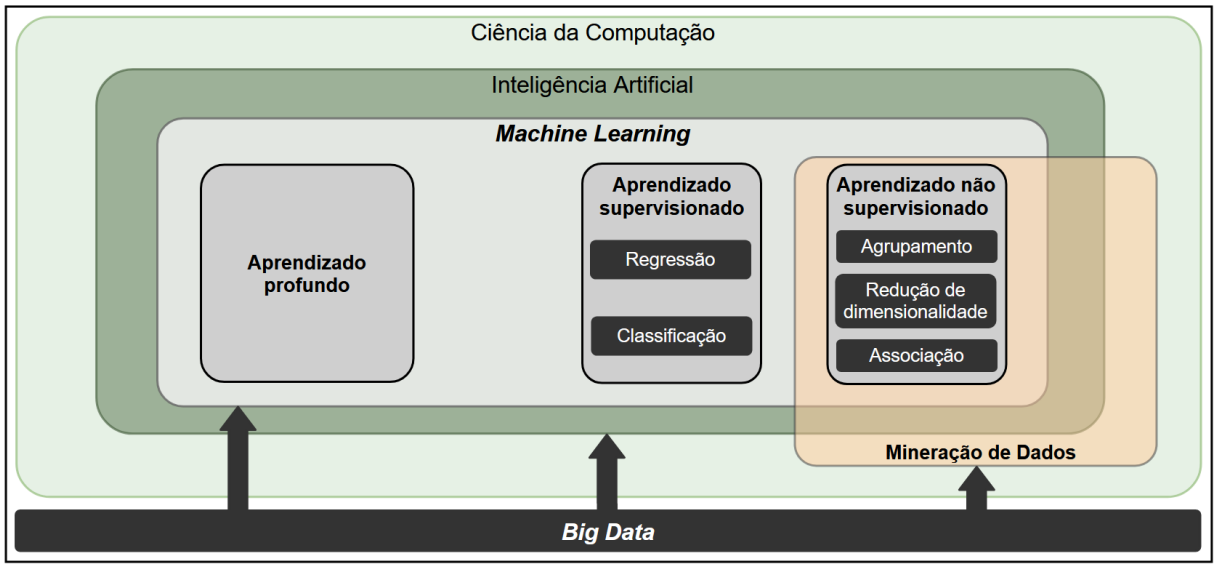
Fonte: Modificado de Fayyad, Piatetsky-Shapiro e Smyth (1996, p. 5).

Segundo Fayyad, Piatetsky-Shapiro e Smyth (1996), a componente mineração de dados depende muito de técnicas conhecidas de *machine learning*, reconhecimento de padrões e estatística para descobrir padrões nos dados do processo *KDD*. Os autores esclarecem, ainda, as diferenças entre, *KDD*, mineração de dados e *machine learning* quando dizem que vários métodos de *machine learning* são utilizados na etapa mineração de dados do processo *KDD*. Embora exista uma variedade imensa de técnicas de mineração de dados, Fayyad, Piatetsky-Shapiro e Smyth (op. cit.) dão destaque ao seguinte subconjunto de técnicas: árvore e regras de decisão; regressão não linear e métodos de classificação; métodos baseados em exemplo; modelos gráficos probabilísticos de dependência; e modelos de aprendizagem relacional.

2.1.2 *Machine learning*

De acordo com Chen e Zhang (2014) e Jordan e Mitchell (2015), *machine learning* é uma subárea da inteligência artificial que tem como objetivo a construção de sistemas computacionais que se adaptam e melhoram automaticamente com a experiência deles. De uma maneira mais formal, diz-se que "um computador aprende com a experiência E com relação a alguma classe de tarefas T e com a performance P, se sua performance nas tarefas em T, medido por P, melhorar com a experiência E" (MITCHELL, 1997 apud HEGDE; ROKSETH, 2020, p. 1). Como mostrado na Figura 2, o campo de estudo de *machine learning* está inter-relacionado aos campos de pesquisas aprendizado profundo, mineração de dados, inteligência artificial e ciência da computação.

Figura 2: Ciência da computação e sua relação com *big data*.



Fonte: Modificado de Vougas et al. (2019, p. 5).

De acordo com Boutaba et al. (2018), a mineração de dados possibilita a análise crítica e meticulosa dos dados em grandes bases de dados, contudo, quando o objetivo é a predição de eventos futuros ou sequência de eventos, técnicas de *machine learning* retornam resultados mais relevantes. As características mais nítidas dos métodos de *machine learning* são a descoberta de conhecimento e a tomada de decisão, de forma automática, inteligente e baseado em evidências, Chen e Zhang (2014) e Jordan e Mitchell (2015).

Para Castro Vieira et al. (2019), os métodos de *machine learning* são avanços importantes que podem ser empregados em inúmeros campos de pesquisa, como medicina, engenharia e computação, por exemplo: Kavakiotis et al. (2017), Vougas et al. (2019), Hegde e Rokseth (2020) e Chen e Zhang (2014). Na área da medicina, Kavakiotis et al. (2017) revisaram a literatura de aplicações de técnicas de *machine learning* na pesquisa do diabetes. Dessas aplicações, 85% foram realizadas com métodos supervisionados; e 15%, com métodos não supervisionados.

Na engenharia, Hegde e Rokseth (2020) analisaram 124 artigos sobre aplicações de *machine learning* para auxiliar na avaliação de riscos de engenharia. Os resultados do estudo mostram que das dezessete indústrias, diferentes, que aplicam técnicas de *machine learning* na avaliação de riscos, duas, a automotiva e a de construção, possuem o maior número de aplicações de *machine learning*. Segundo esses autores, as redes neurais artificiais são os modelos computacionais mais utilizados nesse tipo de aplicação.

Outra aplicação prática é na área de negócios. Segundo Chen e Zhang (2014), técnicas de *machine learning* são usadas para encontrar padrões em enormes volumes de dados de varejistas, como *Wal-Mart* e *Taobao* (*website* de compras *on-line* chinês, com sede em Hangzhou na China e de propriedade da *Alibaba*), gerando melhor eficiência, nas suas estratégias de preços e campanhas publicitárias, possibilitando maior competitividade nos negócios.

Ainda de acordo com Chen e Zhang (2014), muitos projetos (por exemplo, o *Apache Mahout*) de gigantes como *Google*, *Amazon*, *Yahoo!*, *IBM*, *Twitter* e *Facebook*, envolvem problemas de *big data* e, com isso, necessitam de métodos de *machine learning scalable* (aprendizado de máquina escalável, ou seja, de grande escala). O *Apache Mahout* é, segundo esses autores, um projeto da empresa americana *Apache Software Foundation*, que tem como objetivo o fornecimento de técnicas de *machine learning scalable* para aplicativos de análise de dados inteligentes.

Nas últimas décadas, com o aumento da disponibilidade de grandes bancos de dados e com a elevada capacidade computacional disponível e de baixo custo, Ralha e Silva (2012) e Rozario e Issa (2020), novos métodos de inteligência artificial, com base em dados, foram criados (HANDAN-NADER; HO; LIU, 2021), (JORDAN; MITCHELL, 2015), (KUZIEMSKI; MISURACA, 2020) e (SOUSA et al., 2019). A literatura, por exemplo, Berru et al. (2019), Boutaba et al. (2018), Jordan e Mitchell (2015), Kavakiotis et al. (2017) e Liu et al. (2021), classifica esses métodos, do tipo computacional, em três categorias principais: aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço.

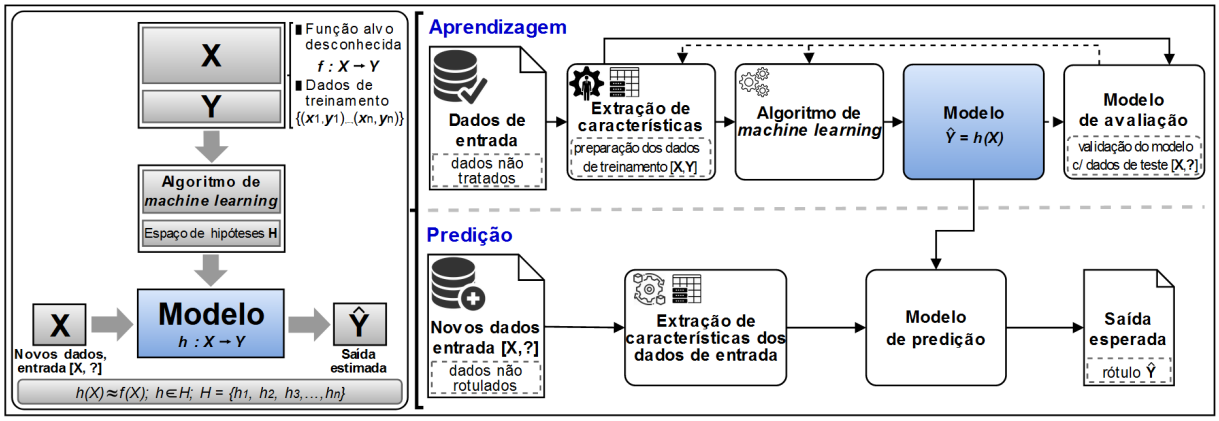
De acordo com Boutaba et al. (2018), Kavakiotis et al. (2017) e Liu et al. (2021), esses métodos diferenciam-se nos seguintes aspectos: *i) supervisionado*, o sistema de aprendizagem deduz uma função que mapeia as variáveis de entrada (x) para uma variável de saída (y), com base em dados de treinamento rotulados; *ii) não supervisionado*, o sistema de aprendizagem usa dados de treinamento não rotulados para criar modelos que tenta descobrir a estrutura oculta nos dados ou associações entre variáveis; e *iii) reforço*, o sistema aprende por meio da interação direta com o ambiente a fim de maximizar alguma noção de recompensa cumulativa, isto é, o sistema aprende por meio de tentativa e erro, com base em *feedback*.

Os modelos gerados por meio da aprendizagem supervisionada, abordagem utilizada nesta pesquisa, têm como objetivo a análise de dados novos não rotulados, normalmente, em problemas de classificação ou em problemas de regressão (BOUTABA et al., 2018), (LIU et al.,

2021) e (AZODI; TANG; SHIU, 2020). Neste tipo de método, de acordo com Jordan e Mitchell (2015), para cada variável de entrada (x), também chamada de variáviel independente ou atributo, há sempre uma variável de saída (y) correspondente, chamada de variável dependente, atributo-alvo ou rótulo (rótulos categóricos, problema de classificação; e rótulos contínuos, problema de regressão).

Segundo Boutaba et al. (2018), Kamath, Liu e Whitaker (2019) e Liu et al. (2021), neste tipo de aprendizagem, os algoritmos são utilizados para aprender uma função $f : X \rightarrow Y$ de um determinado conjunto de exemplos, representado linearmente por $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, onde $x_i \in X$ é um vetor de característica e $y_i \in Y$ é um rótulo conhecido de x_i. Em outras palavras, dado um conjunto de exemplos $X = \{x_1, x_2, x_3, ..., x_n\}$ e um conjunto de rótulos $Y = \{y_1, y_2, y_3, ..., y_n\}$, o objetivo do algoritmo é encontrar o mapeamento $h : X \rightarrow Y$, que possa ser empregado para classificar dados novos, ainda não rotulados (BOUTABA et al., 2018), (JORDAN; MITCHELL, 2015) e (KAMATH; LIU; WHITAKER, 2019). O fluxo de tabalho de um sistema de aprendizagem supervisionada é ilustrado na Figura 3.

Figura 3: Fluxo de trabalho de um sistema de aprendizagem supervisionada.



Fonte: Modificado de Boutaba et al. (2018, p. 6).

Conforme apresentado na Figura 3, após o pré-processamento dos dados, um conjunto de dados de treinamento, expresso por $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ e representado por [X, Y], é utilizado para treinar um algoritmo de *machine learning*. O passo seguinte é encontrar no espaço de hipóteses (H) uma função h(x) que mais se aproxima à função-alvo desconhecida f(x), isto é, h(x) $\approx f$(x). Utilizando como base a hipótese $h : X \rightarrow Y$, obtida na etapa anterior, novos dados de entrada não rotulados [X, ?] são empregados logo a seguir, nas etapas de avaliação e predição, obtendo como retorno os rótulos $(\hat{Y})$ correspondentes aos dados de entrada. Métricas que podem ser utilizadas na avaliação do desempenho de um algoritmo são apresentadas na

Seção 3.4.

2.1.3 Problemas de *big data*

Para Chen e Zhang (2014), um dos conceitos mais aceitos para *big data* é o da empresa de consultoria *Gartner*, empresa global com sede na cidade de Stamford nos Estados Unidos. Para a *Gartner* "*big data* são ativos de informações de alto volume, alta velocidade e/ou alta variedade que requerem novas formas de processamento para permitir tomada de decisão aprimorada, descoberta de *insights* e otimização de processos". Segundo Chen e Zhang (op. cit.), o volume de dados cresce de forma acelerada porque dispositivos móveis, dentre os quais, sensor de informações, tecnologias sensoriais aéreas, sensoriamento remoto, registros de *software*, câmeras, microfones, leitores de identificação por radiofrequência, redes de sensores sem fio, entre outros, estão sendo cada vez mais usados para a captura de dados.

Para Chen e Zhang (2014), o grande número de dados produzidos atualmente pelas mais diversas fontes como redes de sensores, telescópios, experimentos científicos e instrumentos, terminam gerando inúmeros problemas de *big data* em áreas importantes como comércio e negócios, administração da sociedade e pesquisa científica. Segundo os autores, esses problemas de *big data*, originalmente chamado de *Data Intensive Scientific Discovery - DISD* (descoberta científica intensiva em dados), surgem como consequência do enorme volume de dados, os quais ocultam valores potencialmente úteis. Segundo os autores, a necessidade de melhoria na produtividade e na eficiência dos serviços do setor público é um problema de *big data*.

O número total de dados da administração pública é muito grande, pois as pessoas demandam por serviços públicos diferentes, de acordo com suas idades. Segundo Chen e Zhang (2014), governos de todo o mundo estão enfrentando condições adversas para melhorar suas produtividades. Estados Unidos e Europa, por exemplo, constituíram diferentes programas de *big data*. De acordo com um relatório do Instituto McKinsey (2012 apud CHEN; ZHANG, 2014, p. 3), o setor público europeu pode reduzir as despesas de atividades administrativas de 15 a 20%, aumentando de 223 bilhões para 446 bilhões de valores.

Chen e Zhang (2014) afirmam que o *big data* possibilita ganhos econômicos significativos. Contudo, de acordo com Kuziemski e Misuraca (2020) e Sousa et al. (2019), diversos desafios, como inconsistência e incompletude, escalabilidade, pontualidade e segurança dos dados, bem como, as implicações éticas relacionadas à proteção de dados do uso de técnicas

avançadas de análise de dados, devem também ser levados em consideração. Segundo Chen e Zhang (op. cit.), os bancos de dados do mundo real são severamente suscetíveis a dados inconsistentes, incompletos e "barulhentos", pois, além de serem volumosos, têm como origem fontes heterogêneas (falhas na captura e no armazenamento). Chen e Zhang (2014) apontam também que, em muitas aplicações importantes de *big data*, técnicas e tecnologias de ponta não conseguem resolver idealmente os problemas reais, especialmente para análises em tempo real.

Apesar disso, o setor público segue a tendência mundial, disponibilizando dados em portais de dados abertos, promovendo transparência e responsabilidade na execução da despesa pública (RALHA; SILVA, 2012) e (ROZARIO; ISSA, 2020). Uma das possíveis soluções aos problemas de *big data* da administração pública pode ser encontrada na pesquisa de Attard et al. (2015), sobre iniciativas de dados abertos do governo.

Attard et al. (op. cit.) trazem à tona questões importantes como publicação, consumo e ciclo de vida dos dados abertos do governo, bem como, os motivos que impedem as iniciativas abertas do governo de atingirem todo o seu potencial. Sobre esses motivos, os autores apontam que a natureza heterogênea dos formatos dos dados publicados pela administração pública (por exemplo: imagens, *PDF CSV*, planilhas e arquivos *XML*) continua sendo o maior obstáculo para a exploração total de iniciativas abertas no setor público.

Para Attard et al. (2015), existem três motivos principais para se abrir os dados do setor público: *i) transparência* - com a transparência, há um desincentivo à corrupção, pois cidadãos e outras partes interessadas podem monitorar iniciativas do governo e verificar sua legitimidade; *ii) valor social e comercial* - os dados do setor público têm valor social e comercial, com isso, ao publicar esses dados, os governos incentivam os interessados a inovar e criar novos serviços; e *iii) governança participativa* - com iniciativas de dados abertos, cidadãos podem participar de processos de governança, como tomadas de decisões e formulações de políticas.

2.2 Inadimplemento contratual em compras do setor público

De acordo com Choi, Son e Kim (2018), Wang, Asghari et al. (2020) e Zhu et al. (2019), problemas como dificuldades financeiras, alto risco operacional, ambiente competitivo incerto e recursos humanos e de capital inadequados às adversidades econômicas induzem fornecedores a altos índices de inadimplência ou à falência. Diante disso, com o avanço dos métodos de análise de dados, técnicas de *machine learning* passaram a ser utilizadas em apoio à previsão

do risco de crédito corporativo, Wang e Ma (2012) e Zhu et al. (2019), tornando-se úteis na previsão do desempenho de fornecedores (CAVALCANTE et al., 2019) e (CHOI; SON; KIM, 2018).

Neste trabalho, técnicas de *machine learning* foram utilizadas para estimar a probabilidade de inadimplência de fornecedores, em contratos com o governo federal brasileiro. O inadimplemento contratual, descumprimento de parte ou da integralidade do contrato, foi estimado a partir da análise do risco de crédito do fornecedor. A fundamentação teórica, baseada na análise de crédito corporativo, e as técnicas utilizadas nesta pesquisa são apresentadas, respectivamente, nas Seções 2.2.1 e 3.2.

2.2.1 Risco de crédito

Modelos de previsão de crise financeira de empresas contratadas, utilizados por diversas instituições para estimar a probabilidade de inadimplência da contraparte, foram extensivamente estudados (CHOI; SON; KIM, 2018). Nos últimos anos, de acordo com Wang, Ma, Huang et al. (2012), Wang, Ma e Yang (2014) e Zhu et al. (2019), a avaliação do risco de crédito corporativo, sobretudo a previsão do risco de crédito das pequenas e médias empresas, tem sido uma das questões críticas do financiamento da cadeia de suprimento e uma das medidas decisivas em apoio à tomada de decisão em muitas instituições financeiras.

Segundo Zhu et al. (2019), no ano de 2019, o índice de inadimplência das grandes empresas da China era de 1,19% e o das pequenas e médias empresas era de quase 5,94%, uma diferença de aproximadamente 4,75%. Esses autores argumentam que a alta dificuldade financeira, o alto risco operacional e o ambiente competitivo incerto são os motivos mais relevantes que levam as pequenas e médias empresas a elevados índices de inadimplência.

Nesse contexto, de acordo com Loukis, Kyriakou e Maragoudakis (2020), modelos de análise de crédito corporativo são muito importantes, sobretudo, em período de crise econômica recessiva, pois auxiliam na identificação de empresas com elevada vulnerabilidade econômica ou com elevado risco de inadimplência. Com isso, esses métodos podem ser utilizados, nos setores público e privado, em processo de recuperação econômica de empresas, Loukis, Kyriakou e Maragoudakis (2020), ou, conforme proposto neste trabalho, com o objetivo de priorizar a execução de atividades de acompanhamento e fiscalização de contratos públicos, Figura 6.

Instituições financeiras, gestores de fundos, credores, governos e participantes do mer-

cado financeiro buscam desenvolver modelos para avaliar, com eficiência, a probabilidade de inadimplência da contraparte (BARBOZA; KIMURA; ALTMAN, 2017). De acordo com Kimura, Basso e Kayo (2015), a análise de crédito pode ser realizada por meio de técnicas estatísticas tradicionais de discriminação (por exemplo: análise de discriminante e regressão logística) e por meio de métodos automáticos de classificação (por exemplo: redes neurais, árvores decisão, *bagging* e *boosting*). Segundo Kimura, Basso e Kayo (op. cit.), a discriminação e a classificação têm grandes utilidades na análise de crédito, pois além de possibilitar a compreensão das características diferentes dos dados que discriminam, possibilitam também a alocação dos dados em grupos predefinidos.

No contexto de compras e contratações do setor público, em muitas ocasiões, gestores governamentais precisam decidir entre a continuidade e a interrupção de um contrato tendencioso. Casos como esses podem ser entendidos melhor quando se analisa todas as componentes do risco contratual. Na pesquisa sobre risco sistêmico em contratos públicos, Bloomfield et al. (2019) classificaram o risco contratual em quatro categorias: *i) risco de representação*, composto por risco de informação e risco relacional; *ii) risco de desempenho*, resultados intermediários e finais do contrato; *iii) risco financeiro*, transações financeiras, risco de inadimplência, remuneração e multas; e *iv) risco de contrato*, renegação de uma das partes do contrato.

No processo de contratação pública, preços elevados, apresentados por fornecedores de boa reputação, Klabi, Mellouli e Rekik (2018), podem caracterizar a entrega de produtos de ótima qualidade, por isso, é preciso avaliar, não apenas o preço do produto, mas também o custo de transação e o risco de não conformidade do fornecedor (TKACHENKO; YAKOVLEV; KUZNETSOVA, 2017). Ao contrário dos produtos complexos, em que as despesas de redação dos termos do contrato, para todos os cenários possíveis, superam os benefícios recíprocos da negociação, Brown, Potoski e Van Slyke (2010, p. i44), na contratação de produtos simples, os fatores *custo*, *qualidade* e *quantidade* podem ser facilmente definidos e avaliados (BROWN; POTOSKI; VAN SLYKE, 2010). Com isso, a análise contratual de produtos simples com técnicas de análise de dados é muito oportuna, sobretudo, para fins de fiscalização e controle de contratos vinculados a fornecedores de má reputação.

Sistemas de reputação, comuns em *websites* de comércio eletrônico como *eBay* e *Amazon*, com base em *feedback* de transações passadas, permitem a identificação de entidades confiáveis para transações futuras, Hendrikx, Bubendorfer e Chard (2015), por isso, análises fei-

tas por meio desses sistemas são utilizadas para determinar os vencedores de leilões (KLABI; MELLOULI; REKIK, 2018). Para o funcionamento desses sistemas, contribuições importantes são obtidas de modelos de pontuação de crédito (HENDRIKX; BUBENDORFER; CHARD, 2015). Estudados há décadas, Barboza, Kimura e Altman (2017) e Feng et al. (2018), os modelos de pontuação de crédito permitem que instituições financeiras classifiquem mutuários, em níveis de risco de crédito, a partir de dados históricos e de técnicas estatísticas (CHEN; MA; MA, 2009), (YAP; ONG; HUSAIN, 2011) e (ZHAO et al., 2015).

Nesta pesquisa, sobre prevenção de irregularidades em contratos públicos, um dos objetivos é antever situações de riscos, como o inadimplemento contratual de fornecedores do governo federal, que podem comprometer o alcance de metas governamentais. Para tal fim, é essencial avaliar a reputação desses fornecedores, identificando características que dão subsídios para uma maior probabilidade de os mesmos virem descumprir suas obrigações firmadas em contrato. Deste modo, permitem que os riscos sejam conhecidos antes da assinatura do contrato, antes do pagamento da despesa ou antes do fornecimento do bem e/ou da prestação do serviço.

Uma das possíveis soluções para problemas dessa natureza pode ser obtida utilizando algoritmos de reconhecimento de padrões, os quais, segundo Castro Vieira et al. (2019), podem ser usados para classificar a credibilidade da contraparte. Conforme apresentado, ao longo deste trabalho, a elaboração de um modelo de avaliação de risco de crédito é um processo composto por várias etapas (coleta e pré-processamento de dados; modelagem; validação e implantação). Na construção desses modelos, de acordo com Garcıa, Marqués e Sánchez (2012), várias estratégias são utilizadas na etapa de modelagem, incluindo técnicas estatísticas, métodos de inteligência computacional (ramo da inteligência artificial) e técnicas de pesquisa operacional.

De acordo com Feng et al. (2018), durante as últimas duas décadas, vários modelos de classificação, baseados em técnicas estatísticas tradicionais e/ou em técnicas de *machine learning*, vêm sendo utilizados em sistemas de pontuação de crédito. Nos últimos anos, principalmente após a divulgação dos Acordos de Basileia pelo Comitê de Basileia de Supervisão Bancária, o emprego de modelos de pontuação de crédito cresceu muito, tanto para decisões de concessão de crédito como também para fins de gestão de risco (LOUZADA; ARA; FERNANDES, 2016) e (MARQUÉS; GARCIA; SÁNCHEZ, 2012).

Segundo Louzada, Ara e Fernandes (2016), o objetivo principal dos modelos de pon-

tuação de crédito é identificar as características (atributos) que influenciam o pagamento ou o comportamento de não pagamento do cliente, bem como seu risco de inadimplência. Após revisarem a literatura sobre a aplicação de técnicas de classificação binária no ranqueamento de crédito, Louzada, Ara e Fernandes (op. cit.), apresentaram um estudo de simulação experimental sob nove métodos gerais: *neural network (NN)*, *support vector machine (SVM)*, *decision tree (DT)*, *logistic regression (LR)*, *fuzzy logic*, *genetic programming*, *discriminant analysis (DA)*, *bayesian network (BN)* e métodos *ensemble* (*bagging* e *boosting*).

Segundo Castro Vieira et al. (2019), *discriminant analysis (DA)* e *logistic regression (LR)* são métodos tradicionais de análise de crédito, baseados em técnicas estatísticas mais simples, usados como referência para avaliar a capacidade preditiva de outros modelos de pontuação de crédito. Castro Vieira et al. (op. cit.) dão destaque a técnicas computacionais, como *support vector machine (SVM)*, *random forest (RF)*, *bagging* e *boosting*. Esses autores defendem que métodos tradicionais, como *DA* e *LR*, quando utilizados em aplicações do tipo não linear, retornam bons desempenhos, mas quando comparados com métodos computacionais, que utilizam técnicas de *machine learning*, apresentam precisão menor.

Várias pesquisas de análise comparativa confirmaram a superioridade de desempenho das técnicas de *machine learning* sobre os métodos estatísticos tradicionais, por exemplo: Feng et al. (1993 apud CASTRO VIEIRA et al., 2019, p. 3) compararam o desempenho dos algoritmos de *machine learning*, *DT* e *NN*, com os classificadores estatísticos, *DA* e *LR*; Barboza, Kimura e Altman (2017), usando dados de empresas norte-americanas, compararam o desempenho de algoritmos de *machine learning* com a técnica estatística, *LR*; e Tsai, Hsu e Yen (2014) e Yeh, Chi e Lin (2014), em estudos sobre previsão de falência, reconheceram também a superioridade, em termos de precisão e erro de previsão, das técnicas de *machine learning* sobre os métodos estatísticos clássicos.

Em contraste com as abordagens tradicionais de pontuação de crédito, baseadas em modelos de programação estatística, os métodos de inteligência artificial, em vez de assumir conhecimento prévio específico, escrito por um programador, extraem automaticamente conhecimentos dos exemplos em análise, Jordan e Mitchell (2015) e Marqués, Garcıa e Sánchez (2012, p. 1), ou seja, enquanto na programação estatística tradicional a variável de saída, *soluções*, é obtida a partir das variáveis de entrada, *dados* e *regras* (*dados* + *regras* $\rightarrow$ *soluções*), nos métodos baseados em inteligência artificial, Figura 3, a variável de saída, *regras*, é obtida a

partir das variáveis de entrada, *dados* e *soluções* (*dados* + *soluções* $\rightarrow$ *regras*).

Diante disso, com a adoção de plataformas de compras eletrônicas no setor público, torna-se cada vez mais importante a utilização de técnicas de *machine learning* em apoio à tomada de decisão em muitos órgãos públicos, que precisam alocar recursos escassos em prioridades concorrentes. Portanto, utilizando modelos de análise de crédito, um dos objetivos nesta pesquisa é prever a inadimplência de fornecedores do governo federal brasileiro. Para atingir este objetivo, classificar, de forma automática, fornecedores como *adimplentes* ou *inadimplentes*, dez técnicas de classificação, Seção 3.2, foram avaliadas.

2.2.2 Contratação pública

A Transparência Internacional (TI), Kühn e Sherman (2014, p. 6), define compras públicas como "aquisição de bens ou serviços por um departamento governamental ou qualquer instituição de propriedade do governo". De acordo com Dávid-Barrett e Fazekas (2020, p. 2), as compras governamentais representam, em média, de 30 a 50% dos gastos públicos e são altamente propensas à corrupção. Vista como "o mau uso do poder confiado para ganho privado", Kostyo (2006, p. 14) e Kühn e Sherman (2014, p. 6), ou "abuso de cargo público para ganhos privados", Tkachenko, Yakovlev e Kuznetsova (2017, p. 2), a corrupção envolve geralmente de 10 a 25% do valor do contrato, podendo em alguns casos chegar a 50%, Berru et al. (2019) e Kostyo (2006, p. 13).

Os recursos financeiros gastos nas contratações do setor público representam parte significativa dos valores movimentados nas atividades econômicas de muitos países (ADJEI-BAMFO; MALOREH-NYAMEKYE; AHENKAN, 2019). No Brasil, segundo Delmonico et al. (2018), as aquisições de bens e/ou serviços do setor público são responsáveis por aproximadamente 15% do Produto Interno Bruto (PIB). Segundo Obwegeser e Müller (2018), os países que compõem a Organização para a Cooperação e Desenvolvimento Econômico (OCDE) gastam em média 13% de seus *PIB* em aquisições de bens e/ou serviços, o que corresponde a uma média de 29% do total das despesas governamentais.

Gastos do setor público, normalmente, estão sujeitos a regras rígidas de rastreabilidade e de responsabilidade. Por isso, geram grandes desafios aos tomadores de decisão e formuladores de políticas que desejam investir em inovação. Sobre a inovação em contratos públicos, de acordo com Obwegeser e Müller (2018), existem três aspectos importantes a serem analisados:

i) inovação no processo de compras públicas; *ii)* inovação de serviços públicos por meio de compras; e *iii)* uso de compras públicas como ferramenta para a formulação de políticas de inovação no lado da demanda.

Com relação ao papel do governo eletrônico nas compras públicas sustentáveis, Adjei-Bamfo, Maloreh-Nyamekye e Ahenkan (2019) identificaram poucos estudos sobre como a implantação de tecnologias, eletrônica e da *internet*, pode ser aproveitada para estimular o gerenciamento sustentável da cadeia de suprimento, ou ainda, sobre como o emprego da tecnologia da informação pode ser útil à execução de compras públicas sustentáveis. Segundo esses autores, a implantação dessas tecnologias contribui para a efetividade das compras públicas sustentáveis em direção a uma maior agenda de desenvolvimento sustentável.

Desde 2005, o processo de contratação pública no Brasil é realizado por meio de leilão eletrônico e todas as empresas participantes do processo têm seus lances registrados no banco de dados do *ComprasNet*[1], subsistema do *SIASG*[2] (RALHA; SILVA, 2012). De acordo com Concha et al. (2012), as compras governamentais eletrônicas estão se tornando partes essenciais dos programas nacionais de governo eletrônico, pois permitem transparência ativa e favorecem relacionamentos eficientes com fornecedores. A redução dos custos de transação e a restrição do poder discricionário de servidores públicos que supervisionam os processos de compras são, segundo Dávid-Barrett e Fazekas (2020), ganhos importantes dessa nova modalidade de compra.

Ainda sobre governo eletrônico, tendo em vista questões importantes, apontadas anteriormente, como publicação, consumo e ciclo de vida de dados, Attard et al. (2015), torna-se necessário cada vez mais a utilização de meios que possibilitem a mensuração periódica do desempenho e satisfação dos objetivos dos portais de compras do setor público. Instrumentos de avaliação da maturidade de portais, como o utilizado para avaliar a maturidade de portais de 14 países da América Latina e do Caribe, Concha et al. (2012), podem ser empregados com essa finalidade.

No Brasil, o processo para a aquisição de bens e/ou serviços do governo federal segue conforme regras previstas na Lei nº 14.133/21, Brasil (2021), chamada de Lei de Licitações e Contratos Administrativos. A Lei nº 14.133/21 detalha as modalidades de licitação existentes, as fases do processo licitatório, os tipos de contratos, a qualificação das empresas e as sanções

[1]Portal de Compras do Governo Federal (ComprasNet).
[2]Sistema de Administração de Serviços Gerais (SIASG).

administrativas e penais que podem ser aplicadas ao fornecedor no caso do seu inadimplemento contratual. De acordo com o Art. 155, da Lei 14.133/21, Brasil (2021), o fornecedor contratado será culpabilizado administrativamente pelas infrações descritas a seguir:

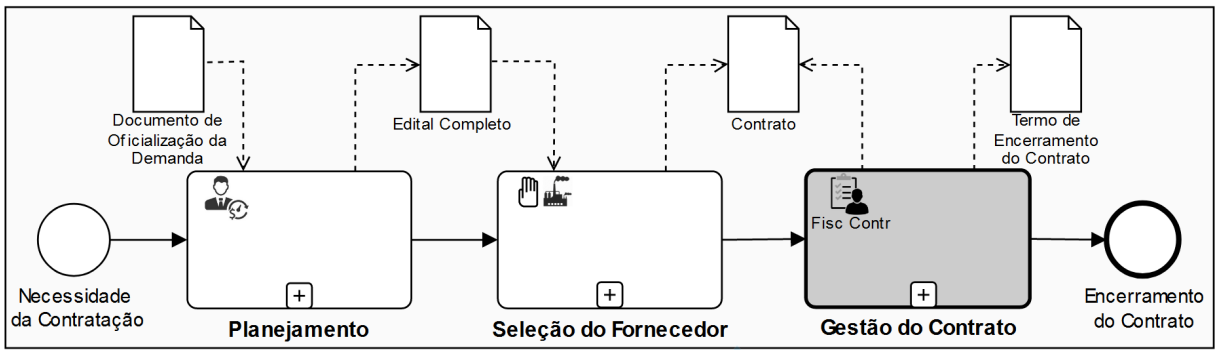
> *i)* dar causa à inexecução parcial do contrato; *ii)* dar causa à inexecução parcial do contrato que cause grave dano à Administração, ao funcionamento dos serviços públicos ou ao interesse coletivo; *iii)* dar causa à inexecução total do contrato; *iv)* deixar de entregar a documentação exigida para o certame; *v)* não manter a proposta, salvo em decorrência de fato superveniente devidamente justificado; *vi)* não celebrar o contrato ou não entregar a documentação exigida para a contratação, quando convocado dentro do prazo de validade de sua proposta; *vii)* ensejar o retardamento da execução ou da entrega do objeto da licitação sem motivo justificado; *viii)* apresentar declaração ou documentação falsa exigida para o certame ou prestar declaração falsa durante a licitação ou a execução do contrato; *ix)* fraudar a licitação ou praticar ato fraudulento na execução do contrato; *x)* comportar-se de modo inidôneo ou cometer fraude de qualquer natureza; *xi)* praticar atos ilícitos com vistas a frustrar os objetivos da licitação; e *xii)* praticar ato lesivo previsto no art. 5º da Lei nº 12.846, de 1º de agosto de 2013. (BRASIL, 2021, p. 63).

Ao cometer essas infrações, os responsáveis estarão sujeitos a sanções administrativas previstas no Art. 156, da Lei nº 14.133/21, Brasil (2021), dentre as quais: advertência; multa; impedimento de licitar e contratar; e declaração de inidoneidade para licitar ou contratar.

> *i)* advertência, [...] quando não se justificar a imposição de penalidade mais grave; *ii)* multa, [...] não poderá ser inferior a 0,5% (cinco décimo porcento) nem superior a 30% (trinta porcento) do valor do contratado licitado ou celebrado com contratação direta e será aplicada ao responsável por qualquer das infrações administrativas previstas no art. 155 desta Lei; *iii)* impedimento de licitar e contratar, [...] impedirá o responsável de licitar ou contratar no âmbito da Administração Pública direta e indireta do ente federativo que tiver aplicado a sanção, pelo prazo máximo de 3 (três) anos; *iv)* declaração de inidoneidade para licitar ou contratar, [...] impedirá o responsável de licitar ou contratar no âmbito da Administração Pública direta e indireta do ente federativo que tiver aplicado a sanção, pelo prazo mínimo de 3 (três) anos e máximo de 6 (seis) anos. (BRASIL, 2021, p. 64).

De modo geral, o processo de contratação pública no Brasil é composto por três fases, Brasil (2017, p. 16), quais sejam: Planejamento; Seleção do Fornecedor; e Gestão do Contrato. De acordo com os Art. 11, 18 e 117, da Lei nº 14.133/21, Brasil (2021, p. 2, 13 e 54), cada etapa do processo de contratação pública, inclusive a fase *Gestão do Contrato* (foco deste trabalho), tem fim específico e ocorre de forma ordenada, conforme mostrado na Figura 4.

Figura 4: Fases do processo de contratação pública.



Fonte: Modificado de Brasil (2017, p. 16).

O *Planejamento*, fase interna e preparatória, é a etapa em que se realiza a análise de via-

bilidade da contratação; se inicia com a formalização da demanda e se encerra com a publicação do edital de licitação. Com a publicação do edital de licitação, dá-se início a fase *Seleção do Fornecedor* (fase externa), cuja finalidade é a seleção da proposta mais vantajosa para a Administração Pública; se encerra com a assinatura do contrato. Por último, temos a fase *Gestão do Contrato*, na qual é realizado o acompanhamento e a fiscalização do contrato; se inicia com a nomeação da equipe, gestor e fiscal, e se encerra após o cumprimento das obrigações previstas no contrato.

A fiscalização de contrato (Fisc Contr) é tarefa crítica da fase *gestão do contrato*. Sua ineficiência pode resultar em vantagem indevida ao fornecedor, Art. 178, da Lei nº 14.133/21, Brasil (2021, p. 71-73), pois dificulta a identificação, tempestiva, de irregularidades e/ou de ilegalidades, dissimuladas, ou não, no processo de contratação pública (SANTOS, 2017). De acordo com o Art. 121, da Lei nº 14.133/21, Brasil (2021) e Santos (2017), a fiscalização de contrato inadequada pode, também, no caso de inadimplência do fornecedor, levar a Administração Pública a responder subsidiariamente por encargos trabalhistas não pagos pelo prestador de serviços.

Embora a fiscalização de contrato seja tarefa obrigatória, Art. 117, da Lei nº 14.133/21, Brasil (2021), que possibilita, de forma eficaz, efetiva e eficiente, o fiel cumprimento das cláusulas contratuais, de acordo com Santos (2017), na maioria dos casos, essa tarefa tem sido mais uma formalidade cumprida durante a execução do contrato, realizada como atividade acessória em segundo plano. Ainda, de acordo com Santos (op. cit.), não pode o agente da Administração, nomeado como fiscal de contrato, esperar o término do contrato para verificar a conformidade do objeto contratatual, pois, no momento da entrega do objeto do contrato, muitas práticas ilegais podem já se encontrar encobertas.

Conforme apresentado na Seção 2.2.1, técnicas de *machine learning* podem ser empregadas na avaliação do risco de inadimplência corporativo, sendo importantes instrumentos de suporte à gestão que possibilitam a tomada de decisão fundamentada em evidências. Deste modo, com a utilização dessas técnicas, é possível priorizar de forma automática a execução da fiscalização de contratos, vinculados a fornecedores de má reputação, que oferecem alto risco de inadimplência e, com isso, conferir mais eficiência à fase *Gestão do Contrato*. Nestes casos, a utilização de tais técnicas contribui para o emprego e alocação adequada de fiscais de contratos, sendo úteis no combate à corrupção e na prevenção de irregularidades no processo de

contratação pública.

Sobre a ineficiência e a corrupção no processo de contratação pública, a literatura mostra que essas anormalidades são distorções fortemente correlacionadas, Dal Bó e Rossi (2007), típicas de países em desenvolvimento (GALLEGO; RIVERO; MARTINEZ, 2020). No Brasil, de acordo com Santos (2017, p. 196), os danos causados por contratos ineficazes, não fiscalizados ou não acompanhados, podem ser até mais prejudiciais ao erário do que a corrupção. Por esses motivos, de acordo com Gallego, Rivero e Martınez (op. cit.), contratos públicos transparentes são cruciais para evitar resultados indesejáveis como improbidade, quebra de contrato e ineficiência contratual.

De acordo com Gallego, Rivero e Martınez (2020), em muitos casos, a corrupção e a ineficiência nos contratos públicos são geradas por ações, de servidores públicos e/ou de fornecedores, inexoravelmente rentáveis e oportunistas. Por isso, defendem que descumprimentos de cláusulas contratuais, como a não entrega do bem ou do serviço, no tempo ou pelo custo inicialmente previsto, ou a não produção do bem ou do serviço contratado, podem ser vistos como possíveis alertas de corrupção.

De acordo com a Transparência Internacional (TI), Kostyo (2006, p. 13), "poucas atividades criam tentações maiores ou oferecem mais oportunidades de corrupção do que as compras do setor público". Segundo Dávid-Barrett e Fazekas (2020, p. 3), a corrupção nos contratos públicos ocorre, geralmente, quando detentores de informações privilegiadas manipulam diferentes partes do processo de compra, por exemplo: especificando excessivamente o objeto, de modo que apenas uma empresa atenda às condições exigidas; ou divulgando a licitação por um período muito curto, para que apenas empresas com conhecimento prévio tenham tempo para fazer ofertas.

De acordo com Fazekas e Kocsis (2017, p. 1), "o objetivo da corrupção nos contratos públicos é direcionar o contrato ao licitante favorito sem detecção". Segundo Fazekas e Kocsis (op. cit.), esse objetivo é atingido através de contratos diretos (dispensa de licitação) ou através do favorecimento de determinado concorrente, por meio de adaptações nas especificações, restringindo a licitação, e/ou por meio do compartilhando de informações privilegiadas. Em outras palavras, a corrupção nos contratos públicos assume formas diversas (suborno, conluio, abuso dos recursos públicos, apropriação indébita, favoritismo e nepotismo) e o maior risco ocorre durante a fase de *Planejamento* do processo de contratação (BERRU et al., 2019).

A corrupção nos contratos públicos, devido sua característica essencial, ato secreto e invisível, é difícil de medir (GALLEGO; RIVERO; MARTINEZ, 2020), (LÓPEZ-ITURRIAGA; SANZ, 2018) e (WACHS; FAZEKAS; KERTÉSZ, 2020). De acordo com Fazekas, Tóth e King (2016), análises diretas de casos individuais (por exemplo: análises científicas, registros policiais e relatórios de auditorias) são altamente confiáveis no estabelecimento da corrupção. Entretanto, devido a fatores específicos, como o grande número de casos a serem investigados e o alto risco de captura de agentes públicos envolvidos (principalmente nos países com elevado índice de corrupção), segundo Fazekas, Tóth e King (op. cit.), esses tipos de análises, compostos por investigação e comprovação, ocorrem em menor grau, restringindo-se apenas aos órgãos policiais e tribunais, nível organizacional.

Dados são de vital importância para a análise eficiente do risco de corrupção. A grande quantidade de dados administrativos coletados pelos governos nos últimos anos, Kuziemski e Misuraca (2020), Rozario e Issa (2020) e Sousa et al. (2019), representa uma oportunidade para o estudo do risco da corrupção sob um novo panorama, usando novas técnicas de ciência de dados (WACHS; FAZEKAS; KERTÉSZ, 2020). De acordo com Fazekas, Tóth e King (2016), Fazekas e Kocsis (2017), Fazekas e Tóth (2018), Wachs, Fazekas e Kertész (2020) e Dávid-Barrett e Fazekas (2020), recentemente houve um aumento no uso de dados para descobrir padrões desconhecidos de corrupção em contratos públicos.

Atualmente, existem muitos dados disponíveis no setor público, Chen e Zhang (2014), Kuziemski e Misuraca (2020), Rozario e Issa (2020) e Sousa et al. (2019), que podem ser analisados. Somando-se a isso, à medida que o volume de dados aumenta, novos métodos de análise de dados são empregados em finalidades diversas, principalmente, na iniciativa privada, por exemplo: análise de crédito, Louzada, Ara e Fernandes (2016); prevenção de fraudes corporativas, Jans, Lybaert e Vanhoof (2010); e detecção de fraudes financeiras, Ngai et al. (2011) e Phua et al. (2010). Diante disso, com as iniciativas de dados abertos, Attard et al. (2015), novos métodos de análise de dados, Rozario e Issa (2020) e Sousa et al. (2019), vêm sendo utilizados como ferramentas importantes no setor público.

Ao revisar a literatura sobre prevenção de irregularidades e ilegalidades em contratos públicos, verificou-se que poucos estudos investigaram o tema com métodos de análise dados. Com isso, tendo em vista a grande quantidade de dados de contratos públicos disponíveis atualmente, Fazekas, Tóth e King (2016) e Ralha e Silva (2012), e a capacidade das técnicas de

machine learning de extrair informações úteis nesse enorme volume de dados, Chen e Zhang (2014), é extremamente importante que essas técnicas sejam utilizadas na prevenção de irregularidades em contratos públicos. Os principais estudos quantitativos sobre o assunto são apresentados, de forma resumida, Tabela 2, (Seção 2.3.2).

2.3 Emprego da análise de dados na prevenção de fraudes

Ultimamente, com a facilidade de acesso a dados, privados e públicos, métodos de análise de dados baseados em inteligência artificial, dentre os quais, *data mining* (mineração de dados), *machine learning* (aprendizado de máquina) e *deep learning* (aprendizagem profunda), vêm sendo cada vez mais utilizados e têm possibilitado uma série de benefícios, otimizando processos e facilitando a tomada de decisão em muitas áreas. Amplamente empregados na prevenção e detecção de fraudes financeiras em instituições privadas (Seção 2.3.1), aos poucos, esses métodos vêm sendo utilizados na prevenção de fraudes em contratos públicos (Seção 2.3.2), sendo valiosos no combate à corrupção.

2.3.1 Prevenção de fraudes financeiras

Para o Oxford English Dictionary (1999, p. 562 apud NGAI et al., 2011, p. 559), fraude financeira é definida como "engano ilícito ou ato criminoso destinado a resultar em ganho financeiro ou pessoal". A fraude financeira é vista também como "um ato deliberado que é contrário à lei, regra ou política com intenção de obter benefício financeiro não autorizado", Wang et al. (2006, p. 1120 apud NGAI et al., 2011, p. 559).

Dentre os crimes que mais se destacam nesse tipo de fraude, têm-se as fraudes com cartões de crédito, as fraudes corporativas e a lavagem de dinheiro (NGAI et al., 2011). Segundo Ngai et al. (op. cit.), esses crimes causam grande impacto econômico e são difíceis de serem detectados através de tecnologias rudimentares. Por esses motivos, de acordo com Neves, Silva e Carvalho (2019) e Sousa et al. (2019), ultimamente, instituições privadas e públicas têm recorrido à utilização de sistemas, baseados em inteligência artificial, como ferramenta de auxílio às atividades de fiscalização e controle.

Presentes em muitas instituições financeiras, no setor privado, Ngai et al. (2011), técnicas de mineração de dados, utilizadas na detecção de fraudes, foram influenciadas por diversos estudos, dentre os quais, Fawcett e Provost (1997). Em estudo empírico, baseado em métodos

automáticos de detecção de fraude, Fawcett e Provost (op. cit.) utilizaram técnicas de mineração de dados para verificar a existência de alterações suspeitas no comportamento de usuários em um grande banco de dados de transações telefônicas. O modelo desenvolvido foi aplicado em um problema de detecção de fraudes de clonagem de telefone celular. Os resultados indicaram que abordagens automáticas têm desempenho superior aos métodos manuais.

Nos últimos anos, aplicações de técnicas de mineração de dados, como sistema de apoio à detecção de fraudes financeiras, têm sido recorrentes. Vários estudos relevantes sobre o assunto foram desenvolvidos, por exemplo: Zhang e Zhou (2004 apud NGAI et al., 2011), utilização de técnicas de mineração de dados em aplicações financeiras (mercado de ações, previsão de falência e detecção de fraude); e Phua et al. (2010), detecção automatizada de fraude, baseada em técnicas de mineração de dados (fraude de transações de crédito, fraude de assinaturas em telecomunicações, fraude de seguro de automóvel e outras fraudes similares).

Ngai et al. (2011) analisaram 49 artigos sobre este tema, publicados no período de 1997 a 2008. Utilizando como referência a categorização de crimes financeiros do *Federal Bureau of Investigation - FBI* (Departamento Federal de Investigação), descrita na Tabela 1, esses artigos foram classificados em quatro tipologias distintas: *i)* fraudes bancárias; *ii)* fraudes de seguro; *iii)* fraudes de valores mobiliários e de mercadorias; e *iv)* outras fraudes financeiras. Para quantificar a fraude nessas quatro tipologias, os autores propuseram a utilização de seis métodos de mineração de dados (classificação, regressão, agrupamento, previsão, detecção de anomalia e visualização).

Tabela 1: Classificação para fraudes financeiras com base no *FBI*.

Categorias baseadas em fraudes financeiras	**Atividades fraudulentas**
Fraude bancária	Fraude hipotecária, Lavagem de dinheiro (Confisco bens)
Fraude de seguro	Fraude na saúde, Fraude de seguros
Fraude de valores mobiliários e mercadorias	Fraude de valores mobiliários e mercadorias
Outras fraudes financeiras relacionadas	Fraude corporativa, Fraude de *marketing* de massa

Fonte: Ngai et al. (2011, p. 561).

Conforme mostrado na Tabela 1, a categorização de crimes financeiros do *FBI*, utilizada por Ngai et al. (op. cit.), é composta por dois níveis: nível superior, composto por fraudes financeiras; e nível inferior, composto por atividades fraudulentas. O nível superior se subdivide em fraude bancária, fraude de seguro, fraude de valores mobiliários e de mercadorias e em outras fraudes financeiras relacionadas; e o nível inferior se subdivide em fraude hipotecária, confisco de bens/lavagem de dinheiro, fraude na saúde, fraude de seguros, fraude de valores

mobiliários e de mercadorias, fraudes corporativas e fraude de *marketing* de massa.

Os resultados da análise, realizada por Ngai et al. (2011), mostraram que técnicas de mineração de dados continua sendo aplicadas na detecção de fraudes financeiras e os algoritmos *logistic regression* (regressão logística), *neural network* (rede neural), *bayesian belief network* (rede de crenças bayesianas) e *decision tree* (árvore de decisão) são os principais métodos utilizados. A estrutura para a revisão e análise de artigos, proposta pelos autores, se subdividiu em duas partes, fraudes financeiras e técnicas de mineração de dados.

As fraudes financeiras foram divididas em: *i)* fraude bancária (cartão de crédito e lavagem de dinheiro); *ii)* fraude de seguro (seguro de automóvel, seguro de colheita e seguro de saúde); *iii)* fraude de valores mobiliários e de mercadorias; e *iv)* outras fraudes financeiras relacionadas (fraude corporativa e fraude de *marketing* em massa). As técnicas utilizadas, quais sejam, *neural network*, *regression model*, *naive bayes*, *decision tree*, *fuzzy logic*, *CART*, *genetic algorithm*, *k-nearest neighbors* e *bayesian belief network*, foram empregadas em tarefas de classificação, agrupamento, detecção de *outliers*, predição, regressão e visualização.

Ainda sobre aplicação de técnicas de mineração de dados, como ferramentas de detecção automatizada de fraude, outra contribuição importante foi dada por Phua et al. (2010). Os autores relacionaram 51 artigos, publicados no período de 1994 a 2004, para análise e categorização. Nesse estudo, foram apresentados os métodos e técnicas de detecção de fraudes, mais relevantes na literatura, bem como, os desafios existentes para os diferentes tipos de dados no campo de estudo de mineração de dados.

Tomando-se como base os artigos analisados por Phua et al. (2010), as fraudes que podem ser detectadas, a partir de técnicas de mineração de dados, foram classificadas em 4 grupos: *i)* fraude interna, gerada por agentes da organização; *ii)* fraude de seguro, subdividida em seguro residencial, seguro agrícola, seguro de automóvel e seguro médico; *iii)* fraude de crédito, ligada à concessão de crédito e transação com cartão de crédito; e *iv)* fraude de telecomunicações. Nessa análise, a tipologia que recebeu maior atenção foi a transação de crédito (tipologia, pertencente ao grupo fraude de crédito).

Embora utilizando diferentes técnicas de mineração de dados, essas pesquisas têm em comum o mesmo objetivo, a automatização do processo de detecção de fraude. De acordo com Jans, Lybaert e Vanhoof (2010), muitos estudos investigaram diferentes tipos de fraudes, utilizando métodos supervisionados como tarefa de mineração de dados, contudo, os resultados

se limitaram apenas em descobrir se *houve* ou *não houve* fraude, característica particular de problemas de classificação binária.

Com isso, Jans, Lybaert e Vanhoof (2010) propuseram uma abordagem mais abrangente, que combinou controle de detecção com prevenção de fraude. Nesse estudo, os autores consideraram o método não supervisionado mais apropriado para esse tipo de problema, pois o mesmo gera informações sobre todo o conjunto de dados, em vez de apenas um aspecto dele, ou seja, *fraudulento* ou *não fraudulento*. Para esses autores, estratégias que priorizam este tipo de resultado, isto é, *detecção* e *prevenção*, são mais efetivas na redução de fraudes no mundo real, pois diminuem substancialmente a exposição ao risco, tornando-se desnecessárias ações reativas, consequências comuns em estratégias que priorizam apenas o controle de detecção.

Nesse contexto, segundo Jans, Lybaert e Vanhoof (2010), a utilização de métodos não supervisionados é uma estratégia importante de apoio à redução do risco de fraude. No estudo proposto por Jans, Lybaert e Vanhoof (op. cit.), sobre redução do risco de fraude interna, os resultados dos algoritmos de agrupamento (método não supervisionado), utilizados em dados coletados de um provedor internacional de serviços financeiros, foram satisfatórios para a avaliação do risco de fraude interna no processo de aquisição. Segundo os autores, seria impossível obter resultados equivalentes com a utilização de métodos supervisionados.

2.3.2 Prevenção de fraudes em contratos públicos

Embora pouco utilizados no âmbito governamental, nos últimos anos, sobretudo no período de 2012 a 2017, houve um interesse crescente por métodos de análise de dados, baseados em inteligência artificial, no setor público, Sousa et al. (2019), dentre os quais, mineração de dados, *machine learning* e aprendizagem profunda. Ao revisar a literatura sobre o assunto, vários estudos com aplicações práticas desses métodos no setor público foram encontrados.

Dentre os estudos encontrados, utilizando técnicas de *machine learning*, merece destaque as seguintes aplicações práticas, úteis ao setor público: monitoramento e fiscalização ambiental utilizando dados de satélites, Handan-Nader, Ho e Liu (2021); detecção automatizada de aquisição ilegal de armas na *dark web*[3], Saini e Bansal (2019); análise automatizada de relatórios de crimes em agências policiais americanas, Ku e Leroy (2014); previsão de decisões no Tribunal Europeu dos Direitos Humanos, Medvedeva, Vols e Wieling (2020); previsão da

[3]Redes que fornecem anonimato para usuários e provedores de conteúdo (SAINI; BANSAL, 2019).

duração e desvios de custos em projetos de infraestrutura na Turquia, Erdis (2013); previsão de fidelidade de clientes em programa social belga, De Cnudde e Martens (2015); e classificação de usuários de serviços de governo eletrônico no Egito, Mostafa e El-Masry (2013).

Sobre o emprego desses métodos na prevenção de fraudes, embora sejam largamente utilizados na prevenção e detecção de fraudes financeiras, na iniciativa privada, Ngai et al. (2011), Jans, Lybaert e Vanhoof (2010) e Phua et al. (2010); no setor público, suas aplicações não estão totalmente consolidadas para este fim, detectar precocemente fraudes e/ou previnir atos fraudulentos e corruptos (LIMA; DELEN, 2020). Sabe-se que órgãos de fiscalização e controle do setor público têm acesso a grandes volumes de dados que são capturados e armazenados por esse setor, Chen e Zhang (2014) e Sousa et al. (2019), no entanto, iniciativas de análise quantitativa de dados, destinadas à prevenção e detecção de fraudes e corrupção no setor público, ainda são muito limitadas, Wachs, Fazekas e Kertész (2020) e Fazekas e Kocsis (2017).

No setor público, a maior parte dos processos, envolvendo receitas e despesas públicas, é operacionalizada por sistemas informatizados, como *SIAFI*[4] e *SIASG*, que mantêm registros pormenorizados da execução dos recursos públicos. Por exemplo, de 1998 a 2009, o *SIAFI* registrou um aumento de 223% no número médio de transações executadas mensalmente no sistema, isto é, pesquisas de dados e inserções (RALHA; SILVA, 2012). Esse enorme crescimento no volume de dados reforça, mais ainda, a necessidade de mecanismos de fiscalização e controle, como técnicas de reconhecimento de padrões em dados, em áreas específicas do governo, sobretudo, como ferramentas de apoio ao combate à corrupção.

Um grande número de pesquisas, utilizando abordagens teóricas, correlações e análises estatísticas fundamentadas em regressão, procurou explicar as causas e consequências da corrupção (LIMA; DELEN, 2020). Apesar disso, nos últimos anos, em decorrência do acúmulo crescente de dados públicos, Sousa et al. (2019), técnicas de *machine learning* passaram a ser desenvolvidas e, aos poucos, vem sendo empregadas com essa finalidade (LIMA; DELEN, 2020). Na pesquisa conduzida por Lima e Delen (op. cit.), sobre corrupção entre países, três algoritmos de *machine learning* (*random forest*, *neural network* e *support vector machine*) são utilizados para descobrir os preditores de corrupção mais importantes em 132 países.

Em outro estudo, sobre previsão da corrupção pública, López-Iturriaga e Sanz (2018), com base em fatores econômicos e políticos, utilizaram o algoritmo *neural network* para prever

[4]Sistema Integrado de Administração Financeira (SIAFI).

indícios de corrupção nas províncias espanholas, em até 3 anos antes da consumação do crime. O modelo, utilizado López-Iturriaga e Sanz (2018), funcionou como um sistema de alerta precoce, que usou, como base, casos denunciados pela mídia e/ou casos reais que foram a tribunal, no período de 2000 a 2012. Dentre os preditores de corrupção encontrados, a tributação do estado e o crescimento econômico foram os preditores mais relevantes.

No Brasil, inovações, baseadas em inteligência artificial, vêm sendo implementadas por órgãos de fiscalização e controle do governo federal, como Tribunal de Contas da União e Controladoria-Geral da União (NEVES; SILVA; CARVALHO, 2019). Técnicas de auditoria assistida por computador, como *ALICE*[5], *ADELE*[6], *MONICA*[7] e *SOFIA*[8], utilizadas como sistemas de alerta contra fraude e corrupção em processos de compras do governo federal, Neves, Silva e Carvalho (2019), podem reduzir custos e otmizar significativamente a execução de tarefas no processo de contratação pública. A utilização dessas inovações, como prática contínua e permanente de gestão de riscos e controle preventivo, está prevista no Art. 169, da Lei nº 14.133/21 (BRASIL, 2021, p. 67).

Mineração de dados e *machine learning* são métodos computacionais recentemente empregados em auditorias do setor público, Neves, Silva e Carvalho (2019), que possibilitam a análise crítica e meticulosa de dados e a predição de eventos futuros ou sequência de eventos (BOUTABA et al., 2018). Estudos recentes, Tabela 2, sobre a utilização dessas inovações no setor público, dentre os quais, Sun e Sales (2018), Henrique, Sobreiro e Kimura (2020), Hooda, Bawa e Rana (2018, 2020), Gallego, Rivero e Martınez (2020), Ralha e Silva (2012), Carneiro et al. (2020) e Rozario e Issa (2020), descrevem as oportunidades e os desafios da incorporação de sistemas, baseados em métodos estatísticos, ciência de redes e inteligência artificial, no processo de auditoria de contratos públicos.

Sun e Sales (2018) desenvolveram um modelo a partir de algoritmos de *deep neural network* (redes neurais profundas) para a previsão do risco de irregularidades em contratos públicos no âmbito da Controladoria-Geral da União. Para verificar o desempenho preditivo dessa técnica, o modelo foi testado com outros algoritmos (redes neurais tradicionais, regressão logística e análise de discriminante). Segundo os autores, o modelo foi desenvolvido para funcionar como um sistema de alerta precoce na triagem inicial de fornecedores.

[5] Analisador de Licitações, Contratos e Editais (ALICE).
[6] Análise de Disputa em Licitações Eletrônicas (ADELE).
[7] Monitoramento Integrado para o Controle de Aquisições (MONICA).
[8] Sistema de Orientação sobre Fatos e Indícios para o Auditor (SOFIA).

Utilizando dados públicos disponíveis em bases de dados do governo federal brasileiro, Henrique, Sobreiro e Kimura (2020) avaliaram o desempenho de técnicas estatísticas (*logistic regression* e discriminant analysis) e de *machine learning* (*k-nearest neighbours*, *support vector machine* e *random forest*). Dentre essas cinco técnicas, utilizadas para classificar contratos públicos em níveis de risco de não conformidade contratual, o algoritmo *k-nearest neighbours* (k-vizinho mais próximo) foi a técnica que apresentou o melhor desempenho. Segundo os autores, os modelos, gerados a partir dessas técnicas, podem ter grande utilidade como ferramentas de apoio à decisão, em órgão de auditoria e controle.

Na pesquisa conduzida por Hooda, Bawa e Rana (2018, 2020), sobre classificação de empresa fraudulenta na Índia, dados anuais de 777 empresas foram coletados com a finalidade de explorar a utilidade de dez algoritmos de *machine learning* para melhorar a qualidade do trabalho de auditoria governamental. Utilizando nove fatores de risco, atuais e históricos, os algoritmos foram comparados e testados para serem aplicados na predição de empresas com risco elevado. Os resultados dos algoritmos *bayesian network* e *decision tree* (*J48*) retornaram uma acurácia de 93% para a classificação de empresas suspeitas.

Em outro estudo, sobre previsão de má conduta em contratos públicos (quebra de contrato, ineficiência e corrupção), Gallego, Rivero e Martınez (2020) avaliaram o desempenho da técnica estatística (*LASSO*[9] *regression*) e de *machine learning* (*gradient boosting*) em um conjunto de dados colombiano, contendo mais de dois milhões de contratos públicos. Por meio dessas técnicas, foram obtidos os seguintes resultados: regressão de *LASSO*, acurácia de 78,8%; e *gradiente boosting*, acurácia de 91,2%. De acordo com os autores, os modelos foram desenvolvidos para melhorar a supervisão das compras públicas, reduzir a redundância e detectar situações que resultem em perdas econômicas para o governo.

Ralha e Silva (2012) desenvolveram um sistema de mineração de dados multi-agente para a detecção de cartéis em aquisições do governo federal brasileiro. Segundo os autores, estudos dessa natureza apresentam dois desafios, o grande volume de dados usado e as estratégias dinâmicas e diversificadas que as empresas usam para realizar operações fraudulentas. De acordo com Ralha e Silva (op. cit.), o conhecimento extraído de sistema como esse poderia, segundo especialistas em auditoria, ajudar na detecção, prevenção e monitoramento de cartéis que agem nos processos de licitações públicas.

[9] *Least Absolute Shrinkage and Selection Operator* (operador de seleção e contração menor absoluto).

Carneiro et al. (2020) desenvolveram um sistema híbrido de suporte à decisão, baseado em regras e *machine learning*, para facilitar a detecção de fraudes na contratação pública em Portugal. O método proposto combina um banco de dados orientado a grafos, um mecanismo de regras e uma camada de visualização georeferenciada para melhorar a transparência nas compras públicas. De acordo com Carneiro et al. (op. cit.), por meio da automatização, isto é, classificação preliminar das instâncias, o método reduz o esforço manual do auditor, permitindo o direcionamento da capacidade investigativa do auditor para os casos relevantes.

Ainda sobre o emprego de técnicas de *machine learning* na análise de risco de contratos públicos, Ovsyannikova e Domashova (2020) utilizaram o algoritmo *neural network* (rede neural) para prever os resultados da execução de contratos públicos na Rússia. Segundo os autores, os resultados possibilitaram a identificação de contratos em risco de não desempenho em tempo hábil, a predição de retirada de licenças de organizações de crédito ou seguradoras e o monitoramento de transações suspeitas.

Nesta outra pesquisa, sobre a extensão e custos da corrupção na infraestrutura de transporte europeia, utilizando dados de mais de 40.000 contratos governamentais do período de 2009 a 2014, Fazekas e Tóth (2018) virificaram que as normas de concorrência, legalmente estabelecidas, eram contornadas e os contratos premiavam continuamente as empresas pertencentes ao grupo corrupto. De acordo com os autores, a corrupção direcionava os gastos de infraestrutura para um valor alto, em oposição a projetos de investimento de pequeno valor, e aumentavam os preços em 30 a 35%, em média, com os maiores excessos nas regiões de alto risco de corrupção.

Na pesquisa realizada por Titl e Geys (2019), sobre doações políticas e alocação de contratos públicos, foi constatado que empresas doadoras recebem mais contratos de compras menos rigorosos, enfrentam menos concorrência em procedimentos de compras mais rigorosos e tendem a ganhar com lances mais acima do custo estimado do contrato de compras. Nessa pesquisa, foi constatado que doações a partidos políticos estimula o favoritismo nas alocações de compras públicas. Titl e Geys (po. cit.) usaram como base um conjunto de dados abrangente, cobrindo todos os contratos públicos e todas as doações corporativas para os principais partidos políticos da República Tcheca, no período de 2007 a 2014.

Tabela 2: Emprego da análise de dados na prevenção de fraudes em contratos públicos.

Principais estudos	**Abordagens de pesquisa**	**Dados públicos utilizados***
Lima e Delen (2020)	*Machine learning*	A, B, C, D
López-Iturriaga e Sanz (2018)	*Machine learning*	E
Sun e Sales (2018)	*Machine learning*	F, G, H, I, J, K
Henrique, Sobreiro e Kimura (2020)	*Machine learning*	F, G, I, J, K, L
Gallego, Rivero e Martınez (2020)	*Machine learning*	M
Hooda, Bawa e Rana (2018, 2020)	*Machine learning*	N
Ovsyannikova e Domashova (2020)	*Machine learning*	O
Rozario e Issa (2020)	Método estatístico (*matching*)	P
Ralha e Silva (2012)	*Multi-agent system*	I
Fazekas e Tóth (2018)	Método estatístico	Q
Dávid-Barrett e Fazekas (2020)	Programação adaptativa	R
Wachs, Fazekas e Kertész (2020)	Ciência de redes	S
Fazekas e Kocsis (2017)	Método estatístico	S
Fazekas, Tóth e King (2016)	Método estatístico	T
Carneiro et al. (2020)	Ciência de redes	U
Titl e Geys (2019)	Método estatístico	V

*

A *Ease of doing business index* (Índice de facilidade de fazer negócios): <https://www.doing business.org/>

B *The Heritage Foundation* (Fundação Heritage): <https:// www.heritage.org/>

C *Transparency International* (Transparência Internacional): <https://www.transparency.org /en/>

D *Human Development Reports* (Relatórios de Desenvolvimento Humano): <http://hdr.undp. org/>

E *Corruption database gathered by El Mundo* (banco de dados de corrupção reunido pelo jornal espanhol *El Mundo*): <http://www.elmundo.es/grafico/espana/2014/11/03/5453d2e6268e3e8d7f8b456c.html>

F Relação Anual de Informações Sociais (RAIS)

G Receita Federal do Brasil (RFB): <http://receita.economia.gov.br/orientacao/tributaria/cadastros/cadastro-nacional-de-pessoas-juridicas-cnpj/dados-publicos-cnpj>

H Sistema Integrado de Administração Financeira (SIAFI)

I Sistema Integrado de Administração de Serviços Gerais (SIASG)

J Sistema de Convênios (SICONV)

K Tribunal Superior Eleitoral (TSE)

L Cadastro de Empresas Inidôneas e Suspensas (CEIS)

M *Datos Abiertos de Colombia* (Dados abertos da Colômbia): <http://datos.gov.co.>

N *Comptroller and Auditor General* (Escritório Geral de Auditoria da Índia): <https://www.cag.gov.in/en>

O *Unified procurement information system* (Sistema unificado de informações de compras da Rússia)

P *Procurement dataset of a U.S. county* (Dados de aquisição de um condado dos EUA)

Q *EU public procurement directives* (Diretivas da União Europeia em matéria de contratos públicos): <http://digiwhist.eu/resources/data/>

R *Dataset of World Bank-funded development aid tenders and contracts* (conjunto de dados de licitações e contratos de ajuda ao desenvolvimento financiado pelo Banco Mundial): <https://finances.worldbank.org/Procurement/Major-Contract-Awards/kdui-wcs3>; <http://data.worldbank.org/data-catalog/projects-portfolio>; <http://projects.worldbank.org/procurement/procurementsearch?lang>; <http://www.govtransparency.eu/index.php/2018/02/13/data-publication-foreign-aid-of-world-bank-europeaid-and-iadb/>

S *Tenders Electronic Daily - TED* (Licitações Eletrônicas Diárias, jornal oficial dos contratos públicos da União Europeia): <http://digiwhist.eu/resources/data/>

T *Hungarian public procurement database* (Base de dados de contratos públicos da Hungria): <https://zenodo.org/record/3537986#.XcmMPkX0mgk>

U *Portal BASE.gov* (Portal de contratos públicos de Portugal): <http://www.base.gov.pt/>

V *Dataset from the Czech Republic* (Conjunto de dados da República Tcheca, contratos públicos e doações corporativas a partidos políticos): <http://wiki.zindex.cz/doku.php?id=en: objem_zakazek>

Fonte: Elaboração própria.

3 Método

Neste trabalho, métodos de análise de dados são utilizados para classificar fornecedores do governo federal brasileiro (GFB) em dois níveis de risco de inadimplência contratual (grupo de baixo risco, *Gp BR*; e grupo de alto risco, *Gp AR*). Embora os modelos elaborados se limitem a apenas dois níveis de risco, isto é, *adimplente*, *Gp BR* e *inadimplente*, *Gp AR*, quando utilizados para identificar fornecedores suspeitos, com alta propensão à inadimplência ou à insolvência, permitem que medidas preventivas sejam priorizadas, por parte do setor público, evitando assim possíveis irregularidades na execução de contratos.

Duas categorias de técnicas de pontuação de crédito, Wang, Ma e Yang (2014), Wang, Hao et al. (2011) e Wang, Ma, Huang et al. (2012), Wang e Ma (2012), Marqués, Garcıa e Sánchez (2012), Yeh, Chi e Lin (2014) e Feng et al. (2018), foram utilizadas na elaboração dos modelos de classificação propostos: *i)* técnicas estatísticas tradicionais (*discriminant analysis* [análise de discriminante]; e *logistic regression* [regressão logística]); e *ii)* técnicas de *machine learning* (*k-nearest neighbors* [k-ésimo vizinho mais próximo]; *naive bayes* [classificador de bayes ingênuo]; *support vector machine* [máquina de vetor de suporte]; *decision tree* [árvore de decisão]; *random forest* [floresta aleatória]; *bagging* [agregação de *bootstrap*]; *gradient boosting* [aumento de gradiente]; e *neural network* [rede neural]).

As técnicas *i)* e *ii)*, acima, utilizadas na construção dos modelos, foram processadas no ambiente de desenvolvimento *Jupyter Notebook*, utilizando a linguagem de programação *python* versão 3.7.3 e suas bibliotecas (*numpy* 1.16.2, *pandas* 0.24.2, *scikit-learn* 0.20.3, *matplotlib* 3.0.3 e *seaborn* 0.9.0). O computador utilizado, *desktop HP pavilion*, possui as seguintes configurações: sistema operacional, *microsoft windows* versão 10.0.18363.900; processador, *intel core i7*; CPU, 3,4 *GHz*; e memória *RAM*, 16 *GB*.

3.1 Dados

Os dados para a construção dos modelos são do período de 2015 a 2019 e foram coletados de três bases de dados do *GFB*: *i)* Sistema de Administração de Serviços Gerais (ComprasNet/SIASG); *ii)* Cadastro de Entidades Inidôneas e Suspensas (CEIS)[10], da Controladoria-Geral da União; e *iii)* Cadastro Nacional de Pessoas Jurídicas (CNPJ)[11], da Receita Federal do Brasil.

[10] https://www.portaltransparencia.gov.br/sancoes/ceis?ordenarPor=nomedirecao=asc

[11] http://receita.economia.gov.br/orientacao/tributaria/cadastros/cadastro-nacional-de-pessoas-juridicas-cnpj/dados-publicos-cnpj

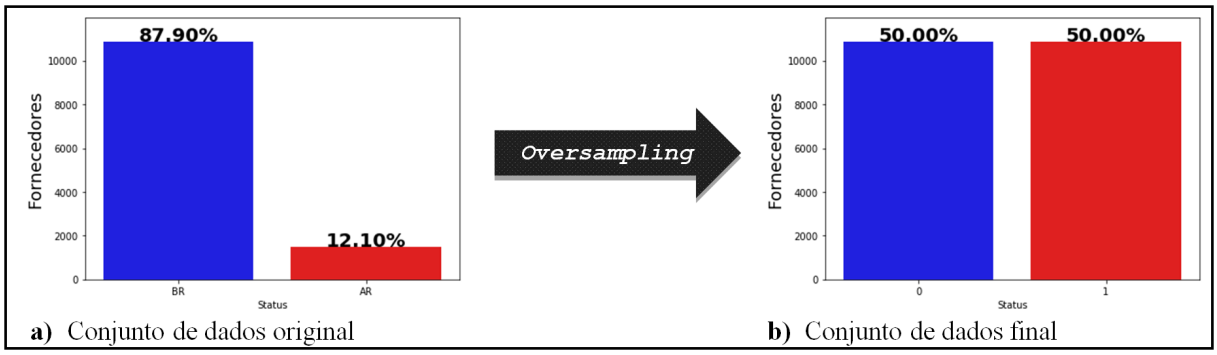
Os dados coletados do *SIASG* (cerca de 88% do total) e os dados coletados do *CEIS* (cerca de 12% do total) são, respectivamente, de fornecedores que *não possuem* e de fornecedores que *possuem* restrições para a venda de bens e/ou de serviços ao *GFB*.

3.1.1 Coleta de dados

A coleta dos dados no sistema *ComprasNet/SIASG*, plataforma onde se operacionalizam as compras do *GFB*, foi feita por meio de uma *Application Programming Interface - API*[12] (Interface de programação de aplicativos). Essa interface permitia um limite máximo de resposta de 500 *outputs*, por consulta realizada, com isso, desenvolveu-se um *script* em *python*, utilizando pacotes *requests*, *json* e *pandas*, para automatizar a extração dos dados. Após o processo de extração, os dados coletados foram organizados em um único arquivo no formato *CSV*.

A amostra de fornecedores utilizada na modelagem, contendo dados do *SIASG* e do *CEIS*, é ilustrada na Figura 5a. Conforme apresentado na Seção 2.2.2, as sanções administrativas previstas no Art. 156, da Lei nº 14.133/21, Brasil (2021, p. 64), que levam ao registro no *CEIS* são: *i)* impedimento de licitar e contratar; e *ii)* declaração de inidoneidade para licitar ou contratar.

Figura 5: Distribuição do *status* de risco dos fornecedores no conjunto de dados.



a) Conjunto de dados original **b)** Conjunto de dados final

Fonte: Elaboração própria.

A Figura 5 mostra a distribuição do *status* de risco dos fornecedores, antes (Figura 5a) e após (Figura 5b) do balanceamento do conjunto de dados. Na construção dos modelos, a amostra original (Figura 5a) foi dividida em dois grupos: fornecedores que *adimplem*, *Gp BR* (10.878 fornecedores, 87,89% do total) e fornecedores que *inadimplem*, *Gp AR* (1.498 fornecedores, 12,10% do total). Os fornecedores rotulados como adimplentes são aqueles que não possuem restrições para a venda de bens ou serviços ao *GFB* e os fornecedores rotulados como

[12] http://compras.dados.gov.br/docs/home.html

inadimplentes são aqueles que possuem restrições para a venda de bens e/ou serviços ao *GFB* (fornecedores cadastrados no *CEIS*).

3.1.2 Balanceamento artificial de dados

A técnica *oversampling* (sobreamostragem) foi empregada para balancear as classes do conjunto de dados original (CHAWLA et al., 2002). Após o balanceamento, o novo conjunto de dados (Figura 5b), gerado a partir da sobreamostragem da classe minoritária, *Gp AR*, passou a ter quantidades equivalentes de dados nas duas classes: classe $\{0\}$ (*Gp BR*), 10.878 fornecedores (*adimplentes*); e classe $\{1\}$ (*Gp AR*), 10.878 fornecedores (*inadimplentes*). Para habilitar os modelos de classificação propostos, os dados da nova amostra (Figura 5b) foram divididos em dados de treino e dados de teste na proporção descrita na Tabela 3.

Tabela 3: Composição do conjunto de dados utilizado nos modelos de classificação.

Base final (Fig. 5b)	**Fornecedores**		**Porcentagem**		**Total de fornecedores**	**Porcentagem sobre o total**
	Adimple	**Inadimple**	**Adimple**	**Inadimple**		
Amostra de treino	7.677	7.552	35,29	34,71	15.229	70
Amostra de teste	3.326	3.201	15,29	14,71	6.527	30
Total da amostra	**10.878**	**10.878**	**50**	**50**	**21.756**	**100**

Fonte: Elaboração própria.

A Tabela 3 organiza o conjunto de dados final, composto por 21.756 exemplos (Figura 5b), na seguinte proporção: amostra de treino, com 15.229 exemplos (70% do total), sendo 7.677 adimplentes e 7.552 inadimplentes; e amostra de teste, com 6.527 exemplos (30% do total), sendo 3.326 adimplentes e 3.201 inadimplentes. O balanceamento do conjunto de dados original foi necessário, pois o desequilíbrio entre as classes estava prejudicando o desempenho dos classificadores, que eram influenciados pela classe majoritária, isto é, mais preciso na classe majoritária, classe priorizada, do que na classe minoritária, classe ignorada.

3.1.3 Variáveis dependentes e independentes

Vários estudos discorrem sobre o emprego de variáveis preditoras, estatisticamente significativas, em modelos de avaliação de risco de crédito, por exemplo: Wang, Hao et al. (2011) e Wang, Ma, Huang et al. (2012), caráter do consumidor, capital, garantia, capacidade, condições econômicas; Jimenez, Salas e Saurina (2006), garantia; Lawrence, Smith e Rhoades (1992), padrões recentes de inadimplência; e Hand e Henley (1997) e Lim, Loh e Shih (2000), tempo de residência, tempo de serviço, tempo de relacionamento com o banco, código de endereçamento

postal, indicadores de posse de cartões de crédito, indicador de poupança, faixa etária, estado civil, renda familiar, escolaridade e custo de habitação.

Nesta pesquisa, as variáveis utilizadas foram extraídas das bases do *ComprasNet/SIASG*, *CEIS* e *RFB*. Na amostra utilizada, a variável dependente, *Status*, corresponde à situação de risco em que o fornecedor se encontra (*adimplente*, {0}; ou *inadimplente*, {1}), e as variáveis independentes correspondem a características do fornecedor, relacionadas à credibilidade. Além da variável binária *Status*, cinco características foram utilizadas como variáveis explicativas: porte do fornecedor (*Porte*); número de *CNAE*[13] secundários (*Nr_CNAE_Sec*); número de ocorrências (*Nr_Ocr*); tempo de existência (*Tp_Exis*); e total da venda (*Tot_Venda*). Essas variáveis se relacionam a três componentes de risco (*Cpcd Op Forn*; *Hist P Forn*; e *Pf Forn*)[14] apresentados no apêndice A.

Essas variáveis foram escolhidas com base em seis estudos importantes: Jimenez, Salas e Saurina (2006), Lawrence, Smith e Rhoades (1992), Tang et al. (2009), Hand e Henley (1997), Lim, Loh e Shih (2000) e Sun e Sales (2018). Todos esses estudos utilizaram variáveis de entrada equivalentes na avaliação do risco de crédito da contraparte (por exemplo: renda do mutuário, histórico de dívida, idade e valor do empréstimo). As variáveis, dependente e independente, do banco de dados, a descrição, a função, o nível de medição e as referências a elas vinculadas, estão organizadas na Tabela 4.

Tabela 4: Variáveis do conjunto de dados utilizadas nos modelos.

Variável	**Descrição da variável**	**Função**	**Tipo**	**Referências***
Porte	Porte do fornecedor (renda anual)	entrada	nominal	A
Nr_CNAE_Sec	Número de CNAE secundários	entrada	contínua	B
Nr_Ocr	Número de ocorrências (histórico de dívida)	entrada	contínua	C
Tp_Exis	Tempo de Existência (idade)	entrada	contínua	D
Tot_Venda	Total de Venda (valor do contrato)	entrada	contínua	E
Status	Adimplentes/Inadimplentes	alvo	binária	-

*

A Sun e Sales (2018), Jimenez, Salas e Saurina (2006), Lawrence, Smith e Rhoades (1992) e Lim, Loh e Shih (2000)
B Sun e Sales (2018)
C Sun e Sales (2018) e Jimenez, Salas e Saurina (2006)
D Sun e Sales (2018), Jimenez, Salas e Saurina (2006), Lawrence, Smith e Rhoades (1992) e Hand e Henley (1997)
E Sun e Sales (2018), Jimenez, Salas e Saurina (2006) e Lawrence, Smith e Rhoades (1992)

Fonte: Elaboração própria.

[13]Classificação Nacional de Atividades Econômicas (CNAE).
[14]Capacidade Operacional do Fornecedor; Histórico de Penalidades do Fornecedor; e Perfil do Fornecedor.

3.2 Técnicas utilizadas

Os modelos desenvolvidos nesta pesquisa utilizaram, como base, técnicas estatísticas tradicionais (*Discriminant Analysis - DA*; e *Logistic Regression - LR*) e técnicas de *machine learning* (*K-Nearest Neighbors - KNN*; *Bayesian Network - BN*; *Support Vector Machine - SVM*; *Decision Tree - DT*; *Random Forest - RF*; *Bagging - BG*; *Gradient Boosting - GB*; e *Neural Network - NN*) apresentadas a seguir:

a) ***Discriminant analysis (DA)***. Neste trabalho, a técnica *DA* segue conforme Kimura, Basso e Kayo (2015), Altman (1968) e Mahmoudi e Duman (2015). Idêntico ao método apresentado por Kimura, Basso e Kayo (2015), a estrutura matemática é definida, formalmente, na equação (1).

$$Y = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + a_3 \cdot x_3 + \cdots + a_n \cdot x_n \tag{1}$$

b) ***Logistic regression (LR)***. Neste trabalho, a técnica *LR* segue conforme Kimura, Basso e Kayo (2015), Jardin (2016), Baesens et al. (2003), De Menezes et al. (2017) e Kruppa et al. (2013). Semelhante ao método apresentado por Kimura, Basso e Kayo (2015), a estrutura matemática é definida, formalmente, na equação (2).

$$P_i[Y = 1|X = x_i] = \frac{1}{1 + e^{-z}} \tag{2}$$

c) ***K-nearest neighbors (KNN)***. Neste trabalho, a técnica *KNN* segue conforme Baesens et al. (2003), Le e Viviani (2018) e Garcıa, Marqués e Sánchez (2012). Idêntico ao método apresentado por Baesens et al. (2003), a estrutura matemática é definida, formalmente, na equação (3).

$$d(X_i, X_j) = ||X_i - X_j|| = [(X_i - X_j)^T \cdot (X_i - X_j)]^{\frac{1}{2}} \tag{3}$$

d) ***Bayesian network (BN)***. Neste trabalho, a técnica *BN* segue conforme Baesens et al. (2003) e Friedman, Geiger e Goldszmidt (1997). Semelhante ao método apresentado por Baesens et al. (2003), a estrutura matemática é definida, formalmente, nas equações

(4, 5).

$$p(y|x) = \frac{p(x|y) \cdot p(y)}{p(x)} \tag{4}$$

$$p(x|y) = \prod_{i=1}^{n} p(x_i|y) \tag{5}$$

e) ***Support vector machine (SVM)***. Neste trabalho, a técnica *SVM* segue conforme Li, Wang e He (2013), Chen, Ma e Ma (2009) e Tsai, Hsu e Yen (2014). Idêntico ao método apresentado por Li, Wang e He (2013), a estrutura matemática é definida, formalmente, nas equações (6, 7, 8, 9, 10).

$$Minimize\ \frac{1}{2}\ w^T w + C \sum_{i=1}^{M} \xi_i \tag{6}$$

sujeito a

$$y_i[w^T \Phi(x_i) + b] \geq 1 - \xi_i\ (i = 1, 2, ..., M), \tag{7}$$

$$\xi \geq 0\ (i = 1, 2, ..., M),$$

$$Maximize\ \frac{1}{2}\ \alpha^T Q_\alpha - e^T \alpha \tag{8}$$

sujeito a

$$0 \leq \alpha_i \leq C\ (i = 1, 2, ..., M) \tag{9}$$

$$y^T \alpha = 0$$

A função classificadora final do *SVM* é obtida por:

$$f(x) = sign\left(\sum_{i=1}^{M} \alpha_i y_i K(x, x_i) + b\right) \tag{10}$$

f) ***Decision trees (DT)***. Neste trabalho, a técnica *DT* segue conforme Tsai, Hsu e Yen (2014), Kim e Upneja (2014), Fitzpatrick e Mues (2016) e Baesens et al. (2003). Semelhante ao método apresentado por Tsai, Hsu e Yen (2014), a estrutura matemática é

definida, formalmente, nas equações (11, 12, 13).

$$P_i = \frac{freq(C_j, S)}{|S|} \tag{11}$$

$$Entropia(S) = \sum_{i=1}^{n} -p_i \, log_2 \, p_i \tag{12}$$

$$Ganho(S, A) = Entropia(S) - \sum_{v \in valores(x_i)} \frac{|S_v|}{|S|} \times Entropia(S_v) \tag{13}$$

g) ***Random forests (RF)***. Neste trabalho, a técnica *RF* segue conforme Breiman (2001) e Yeh, Chi e Lin (2014). A técnica *RF* consiste em uma combinação de árvores classificadoras estruturadas por vetores aleatórios independentes distribuídos de forma idêntica, de modo que cada árvore lança um único voto para a classe mais popular, com base nos dados de entrada (BREIMAN, 2001).

h) ***Bagging (BG)***. Neste trabalho, a técnica *BG* segue conforme Breiman (1996) e Wang, Ma e Yang (2014), Wang, Hao et al. (2011) e Wang, Ma, Huang et al. (2012). Idêntico ao método apresentado por Breiman (1996), a estrutura matemática é definida, formalmente, na equação (14).

$$C(x) = T^{-1} \sum_{t=1}^{T} C_t(x) \tag{14}$$

i) ***Gradient Boosting (GB)***. Neste trabalho, a técnica *GB* segue conforme Wang, Ma e Yang (2014), Wang, Hao et al. (2011) e Wang, Ma, Huang et al. (2012) e Heo e Yang (2014). Semelhante ao método apresentado por Wang, Ma e Yang (2014), Wang, Hao et al. (2011) e Wang, Ma, Huang et al. (2012), a estrutura matemática é definida, formalmente, nas equações (15, 16, 17, 18).

$$\varepsilon_i = P_{i \sim D_i}[h_t(x_i \neq y_i)] \tag{15}$$

$$\alpha_t = \frac{1}{2} ln \frac{1 - \varepsilon_t}{\varepsilon_t} \tag{16}$$

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} \; se \; h_t(x_i) = y_i \\ e^{\alpha_t} \; se \; h_t(x_i) \neq y_i \end{cases} \tag{17}$$

O resultado final de classificação é obtido por:

$$H(x) = sign \sum_{t=1}^{T} a_t h_t(x) \tag{18}$$

j) ***Neural networks (NN)***. Neste trabalho, a técnica *NN* segue conforme Tsai, Hsu e Yen (2014) e Zhao et al. (2015). Idêntico ao modelo apresentado por Tsai, Hsu e Yen (2014), a estrutura matemática é definida, formalmente, nas equações (19, 20, 21, 22, 23).

$$net_j^h = \sum_{i=1}^{N} W_{ji} x_i \text{ e } y_i = f(net_j^h) \tag{19}$$

$$net_k^o = \sum_{j=1}^{J+1} V_{kj} y_i \text{ e } o_k = f(net_k^o) \tag{20}$$

$$f(net) = \frac{1}{1 + e^{-\lambda \cdot net}} \tag{21}$$

$$V_{kj}(t+1) = v_{kj}(t) + c\lambda(d_k - o_k)o_k(1 - o_k)y_j(t) \tag{22}$$

$$W_{ji}(t+1) = w_{ji}(t) + c\lambda^2 y_j(1 - y_j)x_i(t) \left(\sum_{K=1}^{K} (d_k - o_k)o_k(1 - o_k)v_{kj} \right) \tag{23}$$

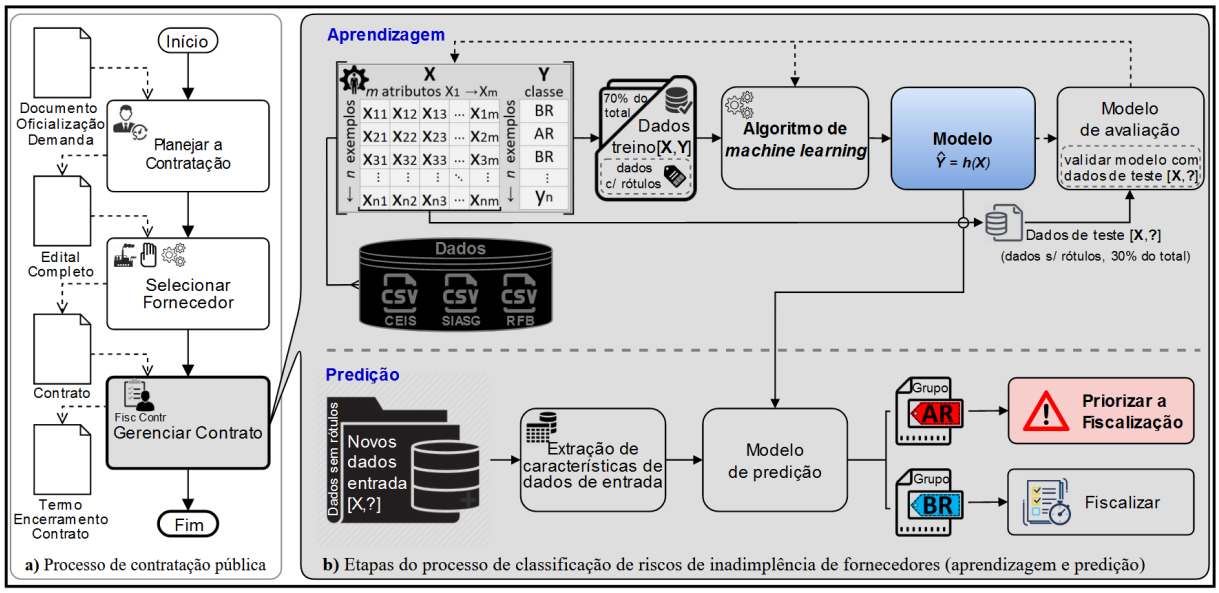
3.3 Estrutura de priorização baseada em riscos

Sistemas baseados em métodos de análise de dados vêm sendo desenvolvidos com a finalidade previnir irregularidades em compras do setor público (Seção 2.3.2). Nesta pesquisa, uma estrutura de priorização, baseada em riscos, foi desenvolvida para proporcionar mais eficiência à atividade de fiscalização de contrato (Fisc Contr), tarefa crítica da etapa *Gerenciar Contrato* (Figura 6). A estrutura funciona como sistema de alerta precoce, que prioriza, com base no risco de inadimplência de cada fornecedor avaliado, a execução da atividade de fiscalização de contrato.

Métodos que possibilitam a tomada de decisão mais fundamentada em evidências, como técnicas de *machine learning*, podem ser empregados na avaliação do risco de inadimplência corporativo, sendo importantes instrumentos de suporte à gestão (Seção 2.2.1). Neste trabalho, tais métodos são utilizados para classificar fornecedores do governo federal brasileiro, com base em risco de inadimplência, sendo úteis na triagem inicial de contratos vinculados a esses fornecedores e na alocação adequada de fiscais de contratos. A estrutura de priorização de fiscalização de contrato, proposta nesta pesquisa, é apresentada na Figura 6.

A Figura 6a mostra as tês etapas básicas do processo de contratação pública e, em destaque, exibe a etapa *Gerenciar Contrato*, foco desta pesquisa. As irregularidades na contratação pública assumem formas diversas e, segundo Berru et al. (2019), Dávid-Barrett e Fazekas (2020) e Fazekas e Kocsis (2017), podem ocorrer em qualquer uma dessas três etapas. Com

Figura 6: Estrutura de priorização de fiscalização de contrato (EPFC).



Fonte: Elaboração própria.

isso, para previnir irregularidades e ilegalidades nessas etapas, mecanismos de fiscalização e controle, como o *ALICE* [15], Neves, Silva e Carvalho (2019), que identifica inconsistências em editais publicados no *ComprasNet/SIASG*, etapa *Selecionar Fornecedor* (Figura 6a), precisam ser desenvolvidos.

Nesta pesquisa, conforme ilustrado na Figura 6b, o método utilizado para classificar os riscos de inadimplência de fornecedores emprega as mesmas etapas do processo de aprendizagem apresentado na Seção 2.1.2, Figura 3, quais sejam: *i)* realizar o pré-processamento dos dados (dados coletados do *CEIS*, *SIASG* e *RFB*); *ii)* treinar o algoritmo com dados de treino (70% do total, dados rotulados [X,Y]); *iii)* validar o modelo com dados de teste (30% do total, dados não rotulados [X,?]); e *iv)* realizar a classificação de novos dados de entrada não rotulados [X, ?], etapa seguinte.

3.4 Procedimentos de avaliação e validação

Várias métricas tradicionais, como acurácia, taxa de precisão, coeficiente de *Gini*, teste *KS*, *BRIER Score*, erro quadrático médio, medida-F (*F-measure/F1 score*), área sob a curva *ROC*, erro tipo I e erro tipo II, podem ser utilizadas para avaliar o desempenho de modelos de pontuação de crédito, Baesens et al. (2003), Dastile, Celik e Potsane (2020), Engelmann, Hayden e Tasche (2003), Fawcett (2006), Marqués, Garcıa e Sánchez (2012), De Menezes et al.

[15] Analisador de Licitações, Contratos e Editais (ALICE)

(2017), Sun e Wang (2005), Tsukahara et al. (2016), Castro Vieira et al. (2019) e Wang, Ma e Yang (2014), Wang, Hao et al. (2011) e Wang, Ma, Huang et al. (2012).

De acordo com Akkoç (2012), Altman (1968), Chen, Ma e Ma (2009), Feng et al. (2018), Garcıa, Marqués e Sánchez (2012), Louzada, Ara e Fernandes (2016), Lu e Chen (2009), Tharwat (2020) e Zeng (2020), para modelos de classificação binária, a maioria dessas métricas podem ter como origem uma matriz de confusão, como a apresentada na Tabela 5. Os critérios de avaliação de desempenho utilizados nesta pesquisa, apresentados de forma resumida na Tabela 6, foram a *accuracy - ACC* (acurácia), *type I error* (erro tipo I), *type II error* (erro tipo II), *precision* (precisão), *recall* (sensibilidade), *F-measure* (medida-F) e a *area under the ROC curve - AUC* (área sob a curva de característica operacional do receptor - ROC).

3.4.1 Matriz de confusão

Matriz de confusão, também chamada de matriz de erro ou tabela de contigência, Tharwat (2020), é, segundo Fawcett (2006), Garcıa, Marqués e Sánchez (2012), Louzada, Ara e Fernandes (2016), Marqués, Garcıa e Sánchez (2012), Tharwat (2020) e Zeng (2020), uma tabela de frequência bidimensional, $C = (c_{ij})_{2x2}$, usada para medir a precisão de um modelo comparando o resultado preditivo do modelo com os valores reais do conjunto de dados para uma determinada pontuação de corte. Neste trabalho, os valores reais estão organizados em duas classes, como mostrado na Tabela 5: classe $\{0\}$, equivalente à soma dos elementos c_{11} e c_{12} (VN + FP); e classe $\{1\}$, equivalente à soma dos elementos c_{21} e c_{22} (FN + VP).

Tabela 5: Matriz de confusão.

		Previsão	
		$\{0\}$	$\{1\}$
Valor real	$\{0\}$	Verdadeiro negativo (VN)	Falso positivo (FP)
	$\{1\}$	Falso negativo (FN)	Verdadeiro positivo (VP)

Fonte: Modificado de Feng et al. (2018, p. 142).

Métricas derivadas da matriz de confusão, dentre as quais, acurácia, *recall* (sensibilidade), taxa de precisão, erro tipo I, erro tipo II e medida-F, foram amplamente utilizadas em modelos de *credit scoring* (pontuação de crédito). Essas métricas são definidas da seguinte forma:

Acurácia (ACC). Também chamada de pontuação de acertos, é a proporção de previ-

sões corretas do modelo ao classificar os casos nas classes $\{0\}$ ou $\{1\}$ (LOUZADA; ARA; FERNANDES, 2016, p. 7). De acordo com Baesens et al. (2003), Marqués, Garcıa e Sánchez (2012) e Garcıa, Marqués e Sánchez (2012), não se pode utilizar apenas a *ACC* como métrica para avaliar o desempenho de um classificador, pois a *ACC* assume custos iguais de classificação incorreta para previsões de falsos positivos e falsos negativos. A *Acurácia (ACC)* é definida, formalmente, na equação (24).

$$\textit{Acurácia (ACC)} = \frac{VP + VN}{VP + FP + FN + VN} \tag{24}$$

Sensibilidade* e *Erro tipo II. *i) sensibilidade* ou taxa de verdadeiros positivos (TVP) mede a proporção de exemplos positivos que são previstos como positivos, Baesens et al. (2003), ou seja, é a fração de casos, entre todos os casos pertencentes à classe $\{1\}$, que a técnica classificou corretamente para a classe $\{1\}$, Louzada, Ara e Fernandes (2016, p. 7); e *ii) erro tipo II* ou taxa de falsos negativos (TFN) é a fração de casos da classe $\{1\}$ classificados erroneamente como pertencente à classe $\{0\}$ (LOUZADA; ARA; FERNANDES, 2016). A relação entre sensibilidade e erro tipo II é mostrada na equação (25).

$$Sensibilidade\ (\textit{recall}) = TVP = 1 - Erro\ tipo\ II = \frac{VP}{VP + FN} \tag{25}$$

Especificidade* e *Erro tipo I. *i) especificidade* ou taxa de verdadeiros negativos (TVN) mede a proporção de exemplos negativos que são previstos como negativos, Baesens et al. (2003), ou seja, é a razão de observações, entre todos os casos pertencentes à classe $\{0\}$, classificada corretamente pelo modelo na classe $\{0\}$, Louzada, Ara e Fernandes (2016, p. 7); e *ii) erro tipo I* ou taxa de falsos positivos (TFP) é a fração de casos da classe $\{0\}$ classificada erroneamente como pertencente à classe $\{1\}$ (LOUZADA; ARA; FERNANDES, 2016). A relação entre especificidade e erro tipo I é mostrada na equação (26).

$$Especificidade = TVN = 1 - Erro\ tipo\ I = \frac{VN}{VN + FP} \tag{26}$$

Precisão (P). Representa a proporção de amostras positivas que foram classificadas corretamente em relação ao número total de amostras preditas positivas, Tharwat (2020, p. 4), ou seja, é a razão entre o número de verdadeiros positivos com o número total de instâncias rotuladas como positivas (ZHU et al., 2019). Mede o percentual de casos, realmente, da classe $\{1\}$, dentre todos os casos previstos como $\{1\}$. A *Precisão (P)* é definida, formalmente, na equação

(27).

$$Precisão\ (P) = \frac{VP}{VP + FP} \tag{27}$$

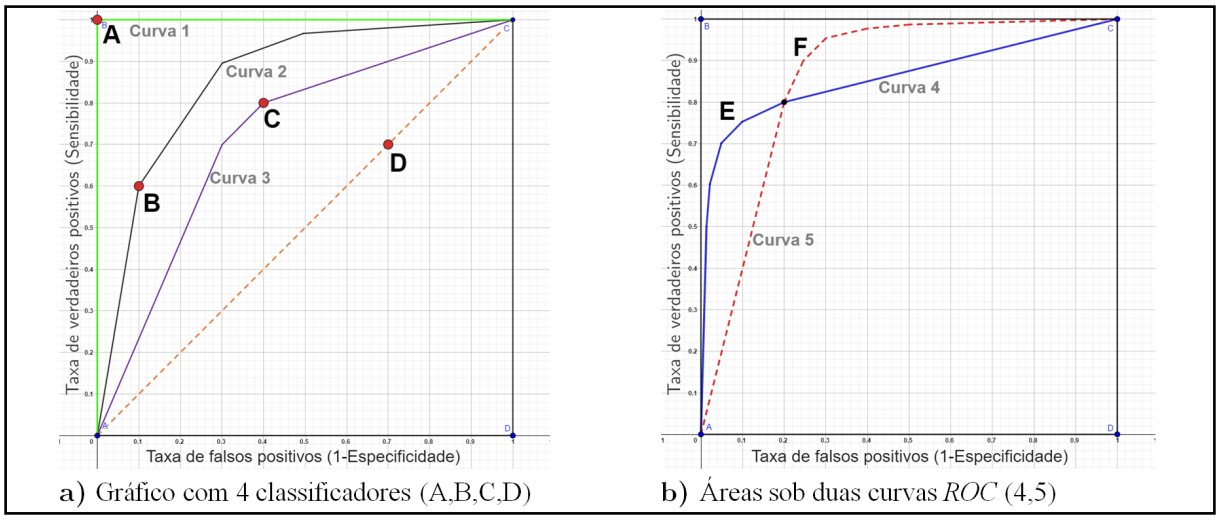
Medida F. A *medida F* (*F-Measure* ou *F1-Score*) combina *precisão (P)* e *sensibilidade (TVP)* em uma única medida de avaliação. Segundo Dastile, Celik e Potsane (2020), Tharwat (2020) e Zhu et al. (2019), é calculada pela média harmônica de *P* e *TVP*. Segundo Tharwat (op. cit.) e Zhu et al. (op. cit.), altos valores da *medida F* indicam alto desempenho de classificação. A *Medida F* é definida, formalmente, na equação (28).

$$Medida\ F\ (F1\ Score) = 2 \times \left(\frac{Sensibilidade \times Precisão}{Precisão + Sensibilidade}\right) \tag{28}$$

3.4.2 Curva *ROC* e área sob a curva *ROC*

Receiver Operating Characteristic (ROC). A curva de características operacionais do receptor (ROC) é uma ilustração gráfica bidimensional da sensibilidade ou taxa de verdadeiros positivos - TVP, no eixo Y e da (1-especificidade) ou taxa de falsos positivos - TFP, no eixo X para vários valores no limiar de classificação (FAWCETT, 2006), (BAESENS et al., 2003) e (DE MENEZES et al., 2017). De acordo com Fawcett (op. cit., p. 1), um gráfico *ROC* tem como finalidade a visualização, a organização e a seleção de classificadores com base em seus desempenhos. A visualização de possíveis desempenhos de um classificador no gráfico *ROC* é mostrada na Figura 7.

Figura 7: Gráficos *ROC* com possíveis desempenhos de um classificador.



Fonte: Elaboração própria.

Na Figura 7a, o espaço *ROC* possui quatro classificadores discretos (*A*, *B*, *C* e *D*) com

diferentes *AUC*: $A_{(0,0;1,0)}$ com *AUC* igual a 1,0; $B_{(0,1;0,6)}$ e $C_{(0,4;0,8)}$ com *AUC* maior que 0,5; e $D_{(0,7;0,7)}$ com *AUC* igual a 0,5. Em síntese, a Figura 7a (curvas 1, 2 e 3) mostra que *A* possui capacidade perfeita de classificação (é chamado de ponto operacional ideal), ou seja, todas as amostras positivas e negativas são classificadas corretamente (THARWAT, 2020); *B* possui capacidade de classificação mais estável que *C*; e *D* possui desempenho aleatório (Y=X).

De acordo com Tharwat (2020), classificadores com curvas *ROC* diferentes podem ter a mesma taxa *AUC*. Na Figura 7b, as áreas sob as curvas *ROC*, 4 e 5, são utilizadas para avaliar o desempenho de dois classificadores, *E* e *F*. Essas curvas mostram que: *i)* no intervalo (0,0 < x < 0,2), *E* possui desempenho melhor que *F*, pois obtém ótimas taxas de *recall* com baixas taxas de falsos positivos; *ii)* na *TFP* igual a 0,2 (x = 0,2), *E* e *F* possuem desempenhos equivalentes (*precisão* igual *recall*); e *iii)* no intervalo (0,2 < x < 1,0), *F* possui elevada taxa de precisão, portanto detém um desempenho melhor que *E*.

A literatura mostra que modelos com elevada taxa de *recall* (sensibilidade) ou com elevada taxa de precisão, Figura 7b, podem ser aplicados em diferentes cenários (FAWCETT, 2006). Se a aplicação for para detectar transações fraudulentas, auxiliar sistema de segurança que não adimite falha ou auxiliar sistema de diagnóstico na detecção precoce de doença grave é mais adequado utilizar um modelo com alta sensibilidade, mesmo ao custo da baixa precisão. No entanto, se a aplicação do modelo for auxiliar sistemas de justiça, no julgamento de indivíduos, é preferível que o modelo tenha alta precisão.

Area under the ROC curve (AUC). A área sob a curva *ROC* (*AUC*) mede o poder discriminatório do modelo preditivo, ou seja, o sucesso do modelo em classificar corretamente *VP* e *VN*, De Menezes et al. (2017), sem levar em consideração o custo de distribuição ou classificação incorreta de classe (BAESENS et al., 2003) e (FAWCETT, 2006). De acordo com Sun e Wang (2005), quando a *AUC* for 0,5, significa que os casos das classes {0} e {1} são classificados aleatoriamente; quando a *AUC* for 1, significa que o modelo classificou com precisão os casos das duas classes {0} e {1}.

Portanto, de acordo com Sun e Wang (2005) e Engelmann, Hayden e Tasche (2003), para qualquer modelo de classificação razoável na prática, a *AUC* tem variações entre 0,5 e 1,0 e quanto maior for a área abaixo da curva *ROC* (área sob a curva ou *AUC*), mais preciso será o modelo de classificação. De acordo com Engelmann, Hayden e Tasche (op. cit.) e Tsukahara et al. (2016), a área sob a curva *ROC* (*AUROC*), considerando a situação ideal (*AUC=1*),

$HR(C) = \frac{H(C)}{N_D}$ e $FAR(C) = \frac{F(C)}{N_{ND}}$, é formalmente definida segundo a equação (29).

$$AUROC = \int_0^1 HR(FAR)d(FAR) \tag{29}$$

Na equação acima, a *hit rate - HR (C)*, taxa de acerto, e a *false alarm rate - FAR (C)*, taxa de falso alarme, ou seja, sensibilidade $\left(TVP = \frac{VP}{VP+FP}\right)$ e erro tipo I $\left(TFP = \frac{FP}{FP+VN}\right)$, respectivamente, são componentes da curva *ROC*; *C* é o ponto de corte (*cutoff score*) utilizado no modelo; *H (C)* é o número de inadimplentes preditos corretamente com o valor de corte *C*; N_D é o número total de inadimplentes na amostra; *F (C)* é o número de falsos alarmes, ou seja, o número de adimplentes que foram classificados incorretamente como inadimplentes usando o valor de corte *C*; e N_{ND} é o número total de adimplentes na amostra.

3.4.3 Custos de classificação incorreta associados aos erros tipo I e tipo II

Para efeito desta pesquisa, taxa de erro tipo I ou alarme falso é a proporção de elementos da classe $\{0\}$ (*Gp BR*) categorizados como elementos da classe $\{1\}$ (*Gp AR*), ou seja, fornecedores adimplentes classificados como inadimplentes e taxa de erro tipo II ou falha é a proporção de elementos da classe $\{1\}$ (*Gp AR*) categorizados como elementos da classe $\{0\}$ (*Gp BR*), ou seja, fornecedores inadimplentes classificados como adimplentes.

Os custos de classificação incorreta associados aos erros tipo I e tipo II devem ser levados em consideração, pois, de acordo com Baesens et al. (2003), Chawla et al. (2002) e Fawcett e Provost (1997), na maioria dos problemas da vida real, um tipo de erro de classificação pode ser muito mais caro que o outro. Tomando-se como base Lu e Chen (2009), Chen, Ma e Ma (2009), Yeh, Chi e Lin (2014), Yap, Ong e Husain (2011) e Zhu et al. (2019), ao avaliar o desempenho dos modelos nesta pesquisa, considerou-se a taxa de erro tipo II mais importante, pois o custo econômico de classificar um fornecedor inadimplente (*Gp AR*) como adimplente (*Gp BR*) é maior que o da classificação reversa.

Em outras palavras, segundo Zhu et al. (2019), Marqués, Garcıa e Sánchez (2012) e Garcıa, Marqués e Sánchez (2012), erro tipo I indica a taxa de erro de previsão do modelo, que categoriza incorretamente adimplentes como inadimplentes. Com isso, os fornecedores idôneos (*Gp BR*), classificados incorretamente, são recusados e o governo tem um custo de oportunidade

causado pela perda de fornecedores de baixo risco. Já a taxa de erro do tipo II, por outro lado, segundo esses autores, apresenta a taxa de erro de previsão do modelo, que categoriza incorretamente inadimplentes como adimplentes. Neste caso, os fornecedores inidôneos (*Gp AR*) classificados incorretamente se tornarão inadimplentes, com isso, se a política de contratação for muito generosa, ela estará propensa a altos riscos de inadimplemento contratual.

Por esses motivos, de acordo com Baesens et al. (2003), Marqués, Garcıa e Sánchez (2012), Garcıa, Marqués e Sánchez (2012) e Feng et al. (2018), é importante medir o erro em cada classe individual usando o erro tipo I ($TFP = \frac{FP}{VN+FP}$) e o erro tipo II ($TFN = \frac{FN}{VP+FN}$). Neste trabalho, assim como em Lu e Chen (2009), Chen, Ma e Ma (2009), Yeh, Chi e Lin (2014), Yap, Ong e Husain (2011) e Zhu et al. (2019), o custo de classificação incorreta associado ao erro do tipo II é muito mais alto do que aquele associado ao erro do tipo I.

Tabela 6: Métricas utilizadas na avaliação dos modelos.

Métricas	**Fórmulas**	**Foco da avaliação**	**Referências***
Acurácia (*ACC*)	$\frac{VP+VN}{VP+VN+FP+FN}$	Eficiência geral de um classificador	A
Sensibilidade (*TVP*)	$\frac{VP}{VP+FN}$	Eficiência para categorizar rótulos positivos	B
Especificidade (*TVN*)	$\frac{VN}{VN+FP}$	Eficiência para categorizar rótulos negativos	C
Precisão (*P*)	$\frac{VP}{VP+FP}$	Razão entre os *VP* e o total de positivos	D
Medida F (*F1-Score*)	$\frac{2\times P\times TVP}{P+TVP}$	Média harmônica entre *Precisão* e *Recall*	E
AUROC (*AUC*)	$\int_0^1 TVP(TFP)d(TFP)$	Grau ou medida de separabilidade de classes	F

*

A Baesens et al. (2003), Louzada, Ara e Fernandes (2016), Dastile, Celik e Potsane (2020) e Feng et al. (2018)
B Baesens et al. (2003), Louzada, Ara e Fernandes (2016), De Menezes et al. (2017) e Barboza, Kimura e Altman (2017)
C Baesens et al. (2003), Louzada, Ara e Fernandes (2016), De Menezes et al. (2017) e Barboza, Kimura e Altman (2017)
D Tharwat (2020), Zhu et al. (2019), Zeng (2020) e Feng et al. (2018)
E Tharwat (2020), Zhu et al. (2019) e Dastile, Celik e Potsane (2020)
F Fawcett (2006), Sun e Wang (2005), Engelmann, Hayden e Tasche (2003) e Tsukahara et al. (2016)

Fonte: Elaboração própria.

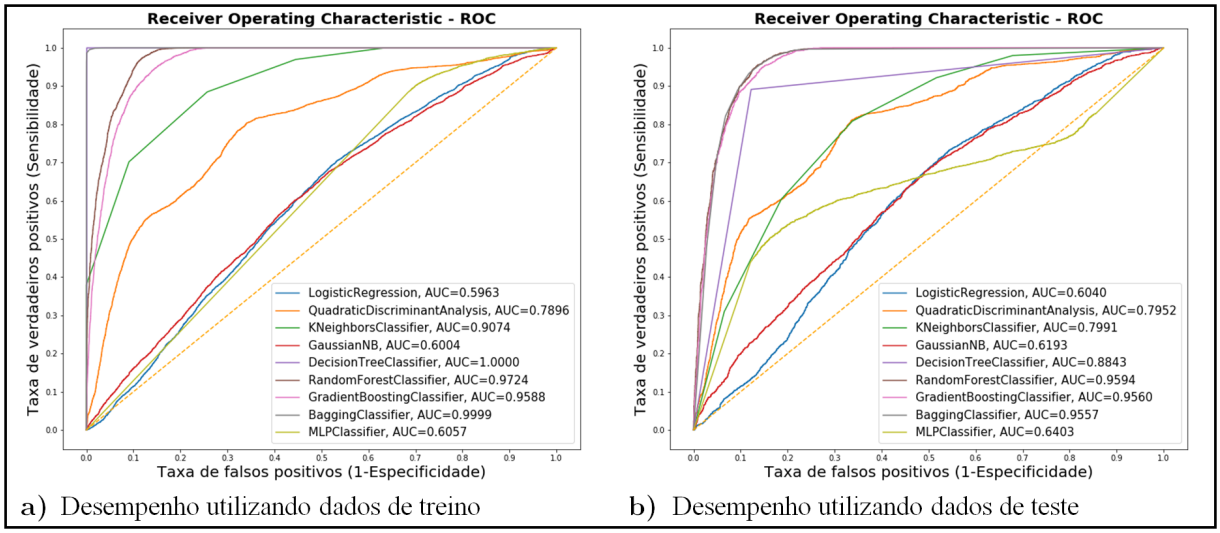
4 Resultados

Para identificar o modelo de classificação com o melhor poder preditivo, obtido a partir de uma das dez técnicas apresentadas na Seção 3.2, estatística tradicional (*DA* e *RL*) e *machine learning* (*KNN*, *BN*, *SVM*, *DT*, *RF*, *BG*, *GB* e *NN*), uma análise comparativa de desempenho entre essas técnicas foi realizada. Nessa análise, os algoritmos *DA* e *LR* são utilizados como parâmetro de comparação na avaliação do desempenho das técnicas empregadas. O processamento completo dos dados, com a ativação dos modelos, é apresentado nos apêndices B e C.

4.1 Análise dos resultados obtidos

Uma análise crítica com a evolução do desempenho de cada modelo avaliado é mostrada nos gráficos *ROC* da Figura 8. As curvas *ROC* ilustradas nos gráficos possibilitam a vizualização e a análise detalhada do desempenho dos modelos, na amostra de treino e na amostra de teste. A área sob a curva *ROC* (*AUC*) também foi utilizada como parâmetro de avaliação de desempenho: para um modelo ser viável, a *AUC* deve variar entre 0,5 e 1,0 (quanto mais próximo de 1, mais preciso é o modelo de classificação).

Figura 8: Gráficos *ROC* com o desempenho dos modelos analisados.



a) Desempenho utilizando dados de treino b) Desempenho utilizando dados de teste

Fonte: Elaboração própria.

Os resultados, apresentados na Figura 8, mostram superioridades significativas dos modelos automáticos, *RF*, *GB*, *BG* e *DT*, sobre os modelos tradicionais, *DA* e *RL*: *i)* na amostra de treino, Figura 8a, as taxas de *AUC* dos modelos *RF* (97,24%), *GB* (95,88%) e *BG* (99,99%) são superiores às taxas de *AUC* dos modelos *DA* (78,96%) e *LR* (59,63%); e *ii)* na amostra de

teste, Figura 8b, as taxas de *AUC* dos modelos *RF* (95,94%), *GB* (95,60%) e *BG* (95,57%) são, também, superiores às taxas de *AUC* dos modelos *DA* (79,52%) e *LR* (60,40%). Os resultados completos, nas amostras de treino e de teste, contendo todos os indicadores de desempenho analisados, são apresentados na Tabela 7.

A Tabela 7 reune as técnicas utilizadas e os resultados dos seguintes indicadores de desempenho: *i)* verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN); *ii)* erro tipo I $\left(TFP = \frac{FP}{VN+FP}\right)$, fornecedores adimplentes que foram classificados como inadimplentes; *iii)* erro tipo II $\left(TFN = \frac{FN}{VP+FN}\right)$, fornecedores inadimplentes que foram classificados como adimplentes; *iv)* precisão total estimada, acurácia $\left(ACC = \frac{VP+VN}{VP+VN+FP+FN}\right)$; e *v)* medida de separabilidade de classes, área sob a curva *ROC* $\left(AUC = \int_0^1 TVP(TFP)d(TFP)\right)$.

Tabela 7: Desempenho dos modelos nas amostras de treino e de teste.

a) Amostra de treino								
Técnica	VP	VN	FP	FN	Erro tipo I (%)	Erro tipo II (%)	ACC (%)	AUC (%)
DA	6593	3573	4104	959	53,46	12,70	66,75	78,96
LR	0	7677	0	7552	0	100,00	50,41	59,63
KNN	6438	5337	2340	1114	30,48	14,75	77,32	90,74
BN	7101	974	6703	451	87,31	5,97	53,02	60,04
SVM	1064	6858	819	6488	10,67	85,91	52,02	53,29
DT	7552	7677	0	0	0	0	100,00	100,00
RF	7367	6737	940	185	12,24	2,45	92,61	97,24
BG	7515	7629	48	37	0,62	0,49	99,44	99,99
GB	7131	6569	1108	421	14,43	5,57	89,96	95,88
NN	7289	1954	5723	263	74,55	3,48	60,69	60,57
b) Amostra de teste								
Técnica	VP	VN	FP	FN	Erro tipo I (%)	Erro tipo II (%)	ACC (%)	AUC (%)
DA	**2930**	**1486**	**1715**	**396**	**53,58**	**11,91**	**67,66**	**79,52**
LR	0	3201	0	3326	0	100,00	49,04	60,40
KNN	**2677**	**2034**	**1167**	**649**	**36,46**	**19,51**	**72,18**	**79,91**
BN	3141	448	2753	185	86,00	5,56	54,99	61,93
SVM	445	2876	325	2881	10,15	86,62	50,88	53,29
DT	**2964**	**2809**	**392**	**362**	**12,25**	**10,88**	**88,45**	**88,43**
RF	**3179**	**2757**	**444**	**147**	**13,87**	**4,42**	**90,95**	**95,94**
BG	**3105**	**2823**	**378**	**221**	**11,81**	**6,64**	**90,82**	**95,57**
GB	**3138**	**2750**	**451**	**188**	**14,09**	**5,65**	**90,21**	**95,60**
NN	2646	619	2582	680	80,66	20,44	50,02	64,03

Fonte: Elaboração própria.

Em destaque na amostra de teste, Tabela 7b, resultados relevantes com *AUC* entre 0,5 e 1,0 são obtidos pelos algoritmos *RF*, *GB*, *BG*, *DT*, *DA* e *KNN*. Esses resultados mostram vantagens significativas, em termos de erro tipo I, erro tipo II, *ACC* e *AUC*, dos modelos *ensemble learning* (aprendizagem em conjunto) *RF*, *GB* e *BG* sobre o modelo tradicional *DA* e sobre

os demais modelos. Os modelos *RF*, *GB* e *BG* geram alto poder de discriminação de classes, nas amostras de treino e de teste, por isso, esses classificadores, sobretudo o *RF*, apresentam-se como os modelos mais adequados à estrutura de priorização (6), proposta nesta pesquisa, e como uma das possíveis soluções ao problema em análise.

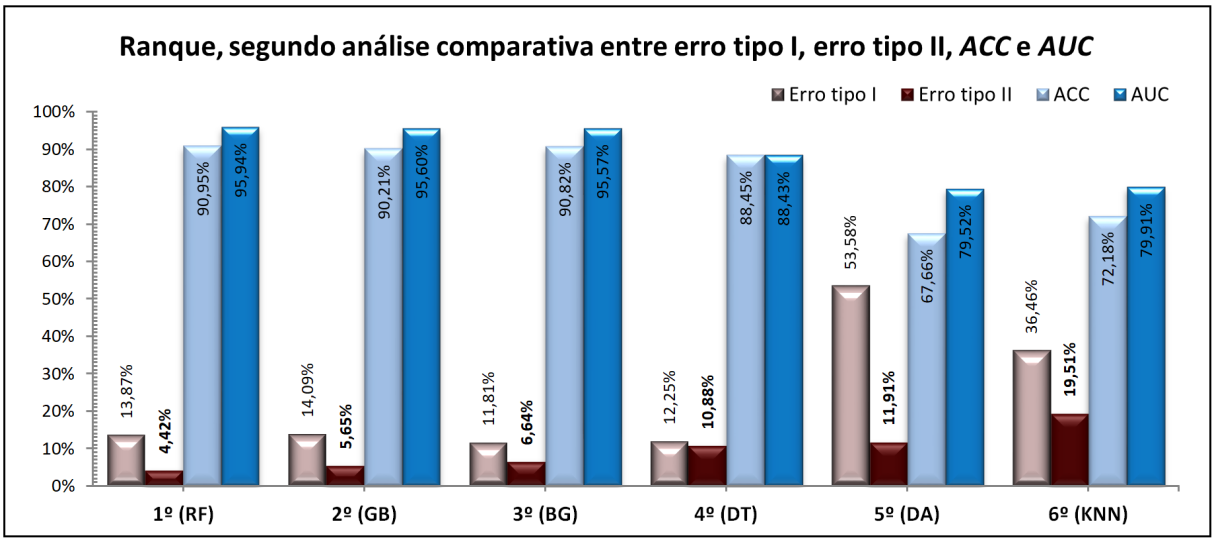
Conforme mostrado na Tabela 7b, o modelo *RF* retorna os melhores resultados, com baixas taxas de erro tipo I (13,87%) e de erro tipo II (4,42%) e elevadas taxas de *ACC* (90,95%) e *AUC* (95,94%). Os modelos *GB* e *BG* retornam também ótimos resultados e produzem, respectivamente, o segundo e o terceiro melhor resultado, quais sejam: erro tipo I (*GB*, 14,00%; e *BG*, 11,81%); erro tipo II (*GB*, 5,65%; e *BG*, 6,64%); *ACC* (*GB*, 90,21%; e *BG*, 90,82%); e *AUC* (*GB*, 95,60%; e *BG*, 95,57%). Com isso, *RF*, *GB* e *BG* são, nesta ordem, os três modelos com melhor desempenho, entre todos os modelos avaliados.

Ainda sobre os modelos *GB* e *BG*, embora esses algoritmos apresentem a terceira e a segunda melhor taxa de *ACC* (90,21% e 90,82%, respectivamente), suas respectivas taxas de erro tipo II, de 5,65% e 6,64%, os levam a ocupar posições opostas às obtidas sem levar em consideração os custos de classificação incorreta. Nesta pesquisa, o custo de classificação incorreta associado ao erro tipo II é muito mais alto do que aquele associado ao erro tipo I (Seção 3.4.3). Por esse motivo, os modelos *GB* e *BG*, são classificados, respectivamente, na segunda e na terceira posição.

Sobre o modelo *DT*, embora o algoritmo apresente desempenho elevado na amostra de treino, *AUC*=100%, na amostra de teste retorna resultados moderados, por exemplo: erro tipo I (12,25%); erro tipo II (10,88%); *ACC* (88,45%); e *AUC* (88,43%). Com isso, esse modelo passa a ocupar a quarta posição. Outros resultados relevantes na amostra de teste surgem logo em seguida, com o modelo *DA* (*ACC*=67,66%, na quinta posição) e com o modelo *KNN* (*ACC*=72,18%, na sexta posição).

Ao se analisar apenas as taxas de *ACC* dos modelos *DA* (67,66%) e *KNN* (72,18%), diríamos que esses modelos ocupam, respectivamente, a sexta e a quinta posição. Contudo, ao se priorizar a redução do custo de classificação incorreta, associado ao erro tipo II (*DA*, 11,91%; e *KNN*, 19,51%), a conclusão que se chega é que, de fato, os modelos *DA* e *KNN* ocupam, respectivamente, a quinta e a sexta posição. Os classificadores com desempenhos relevantes, avaliados segundo análise comparativa entre as medidas de desempenho erro tipo I, erro tipo II, *ACC* e *AUC*, são apresentados na Figura 9.

Figura 9: Performance dos classificadores *RF*, *GB*, *BG*, *DT*, *DA* e *KNN*.



Fonte: Elaboração própria.

4.1.1 Custos de classificação incorreta dos modelos *GB*, *BG*, *DA* e *KNN*

Nesta pesquisa, ao se avaliar a capacidade geral de classificação dos modelos, optou-se pela redução do custo de classificação gerado com o erro tipo II, em detrimento do custo proveniente do erro tipo I. Sobre os erros de classificação, como visto na Seção 3.4.3, o custo gerado com o erro tipo II é muito mais alto do que aquele decorrente do erro do tipo I. A avaliação de desempenho dos modelos *GB*, *BG*, *DA* e *KNN*, com base nas taxas de erro tipo I e tipo II, é apresentada na Tabela 8.

Tabela 8: Taxas de erro tipo I e tipo II dos modelos *GB*, *BG*, *DA* e *KNN*.

Modelo	**Avaliação de desempenho**	
	Erro tipo I	**Erro tipo II**
Gradient boosting (GB)	451 (14,09%)	188 (5,65%)
Bagging (BG)	378 (11,81%)	221 (6,64%)
Discriminant analysis (DA)	1715 (53,58%)	396 (11,91%)
K-nearest neighbors (KNN)	1203 (36,46%)	654 (19,51%)

Fonte: Elaboração própria.

A Tabela 8 reune os quatro algoritmos em discussão, ordenados de acordo com a capacidade geral de classificação de cada modelo e segundo as menores taxas de erro tipo II. Os modelos *GB* e *BG* retornam baixo número de falsos positivos e falsos negativos (*GB* $[451_{FP} + 188_{FN} = 639]$; e *BG* $[378_{FP} + 221_{FN} = 599]$). A análise dos erros de classificação mostra que, mesmo apresentando erro geral (*GB* [639]) maior do que o modelo *BG* (599), a técnica *GB* é mais adequada do que o modelo *BG*, pois retorna menor taxa de erro tipo II (*GB* [5,65%,

FN=188]) do que o modelo *BG* (6,64%, *FN=221*).

De modo análogo à situação anterior, o modelo *DA* retorna erro geral maior do que o modelo *KNN*: *DA* ($1715_{FP} + 396_{FN} = 2111$); e *KNN* ($1203_{FP} + 654_{FN} = 1857$). A análise das taxas de erro de classificação é, também, decisiva na comparação desses modelos. Mesmo apresentando erro geral (*DA* [2111]) maior do que o modelo *KNN* (1857), a técnica *DA* é mais adequada do que o modelo *KNN*, pois retorna menor taxa de erro tipo II (11,91%, *FN=396*) do que o modelo *KNN* (19,51%, *FN=654*).

De acordo com os resultados da Tabela 8, a análise das taxas de erro de classificação é de fundamental imporância, para a identificação adequada da capacidade de classificação dos modelos e para a ordenação correta das técnicas, no ranque dos classificadores de melhor desempenho, apresentado na Figura 9. Portanto, os modelos com desempenhos relevantes ocupam as seis primeiras posições desse ranque, segundo a capacidade de classificação de cada modelo, na seguinte ordem: 1º (*RF*); 2º (*GB*); 3º (*BG*); 4º (*DT*); 5º (*DA*); e 6º (*KNN*).

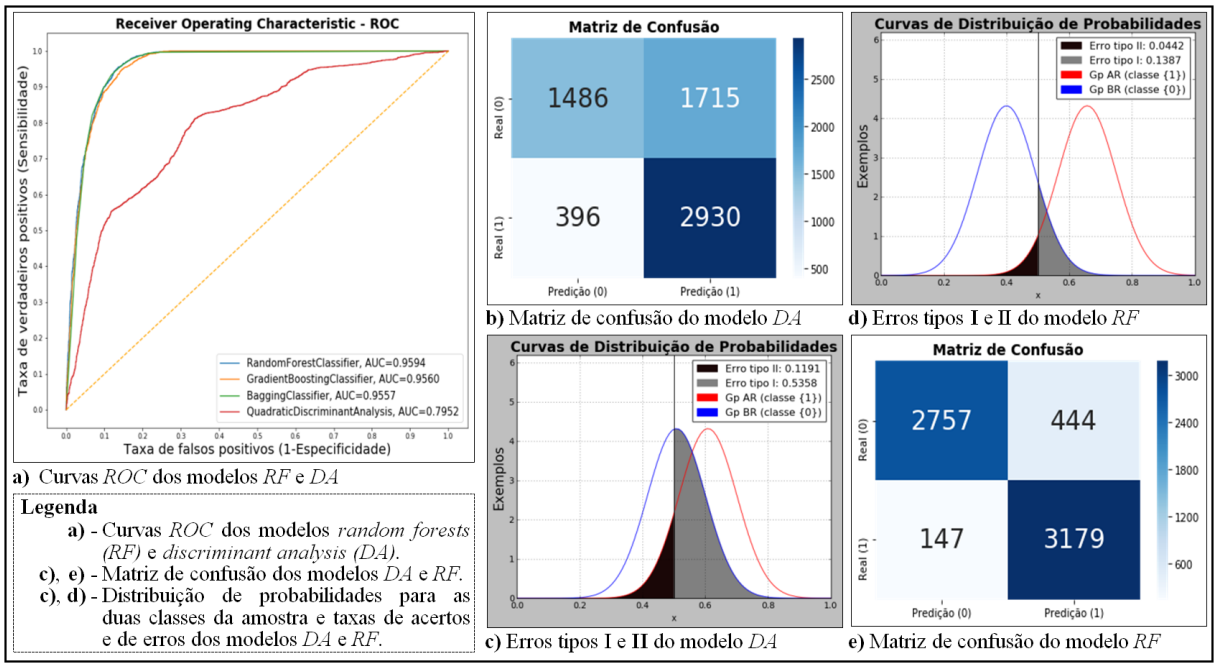
4.1.2 Análise comparativa entre os modelos *RF* e *DA*

A superioridade das técnicas de *machine learning*, sobretudo, das abordagens baseadas em métodos *ensemble*, em termos de precisão e erro de predição, é reconhecida na literatura de análise de crédito (Seção 2.2.2). Nesta pesquisa, métodos de classificação são avaliados e os seis melhores resultados são apresentados, de forma ordenada, segundo a capacidade de classificação de cada modelo, do 1º ao 6º lugar (Figura 9). Esses resultados mostram vantagens significativas do modelo *random forests – RF* (método *ensemble*) sobre o modelo *discriminant analysis – DA* (método tradicional), levando-os, respectivamente, ao 1º e ao 5º lugar.

Conforme afirmado anteriormente, no geral, os modelos automáticos apresentam melhor desempenho do que os modelos estatísticos tradicionais. Em comparação com as abordagens clássicas, baseadas em programação estatística tradicional que utiliza conhecimento prévio específico escrito por um programador, os modelos de *machine learning* extraem automaticamente informações dos exemplos, em análise, para realizar estimativas futuras fora da amostra. Informações úteis, com detalhes da análise comparativa, entre os modelos *RF* e *DA*, são apresentadas na Figura 10.

Resultados de análise na amostra de teste, detalhados na Figura 10, mostram superioridades significativas do modelo *RF* sobre o modelo *DA*. Nessa análise, conforme ilustrado

Figura 10: Curvas *ROC* e erros de classificação dos modelos *DA* e *RF*.



Fonte: Elaboração própria.

nas Figuras 10a, 10d e 10e, o modelo *RF* retorna os melhores resultados, com elevada taxa de *AUC* (95,94%) e com baixas taxas de erro tipo I (13,87%, *FP=444*) e tipo II (4,42%, *FN=147*). Em contrapartida, conforme mostrado nas Figuras 10a, 10b e 10c, o modelo *DA* retorna baixa taxa de *AUC* (79,52%) e elevadas taxas de erro tipo I (53,58%, *FP=1715*) e tipo II (11,91%, *FN=396*).

Portanto, conforme mostrado na Tabela 7 e nas Figuras 8, 9 e 10, o modelo *RF* possui não apenas melhor precisão de classificação, com altas taxas de *ACC* e *AUC*), mas também menor erro geral, com baixas taxas de erro tipo I e tipo II, do que o modelo tradicional *DA* e os demais modelos avaliados. Por esses motivos, o modelo *Random Forests (RF)* é a opção mais indicada para compor a estrutura de priorização de fiscalização de contrato, apresentada na Seção 3.3.

4.2 Implantação

Nesta pesquisa, além do conhecimento técnico obtido em *machine learning*, os resultados da análise dos atributos utilizados (Seção 3.1.3) revelaram informações valiosas sobre os fornecedores avaliados. Portanto, a melhor forma encontrada de apresentar o conhecimento gerado foi delineando as regras do modelo final, nos moldes das análises feitas nas Seções 2.1.2

(Figura 3) e 3.3 (Figura 6).

Sugere-se, portanto, a aplicação do modelo proposto em uma amostra maior e bem definida, bem como, possibilitar a discussão dos resultados por especialistas. Portanto, com os cenários apresentados e os resultados de aplicação dos modelos em determinados casos, pode-se, através de relatórios estruturados e *workshops*, levar o conhecimento obtido pelo trabalho ao alcance de todos os interessados.

5 Considerações Finais

Neste trabalho, técnicas de *machine learning* foram utilizadas para classificar fornecedores do governo federal brasileiro (GFB), com base no risco de inadimplência contratual. O objetivo principal foi demonstrar como essas técnicas podem ser úteis ao *GFB*, reduzindo a inadimplência nas aquisições e prevenindo irregularidades nos contratos administrativos. Para isso, utilizando-as na automatização de tarefa de classificação, buscou-se otimizar a qualidade da Gestão de Contrato, que é executada por mão de obra, muitas vezes, escassa e cara, que não consegue acompanhar e/ou fiscalizar, com eficiência, todos os contratos públicos.

Desde o início desta pesquisa, foi verificado que o processo de contratação pública, embora contando com inúmeras melhorias, ainda está sujeito a resultados indesejáveis como improbidade, quebra de contrato, ineficiência, inadimplência, fraude e corrupção. Assim, constatou-se que a identificação prévia de fornecedor suspeito, com chance elevada de inadimplir ou de estar envolvido em atividade irregular e/ou ilegal, no decorrer desse processo, é essencial, pois, com isso, é possível priorizar a fiscalização dos contratos arriscados, vinculados a fornecedores que oferecem alto risco de inadimplência.

Diante disso, dados de fornecedores do *GFB* foram reunidos e analisados com o propósito de responder as seguintes questões de pesquisa: como a inteligência artificial pode ser utilizada para reduzir a inadimplência nas aquisições do governo federal brasileiro e previnir irregularidades em contratos públicos? Quais algoritmos e métodos de análise de dados podem ser utilizados?

Para atingir o objetivo geral do trabalho e responder a estas questões de pesquisa, uma estrutura de priorização de fiscalização de contrato, baseada em risco, foi desenvolvida. Nessa estrutura, fornecedores são classificados automaticamente, segundo seus riscos de inadimplência, gerando alertas de priorização para a fiscalização dos contratos com os fornecedores mais arriscados. Além de previnir irregularidades, a referida estrutura pode, também, reduzir o esforço manual de gestores e fiscais de contratos, tornando mais eficiente e eficaz a execução dos contratos públicos.

O primeiro objetivo específico foi definir e fundamentar o método mais adequado ao objeto e ao problema de pesquisa; e o segundo, foi investigar as irregularidades contratuais e apresentar possíveis estratégias de prevenção baseadas em técnicas de análise de dados. Para atingir esses objetivos, foi feito uma revisão de literatura sobre os métodos de análise de dados,

trazendo entendimento sobre o uso de técnicas de avaliação de crédito e demonstrando suas possíveis aplicações no setor público e no setor privado.

O terceiro objetivo específico foi selecionar, analisar e estruturar os dados; e o quarto, foi avaliar diferentes técnicas estatísticas de classificação, apresentar uma estrutura de priorização de fiscalização de contrato (EPFC) e empregar diferentes métricas de validação de modelos. Para atingir esses objetivos, após a estruturação dos dados coletados do *SIASG*, *CEIS* e *RFB*, técnicas tradicionais de análise estatística e técnicas de *machine learning* foram selecionadas e, individualmente, foram aplicadas na *EPFC* e avalidas por meio de métricas tradicionais de validação.

O quinto e último objetivo específico foi implementar os diferentes modelos de classificação, gerados na etapa anterior, na *EPFC* e realizar a comparação dos resultados. Para atingir esses objetivos, foi feito uma análise comparativa de desempenho entre os seis melhores modelos de avaliação de risco de inadimplência empregados (*RF*, *GB*, *BG*, *DT*, *DA* e *KNN*), na qual, o modelo *random forest (RF)*, seguido dos modelos *gradient boosting (GB)* e *bagging (BG)*, apresentou os melhores resultados.

Neste trabalho, foi possível apresentar uma das possíveis soluções automáticas para o setor público, destacando o papel da inteligência artificial na análise proativa de dados e o benefício potencial das técnicas de *machine learning* no gerenciamento de contratos públicos. Contudo, diante do método proposto e dos resultados obtidos, percebe-se que o trabalho poderia ter sido realizado, utilizando uma base de dados mais ampla. Portanto, estudos futuros, contendo um número maior de dados caracterizando esses fornecedores, podem gerar novas informações, abrindo novas possibilidades de estudos sobre o tema.

Referências

ADJEI-BAMFO, Peter; MALOREH-NYAMEKYE, Theophilus; AHENKAN, Albert. The role of e-government in sustainable public procurement in developing countries: A systematic literature review. **Resources, Conservation and Recycling**, Elsevier, v. 142, p. 189–203, 2019.

AKKOÇ, Soner. An empirical comparison of conventional techniques, neural networks and the three stage hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) model for credit scoring analysis: The case of Turkish credit card data. **European Journal of Operational Research**, Elsevier, v. 222, n. 1, p. 168–178, 2012.

ALTMAN, Edward I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. **The journal of finance**, JSTOR, v. 23, n. 4, p. 589–609, 1968.

ATTARD, Judie et al. A systematic review of open government data initiatives. **Government Information Quarterly**, Elsevier, v. 32, n. 4, p. 399–418, 2015.

AZODI, Christina B; TANG, Jiliang; SHIU, Shin-Han. Opening the black box: Interpretable machine learning for geneticists. **Trends in Genetics**, Elsevier, 2020.

BAESENS, Bart et al. Benchmarking state-of-the-art classification algorithms for credit scoring. **Journal of the operational research society**, Springer, v. 54, n. 6, p. 627–635, 2003.

BARBOZA, Flavio; KIMURA, Herbert; ALTMAN, Edward. Machine learning models and bankruptcy prediction. **Expert Systems with Applications**, Elsevier, v. 83, p. 405–417, 2017.

BERRU, Yeferson Torres et al. Artificial Intelligence Techniques to Detect and Prevent Corruption in Procurement: A Systematic Literature Review. In: SPRINGER. INTERNATIONAL Conference on Applied Technologies. [S.l.: s.n.], 2019. P. 254–268.

BLOOMFIELD, Katherine et al. Systemic risk in major public contracts. **International Journal of Forecasting**, Elsevier, v. 35, n. 2, p. 667–676, 2019.

BOUTABA, Raouf et al. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. **Journal of Internet Services and Applications**, Springer, v. 9, n. 1, p. 1–99, 2018.

BRASIL. Guia de Boas Práticas em Contratação de Soluções de Tecnologia da Informação. **Ministério do Planejamento, Desenvolvimento e Gestão. Secretaria de Tecnologia da Informação e Comunicação.**, p. 1–204, 2017.

_____. Lei nº 14.133, de 1º de abril de 2021. **Lei de licitações e Contratos Administrativos. Presidência da República: Casa Civil.**, p. 1–75, 2021.

BREIMAN, Leo. Bagging predictors. **Machine learning**, Springer, v. 24, n. 2, p. 123–140, 1996.

_____. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BROWN, Trevor L; POTOSKI, Matthew; VAN SLYKE, David M. Contracting for Complex Products. **Journal of Public Administration Research and Theory: J-PART**, JSTOR, p. i41–i58, 2010.

CARNEIRO, Davide et al. Network Analysis for Fraud Detection in Portuguese Public Procurement. In: SPRINGER. INTERNATIONAL Conference on Intelligent Data Engineering and Automated Learning. [S.l.: s.n.], 2020. P. 390–401.

CASTRO VIEIRA, José Rômulo de et al. Machine learning models for credit analysis improvements: Predicting low-income families' default. **Applied Soft Computing**, Elsevier, v. 83, p. 105640, 2019.

CAVALCANTE, Ian M et al. A supervised machine learning approach to data-driven simulation of resilient supplier selection in digital manufacturing. **International Journal of Information Management**, Elsevier, v. 49, p. 86–97, 2019.

CHAWLA, Nitesh V et al. SMOTE: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, v. 16, p. 321–357, 2002.

CHEN, CL Philip; ZHANG, Chun-Yang. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. **Information sciences**, Elsevier, v. 275, p. 314–347, 2014.

CHEN, Weimin; MA, Chaoqun; MA, Lin. Mining the customer credit using hybrid support vector machine technique. **Expert systems with applications**, Elsevier, v. 36, n. 4, p. 7611–7616, 2009.

CHOI, Hyunchul; SON, Hyojoo; KIM, Changwan. Predicting financial distress of contractors in the construction industry using ensemble learning. **Expert Systems with Applications**, Elsevier, v. 110, p. 1–10, 2018.

CONCHA, Gaston et al. E-Government procurement observatory, maturity model and early measurements. **Government Information Quarterly**, Elsevier, v. 29, s43–s50, 2012.

DAL BÓ, Ernesto; ROSSI, Martın A. Corruption and inefficiency: Theory and evidence from electric utilities. **Journal of Public Economics**, Elsevier, v. 91, n. 5-6, p. 939–962, 2007.

DASTILE, Xolani; CELIK, Turgay; POTSANE, Moshe. Statistical and machine learning models in credit scoring: A systematic literature survey. **Applied Soft Computing**, Elsevier, p. 106263, 2020.

DÁVID-BARRETT, Elizabeth; FAZEKAS, Mihály. Anti-corruption in aid-funded procurement: Is corruption reduced or merely displaced? **World Development**, Elsevier, v. 132, p. 105000, 2020.

DE CNUDDE, Sofie; MARTENS, David. Loyal to your city? A data mining analysis of a public service loyalty program. **Decision Support Systems**, Elsevier, v. 73, p. 74–84, 2015.

DE MENEZES, Fortunato S et al. Data classification with binary response through the Boosting algorithm and logistic regression. **Expert Systems with Applications**, Elsevier, v. 69, p. 62–73, 2017.

DELMONICO, Diego et al. Unveiling barriers to sustainable public procurement in emerging economies: Evidence from a leading sustainable supply chain initiative in Latin America. **Resources, Conservation and Recycling**, Elsevier, v. 134, p. 70–79, 2018.

ENGELMANN, Bernd; HAYDEN, Evelyn; TASCHE, Dirk. Testing rating accuracy. **Risk**, Citeseer, v. 16, n. 1, p. 82–86, 2003.

ERDIS, Ercan. The effect of current public procurement law on duration and cost of construction projects in Turkey. **Journal of Civil Engineering and Management**, Taylor & Francis, v. 19, n. 1, p. 121–135, 2013.

FAWCETT, Tom. An introduction to ROC analysis. **Pattern recognition letters**, Elsevier, v. 27, n. 8, p. 861–874, 2006.

FAWCETT, Tom; PROVOST, Foster. Adaptive fraud detection. **Data mining and knowledge discovery**, Springer, v. 1, n. 3, p. 291–316, 1997.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37–37, 1996.

FAZEKAS, Mihály; KOCSIS, Gábor. Uncovering high-level corruption: cross-national objective corruption risk indicators using public procurement data. **British Journal of Political Science**, Cambridge University Press, p. 1–10, 2017.

FAZEKAS, Mihály; TÓTH, Bence. The extent and cost of corruption in transport infrastructure. New evidence from Europe. **Transportation research part A: policy and practice**, Elsevier, v. 113, p. 35–54, 2018.

FAZEKAS, Mihály; TÓTH, István János; KING, Lawrence Peter. An objective corruption risk index using public procurement data. **European Journal on Criminal Policy and Research**, Springer, v. 22, n. 3, p. 369–397, 2016.

FENG, Xiaodong et al. Dynamic ensemble classification for credit scoring using soft probability. **Applied Soft Computing**, Elsevier, v. 65, p. 139–151, 2018.

FITZPATRICK, Trevor; MUES, Christophe. An empirical comparison of classification algorithms for mortgage default prediction: evidence from a distressed mortgage market. **European Journal of Operational Research**, Elsevier, v. 249, n. 2, p. 427–439, 2016.

FRIEDMAN, Nir; GEIGER, Dan; GOLDSZMIDT, Moises. Bayesian network classifiers. **Machine learning**, Springer, v. 29, n. 2-3, p. 131–163, 1997.

GALLEGO, Jorge; RIVERO, Gonzalo; MARTINEZ, Juan. Preventing rather than punishing: An early warning model of malfeasance in public procurement. **International Journal of Forecasting**, Elsevier, 2020.

GARCIA, Vicente; MARQUÉS, AI; SÁNCHEZ, José Salvador. On the use of data filtering techniques for credit risk prediction with instance-based models. **Expert Systems with Applications**, Elsevier, v. 39, n. 18, p. 13267–13276, 2012.

HAND, David J; HENLEY, William E. Statistical classification methods in consumer credit scoring: a review. **Journal of the Royal Statistical Society: Series A (Statistics in Society)**, Wiley Online Library, v. 160, n. 3, p. 523–541, 1997.

HANDAN-NADER, Cassandra; HO, Daniel E; LIU, Larry Y. Deep learning with satellite imagery to enhance environmental enforcement. In: DATA Science Applied to Sustainability Analysis. [S.l.]: Elsevier, 2021. P. 205–228.

HEGDE, Jeevith; ROKSETH, Børge. Applications of machine learning methods for engineering risk assessment–A review. **Safety science**, Elsevier, v. 122, p. 104492, 2020.

HENDRIKX, Ferry; BUBENDORFER, Kris; CHARD, Ryan. Reputation systems: A survey and taxonomy. **Journal of Parallel and Distributed Computing**, Elsevier, v. 75, p. 184–197, 2015.

HENRIQUE, Bruno M; SOBREIRO, Vinicius A; KIMURA, Herbert. Contracting in Brazilian public administration: A machine learning approach. **Expert Systems**, Wiley Online Library, p. 1–17, 2020.

HEO, Junyoung; YANG, Jin Yong. AdaBoost based bankruptcy forecasting of Korean construction companies. **Applied soft computing**, Elsevier, v. 24, p. 494–499, 2014.

HOODA, Nishtha; BAWA, Seema; RANA, Prashant Singh. Fraudulent firm classification: a case study of an external audit. **Applied Artificial Intelligence**, Taylor & Francis, v. 32, n. 1, p. 48–64, 2018.

_____. Optimizing fraudulent firm prediction using ensemble machine learning: a case study of an external audit. **Applied Artificial Intelligence**, Taylor & Francis, v. 34, n. 1, p. 20–30, 2020.

JANS, Mieke; LYBAERT, Nadine; VANHOOF, Koen. Internal fraud risk reduction: Results of a data mining case study. **International Journal of Accounting Information Systems**, Elsevier, v. 11, n. 1, p. 17–41, 2010.

JARDIN, Philippe du. A two-stage classification technique for bankruptcy prediction. **European Journal of Operational Research**, Elsevier, v. 254, n. 1, p. 236–252, 2016.

JIMENEZ, Gabriel; SALAS, Vicente; SAURINA, Jesus. Determinants of collateral. **Journal of financial economics**, Elsevier, v. 81, n. 2, p. 255–281, 2006.

JORDAN, Michael I; MITCHELL, Tom M. Machine learning: Trends, perspectives, and prospects. **Science**, American Association for the Advancement of Science, v. 349, n. 6245, p. 255–260, 2015.

JURVETSON, S. Intelligence Inside: the future of AI and computation. **Flickr**, n.p. disponível em: https://www.flickr.com/photos/jurvetson/28799579322. acesso em: 30 nov. 2021, 2016.

KAMATH, Uday; LIU, John; WHITAKER, James. **Deep learning for NLP and speech recognition**. [S.l.]: Springer, 2019. v. 84.

KAVAKIOTIS, Ioannis et al. Machine learning and data mining methods in diabetes research. **Computational and structural biotechnology journal**, Elsevier, v. 15, p. 104–116, 2017.

KIM, Soo Y; UPNEJA, Arun. Predicting restaurant financial distress using decision tree and AdaBoosted decision tree models. **Economic Modelling**, Elsevier, v. 36, p. 354–362, 2014.

KIMURA, Herbert; BASSO, Leonardo Fernando Cruz; KAYO, Eduardo Kazuo. Decision models in credit risk management. In: DECISION Models in Engineering and Management. [S.l.]: Springer, 2015. P. 57–73.

KLABI, Hichem; MELLOULI, Sehl; REKIK, Monia. A reputation based electronic government procurement model. **Government Information Quarterly**, Elsevier, v. 35, n. 4, s43–s53, 2018.

KOSTYO, Kenneth. **Handbook for curbing corruption in public procurement**. [S.l.: s.n.], 2006.

KRUPPA, Jochen et al. Consumer credit risk: Individual probability estimates using machine learning. **Expert Systems with Applications**, Elsevier, v. 40, n. 13, p. 5125–5131, 2013.

KU, Chih-Hao; LEROY, Gondy. A decision support system: Automated crime report analysis and classification for e-government. **Government Information Quarterly**, Elsevier, v. 31, n. 4, p. 534–544, 2014.

KÜHN, Susanne; SHERMAN, Laura B. Curbing corruption in public procurement: a practical guide. **Berlin, Germany: Transparency International**, 2014.

KUM, Hye-Chung; DUNCAN, Dean F; STEWART, C Joy. Supporting self-evaluation in local government via Knowledge Discovery and Data mining. **Government Information Quarterly**, Elsevier, v. 26, n. 2, p. 295–304, 2009.

KUZIEMSKI, Maciej; MISURACA, Gianluca. AI governance in the public sector: Three tales from the frontiers of automated decision-making in democratic settings. **Telecommunications Policy**, Elsevier, p. 101976, 2020.

LAWRENCE, Edward C; SMITH, L Douglas; RHOADES, Malcolm. An analysis of default risk in mobile home credit. **Journal of Banking & Finance**, Elsevier, v. 16, n. 2, p. 299–312, 1992.

LE, Hong Hanh; VIVIANI, Jean-Laurent. Predicting bank failure: An improvement by implementing a machine-learning approach to classical financial ratios. **Research in International Business and Finance**, Elsevier, v. 44, p. 16–25, 2018.

LI, Shouwei; WANG, Mingliang; HE, Jianmin. Prediction of banking systemic risk based on support vector machine. **Mathematical Problems in Engineering**, Hindawi, v. 2013, 2013.

LIM, Tjen-Sien; LOH, Wei-Yin; SHIH, Yu-Shan. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. **Machine learning**, Springer, v. 40, n. 3, p. 203–228, 2000.

LIMA, Marcio Salles Melo; DELEN, Dursun. Predicting and explaining corruption across countries: A machine learning approach. **Government Information Quarterly**, Elsevier, v. 37, n. 1, p. 101407, 2020.

LIU, Yun et al. Machine learning for advanced energy materials. **Energy and AI**, Elsevier, v. 3, p. 100049, 2021.

LÓPEZ-ITURRIAGA, Félix J; SANZ, Iván Pastor. Predicting public corruption with neural networks: An analysis of spanish provinces. **Social Indicators Research**, Springer, v. 140, n. 3, p. 975–998, 2018.

LOUKIS, Euripidis; KYRIAKOU, Niki; MARAGOUDAKIS, Manolis. Using Government Data and Machine Learning for Predicting Firms' Vulnerability to Economic Crisis. In: SPRINGER. INTERNATIONAL Conference on Electronic Government. [S.l.: s.n.], 2020. P. 345–358.

LOUZADA, Francisco; ARA, Anderson; FERNANDES, Guilherme B. Classification methods applied to credit scoring: Systematic review and overall comparison. **Surveys in Operations Research and Management Science**, Elsevier, v. 21, n. 2, p. 117–134, 2016.

LU, Chi-Lin; CHEN, Ta-Cheng. A study of applying data mining approach to the information disclosure for Taiwan's stock market investors. **Expert Systems with Applications**, Elsevier, v. 36, n. 2, p. 3536–3542, 2009.

MAHMOUDI, Nader; DUMAN, Ekrem. Detecting credit card fraud by modified Fisher discriminant analysis. **Expert Systems with Applications**, Elsevier, v. 42, n. 5, p. 2510–2516, 2015.

MARQUÉS, Ana I; GARCIA, Vicente; SÁNCHEZ, José Salvador. Exploring the behaviour of base classifiers in credit scoring ensembles. **Expert Systems with Applications**, Elsevier, v. 39, n. 11, p. 10244–10250, 2012.

MARTINEZ-PLUMED, Fernando et al. SALER: a data science solution to detect and prevent corruption in public administration. In: SPRINGER. JOINT European Conference on Machine Learning and Knowledge Discovery in Databases. [S.l.: s.n.], 2018. P. 103–117.

MEDVEDEVA, Masha; VOLS, Michel; WIELING, Martijn. Using machine learning to predict decisions of the European Court of Human Rights. **Artificial Intelligence and Law**, Springer, v. 28, n. 2, p. 237–266, 2020.

MOSTAFA, Mohamed M; EL-MASRY, Ahmed A. Citizens as consumers: Profiling e-government services' users in Egypt via data mining techniques. **International Journal of Information Management**, Elsevier, v. 33, n. 4, p. 627–641, 2013.

NEVES, Fabrıcio Ramos; SILVA, Polyana Batista da; CARVALHO, Hugo Leonardo Menezes de. Artificial ladies against corruption: searching for legitimacy at the Brazilian Supreme Audit Institution. **Revista de Contabilidade e Organizações**, v. 13, p. 31–50, 2019.

NGAI, Eric WT et al. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. **Decision support systems**, Elsevier, v. 50, n. 3, p. 559–569, 2011.

OBWEGESER, Nikolaus; MÜLLER, Sune Dueholm. Innovation and public procurement: Terminology, concepts, and applications. **Technovation**, Elsevier, v. 74, p. 1–17, 2018.

OVSYANNIKOVA, Alexandra; DOMASHOVA, Jenny. Identification of public procurement contracts with a high risk of non-performance based on neural networks. **Procedia Computer Science**, Elsevier, v. 169, p. 795–799, 2020.

PHUA, Clifton et al. A comprehensive survey of data mining-based fraud detection research. **arXiv preprint arXiv:1009.6119**, 2010.

RALHA, Celia Ghedini; SILVA, Carlos Vinıcius Sarmento. A multi-agent data mining system for cartel detection in Brazilian government procurement. **Expert Systems with Applications**, Elsevier, v. 39, n. 14, p. 11642–11656, 2012.

ROZARIO, Andrea M; ISSA, Hussein. Risk-based data analytics in the government sector: A case study for a US county. **Government Information Quarterly**, Elsevier, v. 37, n. 2, p. 101457, 2020.

SAINI, Jaspal Kaur; BANSAL, Divya. A comparative study and automated detection of illegal weapon procurement over dark web. **Cybernetics and Systems**, Taylor & Francis, v. 50, n. 5, p. 405–416, 2019.

SANTOS, Lucimar Rizzo Lopes dos. A fiscalização e o acompanhamento da execução do contrato administrativo na Administração Pública-artigo 67-Lei Nº 8.666/93. **Revista Digital de Direito Administrativo**, v. 4, n. 2, p. 187–199, 2017.

SOUSA, Weslei Gomes de et al. How and where is artificial intelligence in the public sector going? A literature review and research agenda. **Government Information Quarterly**, Elsevier, v. 36, n. 4, p. 101392, 2019.

SUN, Ming-Yi; WANG, Szu-Fang. Validation of credit rating models-a preliminary look at methodology and literature. **Review of Financial Risk Management**, v. 2, n. 94, p. 1–15, 2005.

SUN, Ting; SALES, Leonardo J. Predicting public procurement irregularity: An application of neural networks. **Journal of Emerging Technologies in Accounting**, American Accounting Association, v. 15, n. 1, p. 141–154, 2018.

TANG, Yue et al. Discriminant analysis of zero recovery for China's NPL. **Journal of Applied Mathematics and Decision Sciences**, v. 2009, p. 14, 2009.

THARWAT, Alaa. Classification assessment methods. **Applied Computing and Informatics**, Emerald Publishing Limited, 2020.

TITL, Vitezslav; GEYS, Benny. Political donations and the allocation of public procurement contracts. **European Economic Review**, Elsevier, v. 111, p. 443–458, 2019.

TKACHENKO, Andrey; YAKOVLEV, Andrei; KUZNETSOVA, Aleksandra. ‘Sweet deals’: State-owned enterprises, corruption and repeated contracts in public procurement. **Economic Systems**, Elsevier, v. 41, n. 1, p. 52–67, 2017.

TSAI, Chih-Fong; HSU, Yu-Feng; YEN, David C. A comparative study of classifier ensembles for bankruptcy prediction. **Applied Soft Computing**, Elsevier, v. 24, p. 977–984, 2014.

TSUKAHARA, Fábio Yasuhiro et al. Validation of default probability models: A stress testing approach. **International Review of Financial Analysis**, Elsevier, v. 47, p. 70–85, 2016.

VOUGAS, Konstantinos et al. Machine learning and data mining frameworks for predicting drug response in cancer: An overview and a novel in silico screening process based on association rule mining. **Pharmacology & therapeutics**, Elsevier, v. 203, p. 107395, 2019.

WACHS, Johannes; FAZEKAS, Mihály; KERTÉSZ, János. Corruption risk in contracting markets: A network science perspective. **International Journal of Data Science and Analytics**, Springer, p. 1–16, 2020.

WANG, Gang; HAO, Jinxing et al. A comparative assessment of ensemble learning for credit scoring. **Expert systems with applications**, Elsevier, v. 38, n. 1, p. 223–230, 2011.

WANG, Gang; MA, Jian. A hybrid ensemble approach for enterprise credit risk assessment based on Support Vector Machine. **Expert Systems with Applications**, Elsevier, v. 39, n. 5, p. 5325–5331, 2012.

WANG, Gang; MA, Jian; HUANG, Lihua et al. Two credit scoring models based on dual strategy ensemble trees. **Knowledge-Based Systems**, Elsevier, v. 26, p. 61–68, 2012.

WANG, Gang; MA, Jian; YANG, Shanlin. An improved boosting based on feature selection for corporate bankruptcy prediction. **Expert Systems with Applications**, Elsevier, v. 41, n. 5, p. 2353–2361, 2014.

WANG, Ran; ASGHARI, Vahid et al. Detecting corporate misconduct through random forest in China’s construction industry. **Journal of Cleaner Production**, Elsevier, p. 122266, 2020.

YAP, Bee Wah; ONG, Seng Huat; HUSAIN, Nor Huselina Mohamed. Using data mining to improve assessment of credit worthiness via credit scoring models. **Expert Systems with Applications**, Elsevier, v. 38, n. 10, p. 13274–13283, 2011.

YEH, Ching-Chiang; CHI, Der-Jang; LIN, Yi-Rong. Going-concern prediction using hybrid random forests and rough set approach. **Information Sciences**, Elsevier, v. 254, p. 98–110, 2014.

ZENG, Guoping. On the confusion matrix in credit scoring and its analytical properties. **Communications in Statistics-Theory and Methods**, Taylor & Francis, v. 49, n. 9, p. 2080–2093, 2020.

ZHAO, Zongyuan et al. Investigation and improvement of multi-layer perceptron neural networks for credit scoring. **Expert Systems with Applications**, Elsevier, v. 42, n. 7, p. 3508–3516, 2015.

ZHU, You et al. Forecasting SMEs’ credit risk in supply chain finance with an enhanced hybrid ensemble machine learning approach. **International Journal of Production Economics**, Elsevier, v. 211, p. 22–33, 2019.

APÊNDICE A - Variáveis preditoras e componentes de riscos

Variáveis preditoras	Componentes de riscos	Descrição*
Porte	Capacidade operacional do fornecedor (*Cpcd Op Forn*)	A
Nr_CNAE_Sec	Capacidade operacional do fornecedor (*Cpcd Op Forn*)	B
Tp_Exis	Capacidade operacional do fornecedor (*Cpcd Op Forn*)	C
Nr_Ocr	Histórico de penalidades do fornecedor (*Hist P Forn*)	D
Tot_Venda	Perfil do fornecedor (*Pf Forn*)	E

*

A **Porte do fornecedor (renda anual):** *Porte*, microempresas ou empresas de pequeno porte com licitações de valores incompatíveis. Busca microempresas ou empresas de pequeno porte que venceram licitações cujo valor total em um ano específico superou os limites legais para que elas sejam ainda consideradas microempresas ou empresas de pequeno porte. De acordo com Lei Complementar nº 155/2016, que trata deste assunto, o limite é de: R$ 360.000,00 (trezentos e sessenta mil reais), para microempresas; e R$ 4.800.000,00 (quatro milhões e oitocentos mil reais), para empresas de pequeno porte. Para o cálculo dos valores totais e obtenção do porte, utilizou-se a base de dados do *ComprasNet/SIASG*.

B **Número de CNAE secundários:** *Nr_CNAE_Sec*, grande quantidade de subclasses de atividade econômica secundária. Busca fornecedores com mais de 10 subclasses de atividade econômica secundária. Para obter todas as atividades econômicas de um fornecedor, utilizou-se a base de dados do *ComprasNet/SIASG* e a base de dados de Classificação Nacional de Atividades Econômicas (CNAE) da Receita Federal do Brasil.

C **Tempo de Existência (idade):** *Tp_Exis*, empresas mais novas costumam ser mais inadimplentes.

D **Número de ocorrências (histórico de dívida):** *Nr_Ocr*, quantidade de advertências e punições recebidas em seu histórico de contratações com o governo federal, por exemplo, *advertência*, *multa*, *impedimento de licitar e contratar* e *declaração de inidoneidade para licitar ou contratar*, incisos I a IV do Art. 156 da Lei nº 14.133/21, Brasil (2021).

E **Total de Venda (valor do contrato):** *Tot_Venda*, valor do contrato.
Sabe-se que, juntamente com taxa de juros, data de vencimento e valor do contrato, o uso de garantias em contratos de crédito pode ser exigido para a concessão de empréstimos. Estudos empíricos sobre o assunto, por exemplo, Jimenez, Salas e Saurina (2006) e Tang et al. (2009), mostraram que as garantias são variáveis preditoras significativas para a classificação do risco de crédito. Outra variável, apontada por esses autores, é a área geográfica, isto é, as condições econômicas, do local onde vive o mutuário, podem influenciar a recuperação do crédito.

Jimenez, Salas e Saurina (op. cit.), testaram teorias que vinculam o uso de garantias ao risco de crédito do tomador em uma amostra de empréstimos concedidos a empresas espanholas, no período de 1984 a 2002. Para a realização dos testes, os pesquisadores agruparam a amostra em três classes de risco: *i)* mutuários que possuiam um empréstimo em *default* (inadimplente) no momento em que obtiveram um novo empréstimo; *ii)* mutuários que não possuiam um empréstimo em *default* no momento em que um novo empréstimo era concedido, mas que não pagariam no futuro próximo; e *iii)* todos os outros mutuários.

Dentre as conclusões obtidas, foi verificado que a probabilidade do uso de garantias é maior entre *i)* e *ii)* do que entre os mutuários que não inadimplem antes ou após o recebimento do empréstimo; que, o credor tem uma capacidade melhor de estimar o risco para mutuários mais velhos; e que para os jovens, que não têm registro prévio de atividade financeira ou comercial, a necessidade do uso de garantias é maior.

Fonte: Elaboração própria.

APÊNDICE B - Resultados com dados de treino (código-fonte em *python*)

1. Preparação dos dados

a) Livrarias do *python* utilizadas

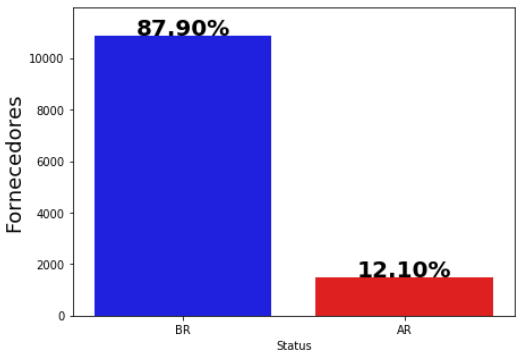
```
In [1]:   import pandas as pd
          import numpy as np
          import seaborn as sns
          from collections import Counter
          import matplotlib.pyplot as plt
          from sklearn.metrics import accuracy_score, confusion_matrix,
          classification_report, precision_score, recall_score, f1_score
          from sklearn.model_selection import cross_val_score
          %matplotlib inline
```

b) Importação dos dados

```
In [2]:   ##### Dados importados das bases do SIASG, CEIS e RFB
          FGF = pd.read_csv('Fornecedores do Governo Federal.csv')
          FGF.head()
Out[2]:
```

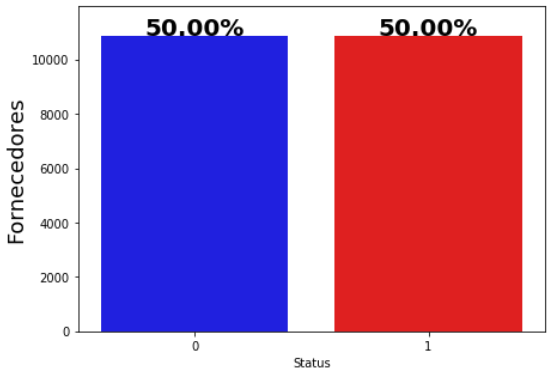
	CNPJ	Porte	Nr_CNAE_Sec	Nr_Ocr	Tp_Exis	Tot_Venda	Status
0	21446397000110	ME	47	6	391	26776.65	BR
1	9077347000107	PP	2	9	3006	50298.00	AR
2	21544098000118	PP	2	0	360	11500.00	BR
3	33457045000149	ME	0	0	17914	1602.70	BR
4	28938421000187	PP	0	2	11232	6035.00	BR

```
In [3]:   print("Conjunto de dados original, linhas e colunas, Grupo de Baixo Risco (Gp
          BR) e Grupo de Alto Risco (Gp AR):{}".format(FGF.shape))
          print ('Gp BR {0} e Gp AR {1}:')
          FGF.Status.value_counts()
          Conjunto de dados original, linhas e colunas, Grupo de Baixo Risco (Gp BR) e
          Grupo de Alto Risco (Gp AR):(12376, 7)
          Gp BR {0} e Gp AR {1}:
Out[3]:   BR    10878
          AR     1498
          Name: Status, dtype: int64

In [4]:   total = len(FGF)
          plt.figure(figsize=(7,5))
          colors = ['blue', 'red']
          g = sns.countplot(x='Status', data=FGF, palette=colors)
          g.set_ylabel('Fornecedores', fontsize=18)
          for p in g.patches:
              height = p.get_height()
              g.text(p.get_x()+p.get_width()/2., height +
          1.5,'{:1.2f}%'.format(height/total*100), ha="center", fontsize=20,
          fontweight='bold')
          plt.margins(y=0.1)
          plt.show()
```



c) Pré-processamento

```
In [5]:   FGF['Porte'] = FGF['Porte'].replace({'PP': 0, 'ME': 1, 'DM': 2})
          FGF['Status'] = FGF['Status'].replace({'BR': 0,'AR': 1})
          FGF = FGF.drop('CNPJ', axis=1)
          X = FGF.drop('Status', axis=1)
          y = FGF ['Status']
          print("Conjunto de dados original, linhas e colunas:{}".format(FGF.shape))
          print ('Gp BR {0} e Gp AR {1}')
          FGF.Status.value_counts()
          Conjunto de dados original, linhas e colunas:(12376, 6)
          Gp BR {0} e Gp AR {1}
Out[5]:   0    10878
          1     1498
          Name: Status, dtype: int64

In [6]:   ### OVERSAMPLING (Aumento fitício da classe minoritária {1} deixando-a igual à
          classe majoritária {0})
          from imblearn.over_sampling import SMOTE
          smt = SMOTE()
          X, y = smt.fit_sample(X, y)
          FGF = pd.DataFrame(X, y)
          FGF['Status'] = y
          total = len(FGF)
          plt.figure(figsize=(7,5))
          colors = ['blue', 'red']
          g = sns.countplot(x='Status', data=FGF, palette=colors)
          g.set_ylabel('Fornecedores', fontsize=18)
          for p in g.patches:
              height = p.get_height()
              g.text(p.get_x()+p.get_width()/2., height +
          1.5,'{:1.2f}%'.format(height/total*100),ha="center", fontsize=20,
          fontweight='bold')
          plt.margins(y=0.1)
          print ('Conjunto de dados final, Gp BR {0} e Gp AR {1}:', format(Counter(y)))
          plt.show()
          Conjunto de dados final, Gp BR {0} e Gp AR {1}: Counter({0: 10878, 1: 10878})
```



d) Divisão da base de dados (dados de treino e de teste)

```
In [7]:   from sklearn.model_selection import train_test_split
          X_train, X_test, y_train, y_test = train_test_split (X, y, test_size = 0.3)

In [8]:   print('Dados de Treino, Gp BR {0} e Gp AR {1}:', format(Counter(y_train)))
          Dados de Treino, Gp BR {0} e Gp AR {1}: Counter({0: 7677, 1: 7552})
```

2. Precisão das técnicas utilizadas

a) *Discriminant analysis - DA* (análise de discriminante)

```
In [9]:   from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
          DA = QuadraticDiscriminantAnalysis().fit(X_train, y_train)
          DA_cv_score = cross_val_score(DA, X, y, cv=10, scoring= 'roc_auc')
          y_pred = DA.predict(X_train)
          cm = confusion_matrix(y_train, y_pred)
          VN = cm[0, 0]
          FP = cm[0, 1]
          FN = cm[1, 0]
          VP = cm[1, 1]
          sensitivity = VP / float(VP + FN)
          specificity = VN / (VN + FP)
```

```
precision = VP / (VP + FP)
F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
print('\033[1m' + 'RESULTADO - Treino:'), print('\033[0m' + "(1) Acurácia
(ACC): {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I
(TFP):", 1 - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),
print("(4) Precisão:", precision), print("(5) Sensibilidade (Recall):",
sensitivity), print("(6) Especificidade:", specificity), print ("(7) F1
Score:", F1_score), print("(8) Área sob a curva ROC
(AUC):",DA_cv_score.mean()), print("(9) Matriz de confusão
[C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
RESULTADO - Treino:
(1) Acurácia (ACC): 0.6675
(2) Erro tipo I (TFP): 0.5345838218053927
(3) Erro tipo II (TFN): 0.12698622881355937
(4) Precisão: 0.6163410301953819
(5) Sensibilidade (Recall): 0.8730137711864406
(6) Especificidade: 0.4654161781946073
(7) F1 Score: 0.7225601402816592
(8) Área sob a curva ROC (AUC): 0.7956942853037121
(9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
[[3573 4104]
 [ 959 6593]]
```

b) *Logistic regression - LR* (regressão logística)

```
In [10]:  from sklearn.linear_model import LogisticRegression
          LR = LogisticRegression().fit(X_train, y_train)
          LR_cv_score = cross_val_score(LR, X, y, cv=10, scoring= 'roc_auc')
          y_pred = LR.predict(X_train)
          cm = confusion_matrix(y_train, y_pred)
          VN = cm[0, 0]
          FP = cm[0, 1]
          FN = cm[1, 0]
          VP = cm[1, 1]
          sensitivity = VP / float(VP + FN)
          specificity = VN / (VN + FP)
          precision = VP / (VP + FP)
          F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
          print('\033[1m' + 'RESULTADO - Treino:'), print('\033[0m' + "(1) Acurácia
          (ACC): {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I
          (TFP):", 1 - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),
          print("(4) Precisão:", precision), print("(5) Sensibilidade (Recall):",
          sensitivity), print("(6) Especificidade:", specificity), print ("(7) F1
          Score:", F1_score), print("(8) Área sob a curva ROC
          (AUC):",DA_cv_score.mean()), print("(9) Matriz de confusão
          [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
          RESULTADO - Treino:
          (1) Acurácia (ACC): 0.5041
          (2) Erro tipo I (TFP): 0.0
          (3) Erro tipo II (TFN): 1.0
          (4) Precisão: nan
          (5) Sensibilidade (Recall): 0.0
          (6) Especificidade: 1.0
          (7) F1 Score: nan
          (8) Área sob a curva ROC (AUC): 0.6445995654341484
          (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
          [[7677    0]
           [7552    0]]
```

c) *K-Nearest neighbors - KNN* (k-ésimo vizinho mais próximo)

```
In [11]:  from sklearn.neighbors import KNeighborsClassifier
          KNN = KNeighborsClassifier(n_neighbors=9).fit(X_train, y_train)
          KNN_cv_score = cross_val_score(KNN, X, y, cv=10, scoring= 'roc_auc')
          y_pred = KNN.predict(X_train)
          cm = confusion_matrix(y_train, y_pred)
          VN = cm[0, 0]
          FP = cm[0, 1]
          FN = cm[1, 0]
          VP = cm[1, 1]
          sensitivity = VP / float(VP + FN)
          specificity = VN / (VN + FP)
          precision = VP / (VP + FP)
          F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
          print('\033[1m' + 'RESULTADO - Treino:'), print('\033[0m' + "(1) Acurácia
          (ACC): {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I
          (TFP):", 1 - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),
```

```
print("(4) Precisão:", precision), print("(5) Sensibilidade (Recall):",
sensitivity), print("(6) Especificidade:", specificity), print ("(7) F1
Score:", F1_score), print("(8) Área sob a curva ROC
(AUC):",DA_cv_score.mean()), print("(9) Matriz de confusão
[C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
RESULTADO - Treino:
(1) Acurácia (ACC): 0.7732
(2) Erro tipo I (TFP): 0.30480656506447834
(3) Erro tipo II (TFN): 0.147510593220339
(4) Precisão: 0.7334244702665755
(5) Sensibilidade (Recall): 0.852489406779661
(6) Especificidade: 0.6951934349355217
(7) F1 Score: 0.7884874464176362
(8) Área sob a curva ROC (AUC): 0.8032580010461994
(9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
[[5337 2340]
 [1114 6438]]
```

d) *Naive bayes - NB* (classificador de bayes ingênuo)

```
In [12]:  from sklearn.naive_bayes import GaussianNB
          NB = GaussianNB().fit(X_train, y_train)
          NB_cv_score = cross_val_score(NB, X, y, cv=10, scoring= 'roc_auc')
          y_pred = NB.predict(X_train)
          cm = confusion_matrix(y_train, y_pred)
          VN = cm[0, 0]
          FP = cm[0, 1]
          FN = cm[1, 0]
          VP = cm[1, 1]
          sensitivity = VP / float(VP + FN)
          specificity = VN / (VN + FP)
          precision = VP / (VP + FP)
          F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
          print('\033[1m' + 'RESULTADO - Treino:'), print('\033[0m' + "(1) Acurácia
          (ACC): {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I
          (TFP):", 1 - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),
          print("(4) Precisão:", precision), print("(5) Sensibilidade (Recall):",
          sensitivity), print("(6) Especificidade:", specificity), print ("(7) F1
          Score:", F1_score), print("(8) Área sob a curva ROC
          (AUC):",DA_cv_score.mean()), print("(9) Matriz de confusão
          [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
          RESULTADO - Treino:
          (1) Acurácia (ACC): 0.5302
          (2) Erro tipo I (TFP): 0.8731275237723068
          (3) Erro tipo II (TFN): 0.05971927966101698
          (4) Precisão: 0.5144161112720951
          (5) Sensibilidade (Recall): 0.940280720338983
          (6) Especificidade: 0.1268724762276931
          (7) F1 Score: 0.6650121745645252
          (8) Área sob a curva ROC (AUC): 0.6059321143624032
          (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
          [[ 974 6703]
           [ 451 7101]]
```

e) *Support vector machine - SVM* (máquina de vetor de suporte)

```
In [13]:  from sklearn.svm import SVC
          SVM = SVC().fit(X_train, y_train)
          SVM_cv_score = cross_val_score(SVM, X, y, cv=10, scoring= 'roc_auc')
          y_pred = SVM.predict(X_train)
          cm = confusion_matrix(y_train, y_pred)
          VN = cm[0, 0]
          FP = cm[0, 1]
          FN = cm[1, 0]
          VP = cm[1, 1]
          sensitivity = VP / float(VP + FN)
          specificity = VN / (VN + FP)
          precision = VP / (VP + FP)
          F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
          print('\033[1m' + 'RESULTADO - Treino:'), print('\033[0m' + "(1) Acurácia
          (ACC): {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I
          (TFP):", 1 - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),
          print("(4) Precisão:", precision), print("(5) Sensibilidade (Recall):",
          sensitivity), print("(6) Especificidade:", specificity), print ("(7) F1
          Score:", F1_score), print("(8) Área sob a curva ROC
          (AUC):",DA_cv_score.mean()), print("(9) Matriz de confusão
          [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
```

```
RESULTADO - Treino:
(1) Acurácia (ACC): 0.5202
(2) Erro tipo I (TFP): 0.10668229777256744
(3) Erro tipo II (TFN): 0.8591101694915254
(4) Precisão: 0.5650557620817844
(5) Sensibilidade (Recall): 0.1408898305084746
(6) Especificidade: 0.8933177022274326
(7) F1 Score: 0.22554319024907263
(8) Área sob a curva ROC (AUC): 0.5328773114348999
(9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
[[6858  819]
 [6488 1064]]
```

f) *Decision tree - DT* (árvore de decisão)

```
In [14]: from sklearn.tree import DecisionTreeClassifier
         DT = DecisionTreeClassifier(random_state=1234).fit(X_train, y_train)
         DT_cv_score = cross_val_score(DT, X, y, cv=10, scoring= 'roc_auc')
         y_pred = DT.predict(X_train)
         cm = confusion_matrix(y_train, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Treino:'), print('\033[0m' + "(1) Acurácia
         (ACC): {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I
         (TFP):", 1 - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),
         print("(4) Precisão:", precision), print("(5) Sensibilidade (Recall):",
         sensitivity), print("(6) Especificidade:", specificity), print ("(7) F1
         Score:", F1_score), print("(8) Área sob a curva ROC
         (AUC):",DA_cv_score.mean()), print("(9) Matriz de confusão
         [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
         RESULTADO - Treino:
         (1) Acurácia (ACC): 1.0000
         (2) Erro tipo I (TFP): 0.0
         (3) Erro tipo II (TFN): 0.0
         (4) Precisão: 1.0
         (5) Sensibilidade (Recall): 1.0
         (6) Especificidade: 1.0
         (7) F1 Score: 1.0
         (8) Área sob a curva ROC (AUC): 0.8906056790816603
         (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
         [[7677    0]
          [   0 7552]]
```

g) *Random forest - RF* (floresta aleatória)

```
In [15]: from sklearn.ensemble import RandomForestClassifier
         RF = RandomForestClassifier(max_depth=9, n_estimators=100,
         random_state=0).fit(X_train, y_train)
         RF_cv_score = cross_val_score(RF, X, y, cv=10, scoring= 'roc_auc')
         y_pred = RF.predict(X_train)
         cm = confusion_matrix(y_train, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Treino:'), print('\033[0m' + "(1) Acurácia
         (ACC): {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I
         (TFP):", 1 - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),
         print("(4) Precisão:", precision), print("(5) Sensibilidade (Recall):",
         sensitivity), print("(6) Especificidade:", specificity), print ("(7) F1
         Score:", F1_score), print("(8) Área sob a curva ROC
         (AUC):",DA_cv_score.mean()), print("(9) Matriz de confusão
         [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
         RESULTADO - Treino:
         (1) Acurácia (ACC): 0.9261
         (2) Erro tipo I (TFP): 0.12244366288914943
         (3) Erro tipo II (TFN): 0.024496822033898358
```

```
(4) Precisão: 0.8868424220536897
(5) Sensibilidade (Recall): 0.9755031779661016
(6) Especificidade: 0.8775563371108506
(7) F1 Score: 0.929062362065704
(8) Área sob a curva ROC (AUC): 0.9587207346700861
(9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
[[6737  940]
 [ 185 7367]]
```

h) *Bagging* - *BG* (agregação de bootstrap)

```
In [16]: from sklearn.ensemble import BaggingClassifier
         BG = BaggingClassifier(random_state=1234).fit(X_train, y_train)
         BG_cv_score = cross_val_score(BG, X, y, cv=10, scoring= 'roc_auc')
         y_pred = BG.predict(X_train)
         cm = confusion_matrix(y_train, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Treino:'), print('\033[0m' + "(1) Acurácia
         (ACC): {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I
         (TFP):", 1 - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),
         print("(4) Precisão:", precision), print("(5) Sensibilidade (Recall):",
         sensitivity), print("(6) Especificidade:", specificity), print ("(7) F1
         Score:", F1_score), print("(8) Área sob a curva ROC
         (AUC):",DA_cv_score.mean()), print("(9) Matriz de confusão
         [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
         RESULTADO - Treino:
         (1) Acurácia (ACC): 0.9944
         (2) Erro tipo I (TFP): 0.006252442360296984
         (3) Erro tipo II (TFN): 0.004899364406779627
         (4) Precisão: 0.9936533121777072
         (5) Sensibilidade (Recall): 0.9951006355932204
         (6) Especificidade: 0.993747557639703
         (7) F1 Score: 0.9943764472378432
         (8) Área sob a curva ROC (AUC): 0.9606801736222884
         (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
         [[7629   48]
          [  37 7515]]
```

i) *Gradient boosting* - *GB* (aumento de gradiente)

```
In [17]: from sklearn.ensemble import GradientBoostingClassifier
         GB = GradientBoostingClassifier(random_state=1234).fit(X_train, y_train)
         GB_cv_score = cross_val_score(GB, X, y, cv=10, scoring= 'roc_auc')
         y_pred = GB.predict(X_train)
         cm = confusion_matrix(y_train, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Treino:'), print('\033[0m' + "(1) Acurácia
         (ACC): {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I
         (TFP):", 1 - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),
         print("(4) Precisão:", precision), print("(5) Sensibilidade (Recall):",
         sensitivity), print("(6) Especificidade:", specificity), print ("(7) F1
         Score:", F1_score), print("(8) Área sob a curva ROC
         (AUC):",DA_cv_score.mean()), print("(9) Matriz de confusão
         [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
         RESULTADO - Treino:
         (1) Acurácia (ACC): 0.8996
         (2) Erro tipo I (TFP): 0.14432721115018887
         (3) Erro tipo II (TFN): 0.05574682203389836
         (4) Precisão: 0.8655176599101833
         (5) Sensibilidade (Recall): 0.9442531779661016
         (6) Especificidade: 0.8556727888498111
         (7) F1 Score: 0.9031726933063137
         (8) Área sob a curva ROC (AUC): 0.9554890165326156
```

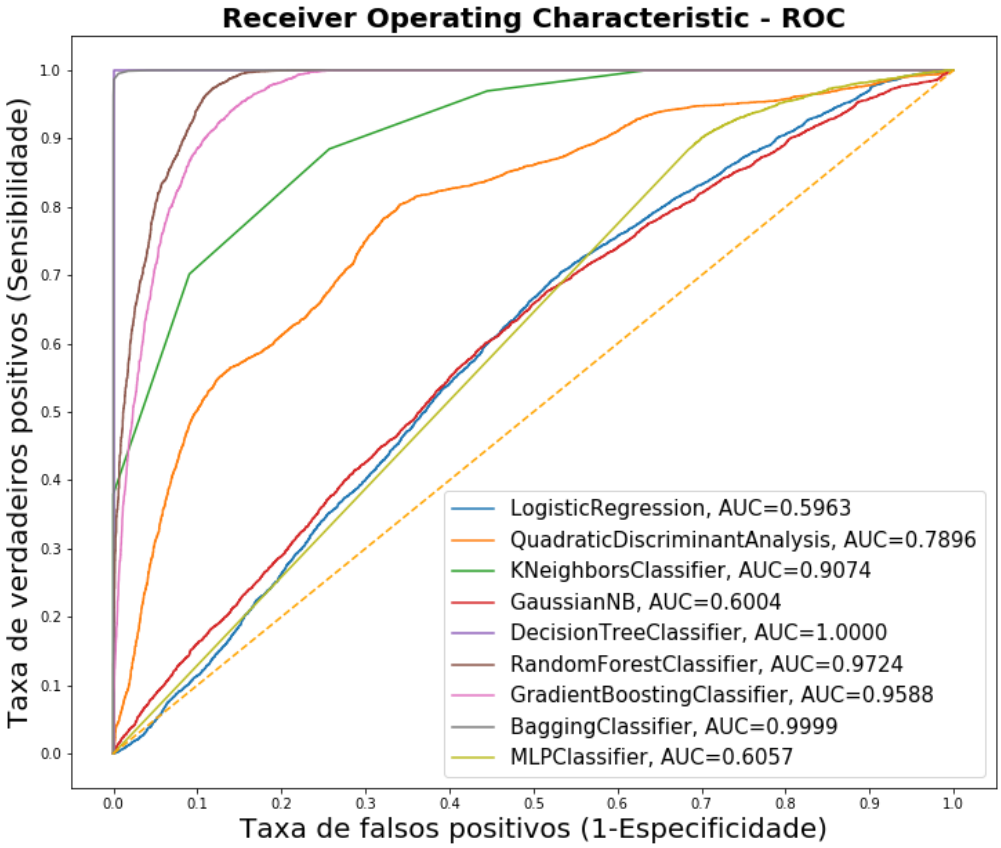
```
(9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
[[6569 1108]
 [ 421 7131]]
```

j) *Neural network - NN* (rede neural)

```
In [18]: from sklearn.ensemble import GradientBoostingClassifier
         GB = GradientBoostingClassifier(random_state=1234).fit(X_train, y_train)
         GB_cv_score = cross_val_score(GB, X, y, cv=10, scoring= 'roc_auc')
         y_pred = GB.predict(X_test)
         cm = confusion_matrix(y_test, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Treino:'), print('\033[0m' + "(1) Acurácia
         (ACC): {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I
         (TFP):", 1 - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),
         print("(4) Precisão:", precision), print("(5) Sensibilidade (Recall):",
         sensitivity), print("(6) Especificidade:", specificity), print ("(7) F1
         Score:", F1_score), print("(8) Área sob a curva ROC
         (AUC):",DA_cv_score.mean()), print("(9) Matriz de confusão
         [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
         RESULTADO - Teste:
         (1) Acurácia (ACC): 0.6069
         (2) Erro tipo I (TFP): 0.7454734922495767
         (3) Erro tipo II (TFN): 0.03482521186440679
         (4) Precisão: 0.5601752228711958
         (5) Sensibilidade (Recall): 0.9651747881355932
         (6) Especificidade: 0.25452650775042335
         (7) F1 Score: 0.7089087726123322
         (8) Área sob a curva ROC (AUC): 0.5721642312518328
         (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
         [[1954 5723]
          [ 263 7289]]
```

3. **Precisão das técnicas utilizadas** (curva *ROC* e *AUC*)

```
In [19]: from sklearn.metrics import roc_curve, roc_auc_score
         classifiers =
         [LogisticRegression(random_state=1234),QuadraticDiscriminantAnalysis(),
         KNeighborsClassifier(),GaussianNB(), DecisionTreeClassifier(random_state=1234),
         RandomForestClassifier(max_depth=9, n_estimators=100, random_state=0),
         GradientBoostingClassifier(random_state=1234),BaggingClassifier(random_state=1234
         ), MLPClassifier()]
         result_table = pd.DataFrame(columns=['classifiers', 'tfp','tvp','auc'])
         print ('\033[1m' + 'RESULTADO - Treino:')
         for cls in classifiers:
         model = cls.fit(X_train, y_train)
         yproba = model.predict_proba(X_train)[::,1]
         fpr, tpr, _ = roc_curve(y_train,  yproba)
         auc = roc_auc_score(y_train, yproba)
         result_table = result_table.append({'classifiers':cls.__class__.__name__,
                                             'tfp':fpr,
                                             'tvp':tpr,
                                             'auc':auc}, ignore_index=True)
         result_table.set_index('classifiers', inplace=True)
         fig = plt.figure(figsize=(12,10))
         for i in result_table.index:
         plt.plot(result_table.loc[i]['tfp'],
                     result_table.loc[i]['tvp'],
                     label="{}, AUC={:.4f}".format(i, result_table.loc[i]['auc']))
         plt.plot([0,1], [0,1], color='orange', linestyle='--')
         plt.xticks(np.arange(0.0, 1.1, step=0.1))
         plt.xlabel("Taxa de falsos positivos (1-Especificidade)", fontsize=20)
         plt.yticks(np.arange(0.0, 1.1, step=0.1))
         plt.ylabel("Taxa de verdadeiros positivos (Sensibilidade)", fontsize=20)
         plt.title('Receiver Operating Characteristic - ROC', fontweight='bold',
         fontsize=20)
         plt.legend(prop={'size':15}, loc='lower right')
         plt.show()
         RESULTADO - Treino:
```


Receiver Operating Characteristic - ROC
Taxa de verdadeiros positivos (Sensibilidade)
Taxa de falsos positivos (1-Especificidade)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
LogisticRegression, AUC=0.5963
QuadraticDiscriminantAnalysis, AUC=0.7896
KNeighborsClassifier, AUC=0.9074
GaussianNB, AUC=0.6004
DecisionTreeClassifier, AUC=1.0000
RandomForestClassifier, AUC=0.9724
GradientBoostingClassifier, AUC=0.9588
BaggingClassifier, AUC=0.9999
MLPClassifier, AUC=0.6057

APÊNDICE C - Resultados com dados de teste (código-fonte em *python*)

1. Preparação dos dados

a) Livrarias do *python* utilizadas

```
In [1]:   import pandas as pd
          import numpy as np
          import seaborn as sns
          from collections import Counter
          import matplotlib.pyplot as plt
          from sklearn.metrics import accuracy_score, confusion_matrix,
          classification_report, precision_score, recall_score, f1_score
          from sklearn.model_selection import cross_val_score
          %matplotlib inline
```

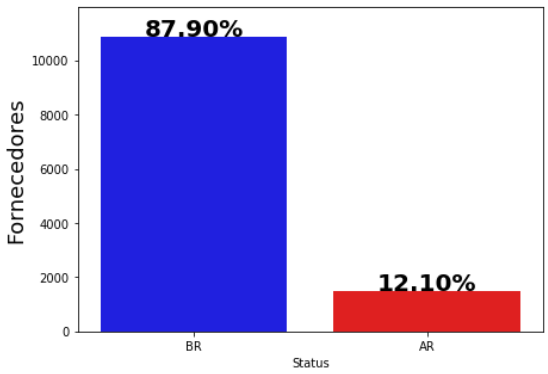
b) Importação dos dados

```
In [2]:   ##### Dados importados das bases do SIASG, CEIS e RFB
          FGF = pd.read_csv('Fornecedores do Governo Federal.csv')
          FGF.head()
Out[2]:
```

	CNPJ	Porte	Nr_CNAE_Sec	Nr_Ocr	Tp_Exis	Tot_Venda	Status
0	21446397000110	ME	47	6	391	26776.65	BR
1	9077347000107	PP	2	9	3006	50298.00	AR
2	21544098000118	PP	2	0	360	11500.00	BR
3	33457045000149	ME	0	0	17914	1602.70	BR
4	28938421000187	PP	0	2	11232	6035.00	BR

```
In [3]:   print("Conjunto de dados original, linhas e colunas, Grupo de Baixo Risco (Gp
          BR) e Grupo de Alto Risco (Gp AR):{}".format(FGF.shape))
          print ('Gp BR {0} e Gp AR {1}:')
          FGF.Status.value_counts()
          Conjunto de dados original, linhas e colunas, Grupo de Baixo Risco (Gp BR) e
          Grupo de Alto Risco (Gp AR):(12376, 7)
          Gp BR {0} e Gp AR {1}:
Out[3]:   BR    10878
          AR     1498
          Name: Status, dtype: int64
```

```
In [4]:   total = len(FGF)
          plt.figure(figsize=(7,5))
          colors = ['blue', 'red']
          g = sns.countplot(x='Status', data=FGF, palette=colors)
          g.set_ylabel('Fornecedores', fontsize=18)
          for p in g.patches:
              height = p.get_height()
              g.text(p.get_x()+p.get_width()/2., height +
          1.5,'{:1.2f}%'.format(height/total*100), ha="center", fontsize=20,
          fontweight='bold')
          plt.margins(y=0.1)
          plt.show()
```



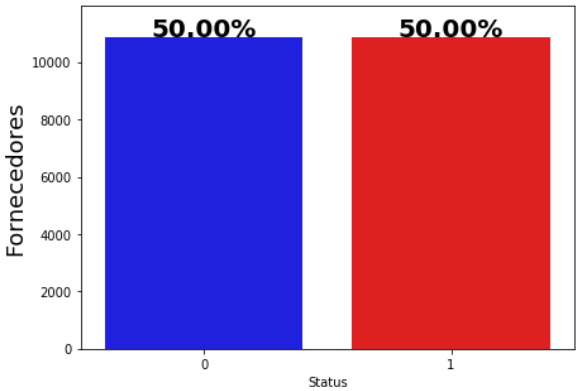
c) Pré-processamento

```
In [5]:   FGF['Porte'] = FGF['Porte'].replace({'PP': 0, 'ME': 1, 'DM': 2})
          FGF['Status'] = FGF['Status'].replace({'BR': 0,'AR': 1})
```

```
FGF = FGF.drop('CNPJ', axis=1)
X = FGF.drop('Status', axis=1)
y = FGF ['Status']
print("Conjunto de dados original, linhas e colunas:{}".format(FGF.shape))
print ('Gp BR {0} e Gp AR {1}')
FGF.Status.value_counts()
Conjunto de dados original, linhas e colunas:(12376, 6)
Gp BR {0} e Gp AR {1}
```

```
Out[5]:   0    10878
          1     1498
          Name: Status, dtype: int64
```

```
In [6]:   ### OVERSAMPLING (Aumento fitício da classe minoritária {1} deixando-a igual à
          classe majoritária {0})
          from imblearn.over_sampling import SMOTE
          smt = SMOTE()
          X, y = smt.fit_sample(X, y)
          FGF = pd.DataFrame(X, y)
          FGF['Status'] = y
          total = len(FGF)
          plt.figure(figsize=(7,5))
          colors = ['blue', 'red']
          g = sns.countplot(x='Status', data=FGF, palette=colors)
          g.set_ylabel('Fornecedores', fontsize=18)
          for p in g.patches:
              height = p.get_height()
              g.text(p.get_x()+p.get_width()/2., height +
          1.5,'{:1.2f}%'.format(height/total*100),ha="center", fontsize=20,
          fontweight='bold')
          plt.margins(y=0.1)
          print ('Conjunto de dados final, Gp BR {0} e Gp AR {1}:', format(Counter(y)))
          plt.show()
          Conjunto de dados final, Gp BR {0} e Gp AR {1}: Counter({0: 10878, 1: 10878})
```



d) Divisão da base de dados (dados de treino e dados de teste)

```
In [7]:   from sklearn.model_selection import train_test_split
          X_train, X_test, y_train, y_test = train_test_split (X, y, test_size = 0.3)
```

```
In [8]:   print('Dados de Teste, Gp BR {0} e Gp AR {1}:', format(Counter(y_test)))
          Dados de Teste, Gp BR {0} e Gp AR {1}: Counter({1: 3326, 0: 3201})
```

2. Precisão das técnicas utilizadas

a) *Discriminant analysis - DA* (análise de discriminante)

```
In [9]:   from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
          DA = QuadraticDiscriminantAnalysis().fit(X_train, y_train)
          DA_cv_score = cross_val_score(DA, X, y, cv=10, scoring= 'roc_auc')
          y_pred = DA.predict(X_test)
          cm = confusion_matrix(y_test, y_pred)
          VN = cm[0, 0]
          FP = cm[0, 1]
          FN = cm[1, 0]
          VP = cm[1, 1]
          sensitivity = VP / float(VP + FN)
          specificity = VN / (VN + FP)
          precision = VP / (VP + FP)
          F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
```

```
print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
{:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I (TFP):", 1
- specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),  print("(4)
Precisão:", precision), print("(5) Sensibilidade (Recall):", sensitivity),
print("(6) Especificidade:", specificity), print ("(7) F1 Score:", F1_score),
print("(8) Área sob a curva ROC (AUC):",DA_cv_score.mean()), print("(9) Matriz
de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
RESULTADO - Teste:
(1) Acurácia (ACC): 0.6766
(2) Erro tipo I (TFP): 0.5357700718525461
(3) Erro tipo II (TFN): 0.11906193625977146
(4) Precisão: 0.6307857911733046
(5) Sensibilidade (Recall): 0.8809380637402285
(6) Especificidade: 0.4642299281474539
(7) F1 Score: 0.7351649730272236
(8) Área sob a curva ROC (AUC): 0.7956942853037121
(9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
[[1486 1715]
 [ 396 2930]]
```

b) *Logistic regression - LR* (regressão logística)

```
In [10]: from sklearn.linear_model import LogisticRegression
         LR = LogisticRegression().fit(X_train, y_train)
         LR_cv_score = cross_val_score(LR, X, y, cv=10, scoring= 'roc_auc')
         y_pred = LR.predict(X_test)
         cm = confusion_matrix(y_test, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
         {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I (TFP):", 1
         - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),  print("(4)
         Precisão:", precision), print("(5) Sensibilidade (Recall):", sensitivity),
         print("(6) Especificidade:", specificity), print ("(7) F1 Score:", F1_score),
         print("(8) Área sob a curva ROC (AUC):",DA_cv_score.mean()), print("(9) Matriz
         de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
         RESULTADO - Teste:
         (1) Acurácia (ACC): 0.4904
         (2) Erro tipo I (TFP): 0.0
         (3) Erro tipo II (TFN): 1.0
         (4) Precisão: nan
         (5) Sensibilidade (Recall): 0.0
         (6) Especificidade: 1.0
         (7) F1 Score: nan
         (8) Área sob a curva ROC (AUC): 0.6445995654341484
         (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
         [[3201    0]
          [3326    0]]
```

c) *K-Nearest neighbors - KNN* (k-ésimo vizinho mais próximo)

```
In [11]: from sklearn.neighbors import KNeighborsClassifier
         KNN = KNeighborsClassifier(n_neighbors=9).fit(X_train, y_train)
         KNN_cv_score = cross_val_score(KNN, X, y, cv=10, scoring= 'roc_auc')
         y_pred = KNN.predict(X_test)
         cm = confusion_matrix(y_test, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
         {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I (TFP):", 1
         - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),  print("(4)
         Precisão:", precision), print("(5) Sensibilidade (Recall):", sensitivity),
         print("(6) Especificidade:", specificity), print ("(7) F1 Score:", F1_score),
         print("(8) Área sob a curva ROC (AUC):",DA_cv_score.mean()), print("(9) Matriz
         de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
```

```
RESULTADO - Teste:
(1) Acurácia (ACC): 0.7218
(2) Erro tipo I (TFP): 0.36457357075913777
(3) Erro tipo II (TFN): 0.19512928442573663
(4) Precisão: 0.6964099895941728
(5) Sensibilidade (Recall): 0.8048707155742634
(6) Especificidade: 0.6354264292408622
(7) F1 Score: 0.7467224546722455
(8) Área sob a curva ROC (AUC): 0.8032580010461994
(9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
[[2034 1167]
 [ 649 2677]]
```

d) *Naive bayes - NB* (classificador de bayes ingênuo)

```
In [12]: from sklearn.naive_bayes import GaussianNB
         NB = GaussianNB().fit(X_train, y_train)
         NB_cv_score = cross_val_score(NB, X, y, cv=10, scoring= 'roc_auc')
         y_pred = NB.predict(X_test)
         cm = confusion_matrix(y_test, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
         {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I (TFP):", 1
         - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),  print("(4)
         Precisão:", precision), print("(5) Sensibilidade (Recall):", sensitivity),
         print("(6) Especificidade:", specificity), print ("(7) F1 Score:", F1_score),
         print("(8) Área sob a curva ROC (AUC):",DA_cv_score.mean()), print("(9) Matriz
         de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
         RESULTADO - Teste:
         (1) Acurácia (ACC): 0.5499
         (2) Erro tipo I (TFP): 0.8600437363323961
         (3) Erro tipo II (TFN): 0.05562236921226704
         (4) Precisão: 0.5329148286392942
         (5) Sensibilidade (Recall): 0.944377630787733
         (6) Especificidade: 0.1399562636676039
         (7) F1 Score: 0.6813449023861171
         (8) Área sob a curva ROC (AUC): 0.6059321143624032
         (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
         [[ 448 2753]
          [ 185 3141]]
```

e) *Support vector machine - SVM* (máquina de vetor de suporte)

```
In [13]: from sklearn.svm import SVC
         SVM = SVC().fit(X_train, y_train)
         SVM_cv_score = cross_val_score(SVM, X, y, cv=10, scoring= 'roc_auc')
         y_pred = SVM.predict(X_test)
         cm = confusion_matrix(y_test, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
         {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I (TFP):", 1
         - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),  print("(4)
         Precisão:", precision), print("(5) Sensibilidade (Recall):", sensitivity),
         print("(6) Especificidade:", specificity), print ("(7) F1 Score:", F1_score),
         print("(8) Área sob a curva ROC (AUC):",DA_cv_score.mean()), print("(9) Matriz
         de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
         RESULTADO - Teste:
         (1) Acurácia (ACC): 0.5088
         (2) Erro tipo I (TFP): 0.10153077163386437
         (3) Erro tipo II (TFN): 0.8662056524353579
         (4) Precisão: 0.577922077922078
         (5) Sensibilidade (Recall): 0.1337943475646422
         (6) Especificidade: 0.8984692283661356
```

```
(7) F1 Score: 0.21728515625
(8) Área sob a curva ROC (AUC): 0.5328773114348999
(9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
[[2876  325]
 [2881  445]]
```

f) *Decision tree - DT* (árvore de decisão)

```
In [14]: from sklearn.tree import DecisionTreeClassifier
         DT = DecisionTreeClassifier(random_state=1234).fit(X_train, y_train)
         DT_cv_score = cross_val_score(DT, X, y, cv=10, scoring= 'roc_auc')
         y_pred = DT.predict(X_test)
         cm = confusion_matrix(y_test, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
         {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I (TFP):", 1
         - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),  print("(4)
         Precisão:", precision), print("(5) Sensibilidade (Recall):", sensitivity),
         print("(6) Especificidade:", specificity), print ("(7) F1 Score:", F1_score),
         print("(8) Área sob a curva ROC (AUC):",DA_cv_score.mean()), print("(9) Matriz
         de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
         RESULTADO - Teste:
         (1) Acurácia (ACC): 0.8845
         (2) Erro tipo I (TFP): 0.12246173070915334
         (3) Erro tipo II (TFN): 0.10883944678292246
         (4) Precisão: 0.8831942789034565
         (5) Sensibilidade (Recall): 0.8911605532170775
         (6) Especificidade: 0.8775382692908467
         (7) F1 Score: 0.8871595330739299
         (8) Área sob a curva ROC (AUC): 0.8906056790816603
         (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
         [[2809  392]
          [ 362 2964]]
```

g) *Random forest - RF* (floresta aleatória)

```
In [15]: from sklearn.ensemble import RandomForestClassifier
         RF = RandomForestClassifier(max_depth=9, n_estimators=100,
         random_state=0).fit(X_train, y_train)
         RF_cv_score = cross_val_score(RF, X, y, cv=10, scoring= 'roc_auc')
         y_pred = RF.predict(X_test)
         cm = confusion_matrix(y_test, y_pred)
         VN = cm[0, 0]
         FP = cm[0, 1]
         FN = cm[1, 0]
         VP = cm[1, 1]
         sensitivity = VP / float(VP + FN)
         specificity = VN / (VN + FP)
         precision = VP / (VP + FP)
         F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
         print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
         {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I (TFP):", 1
         - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),  print("(4)
         Precisão:", precision), print("(5) Sensibilidade (Recall):", sensitivity),
         print("(6) Especificidade:", specificity), print ("(7) F1 Score:", F1_score),
         print("(8) Área sob a curva ROC (AUC):",DA_cv_score.mean()), print("(9) Matriz
         de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
         RESULTADO - Teste:
         (1) Acurácia (ACC): 0.9095
         (2) Erro tipo I (TFP): 0.1387066541705717
         (3) Erro tipo II (TFN): 0.044197233914612166
         (4) Precisão: 0.8774496273806238
         (5) Sensibilidade (Recall): 0.9558027660853878
         (6) Especificidade: 0.8612933458294283
         (7) F1 Score: 0.9149517916246942
         (8) Área sob a curva ROC (AUC): 0.9587207346700861
         (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
         [[2757  444]
          [ 147 3179]]
```

h) *Bagging* - *BG* (agregação de bootstrap)

```
In [16]:  from sklearn.ensemble import BaggingClassifier
          BG = BaggingClassifier(random_state=1234).fit(X_train, y_train)
          BG_cv_score = cross_val_score(BG, X, y, cv=10, scoring= 'roc_auc')
          y_pred = BG.predict(X_test)
          cm = confusion_matrix(y_test, y_pred)
          VN = cm[0, 0]
          FP = cm[0, 1]
          FN = cm[1, 0]
          VP = cm[1, 1]
          sensitivity = VP / float(VP + FN)
          specificity = VN / (VN + FP)
          precision = VP / (VP + FP)
          F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
          print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
          {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I (TFP):", 1
          - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),  print("(4)
          Precisão:", precision), print("(5) Sensibilidade (Recall):", sensitivity),
          print("(6) Especificidade:", specificity), print ("(7) F1 Score:", F1_score),
          print("(8) Área sob a curva ROC (AUC):",DA_cv_score.mean()), print("(9) Matriz
          de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
          RESULTADO - Teste:
          (1) Acurácia (ACC): 0.9082
          (2) Erro tipo I (TFP): 0.11808809746954074
          (3) Erro tipo II (TFN): 0.06644618159951898
          (4) Precisão: 0.8914728682170543
          (5) Sensibilidade (Recall): 0.933553818400481
          (6) Especificidade: 0.8819119025304593
          (7) F1 Score: 0.9120281979732707
          (8) Área sob a curva ROC (AUC): 0.9606801736222884
          (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
          [[2823  378]
           [ 221 3105]]
```

i) *Gradient boosting* - *GB* (aumento de gradiente)

```
In [17]:  from sklearn.ensemble import GradientBoostingClassifier
          GB = GradientBoostingClassifier(random_state=1234).fit(X_train, y_train)
          GB_cv_score = cross_val_score(GB, X, y, cv=10, scoring= 'roc_auc')
          y_pred = GB.predict(X_test)
          cm = confusion_matrix(y_test, y_pred)
          VN = cm[0, 0]
          FP = cm[0, 1]
          FN = cm[1, 0]
          VP = cm[1, 1]
          sensitivity = VP / float(VP + FN)
          specificity = VN / (VN + FP)
          precision = VP / (VP + FP)
          F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
          print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
          {:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I (TFP):", 1
          - specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),  print("(4)
          Precisão:", precision), print("(5) Sensibilidade (Recall):", sensitivity),
          print("(6) Especificidade:", specificity), print ("(7) F1 Score:", F1_score),
          print("(8) Área sob a curva ROC (AUC):",DA_cv_score.mean()), print("(9) Matriz
          de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
          RESULTADO - Teste:
          (1) Acurácia (ACC): 0.9021
          (2) Erro tipo I (TFP): 0.14089347079037806
          (3) Erro tipo II (TFN): 0.056524353577871334
          (4) Precisão: 0.8743382557815548
          (5) Sensibilidade (Recall): 0.9434756464221287
          (6) Especificidade: 0.8591065292096219
          (7) F1 Score: 0.9075921908893709
          (8) Área sob a curva ROC (AUC): 0.9554890165326156
          (9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
          [[2750  451]
           [ 188 3138]]
```

j) *Neural network* - *NN* (rede neural)

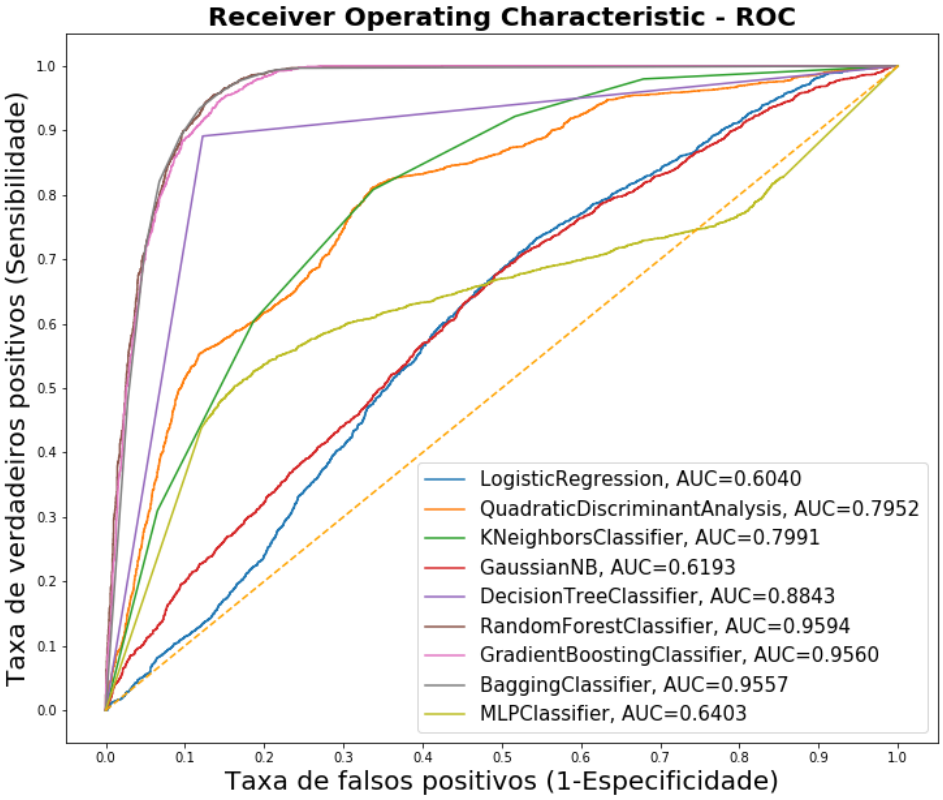
```
In [18]:  from sklearn.neural_network import MLPClassifier
          NN = MLPClassifier().fit(X_train, y_train)
          NN_cv_score = cross_val_score(NN, X, y, cv=10, scoring= 'roc_auc')
          y_pred = NN.predict(X_test)
          cm = confusion_matrix(y_test, y_pred)
```

```
VN = cm[0, 0]
FP = cm[0, 1]
FN = cm[1, 0]
VP = cm[1, 1]
sensitivity = VP / float(VP + FN)
specificity = VN / (VN + FP)
precision = VP / (VP + FP)
F1_score = 2 * (precision * sensitivity) / (precision + sensitivity)
print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
{:.4f}".format(DA.score(X_train, y_train))),  print("(2) Erro tipo I (TFP):", 1
- specificity), print("(3) Erro tipo II (TFN):", 1 - sensitivity),  print("(4)
Precisão:", precision), print("(5) Sensibilidade (Recall):", sensitivity),
print("(6) Especificidade:", specificity), print ("(7) F1 Score:", F1_score),
print("(8) Área sob a curva ROC (AUC):",DA_cv_score.mean()), print("(9) Matriz
de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:"),  print(cm)
RESULTADO - Teste:
(1) Acurácia (ACC): 0.5002
(2) Erro tipo I (TFP): 0.8066229303342706
(3) Erro tipo II (TFN): 0.20444978953698134
(4) Precisão: 0.5061208875286917
(5) Sensibilidade (Recall): 0.7955502104630187
(6) Especificidade: 0.19337706966572946
(7) F1 Score: 0.618657937806874
(8) Área sob a curva ROC (AUC): 0.5463219860418116
(9) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
[[ 619 2582]
 [ 680 2646]]
```

3. Precisão das técnicas utilizadas

a) Curva *ROC e AUC*

```
In [19]: from sklearn.metrics import roc_curve, roc_auc_score
         classifiers =
         [LogisticRegression(random_state=1234),QuadraticDiscriminantAnalysis(),
         KNeighborsClassifier(),GaussianNB(), DecisionTreeClassifier(random_state=1234),
         RandomForestClassifier(max_depth=9, n_estimators=100, random_state=0),
         GradientBoostingClassifier(random_state=1234),BaggingClassifier(random_state=1234
         ), MLPClassifier()]
         result_table = pd.DataFrame(columns=['classifiers', 'tfp','tvp','auc'])
         print ('\033[1m' + 'RESULTADO - Teste:')
         for cls in classifiers:
         model = cls.fit(X_train, y_train)
         yproba = model.predict_proba(X_test)[::,1]
         fpr, tpr, _ = roc_curve(y_test,  yproba)
         auc = roc_auc_score(y_test, yproba)
         result_table = result_table.append({'classifiers':cls.__class__.__name__,
                                             'tfp':fpr,
                                             'tvp':tpr,
                                             'auc':auc}, ignore_index=True)
         result_table.set_index('classifiers', inplace=True)
         fig = plt.figure(figsize=(12,10))
         for i in result_table.index:
         plt.plot(result_table.loc[i]['tfp'],
                  result_table.loc[i]['tvp'],
                  label="{}, AUC={:.4f}".format(i, result_table.loc[i]['auc']))
         plt.plot([0,1], [0,1], color='orange', linestyle='--')
         plt.xticks(np.arange(0.0, 1.1, step=0.1))
         plt.xlabel("Taxa de falsos positivos (1-Especificidade)", fontsize=20)
         plt.yticks(np.arange(0.0, 1.1, step=0.1))
         plt.ylabel("Taxa de verdadeiros positivos (Sensibilidade)", fontsize=20)
         plt.title('Receiver Operating Characteristic - ROC', fontweight='bold',
         fontsize=20)
         plt.legend(prop={'size':15}, loc='lower right')
         plt.show()
         RESULTADO - Teste:
```



b) Algoritmos com melhor desempenho

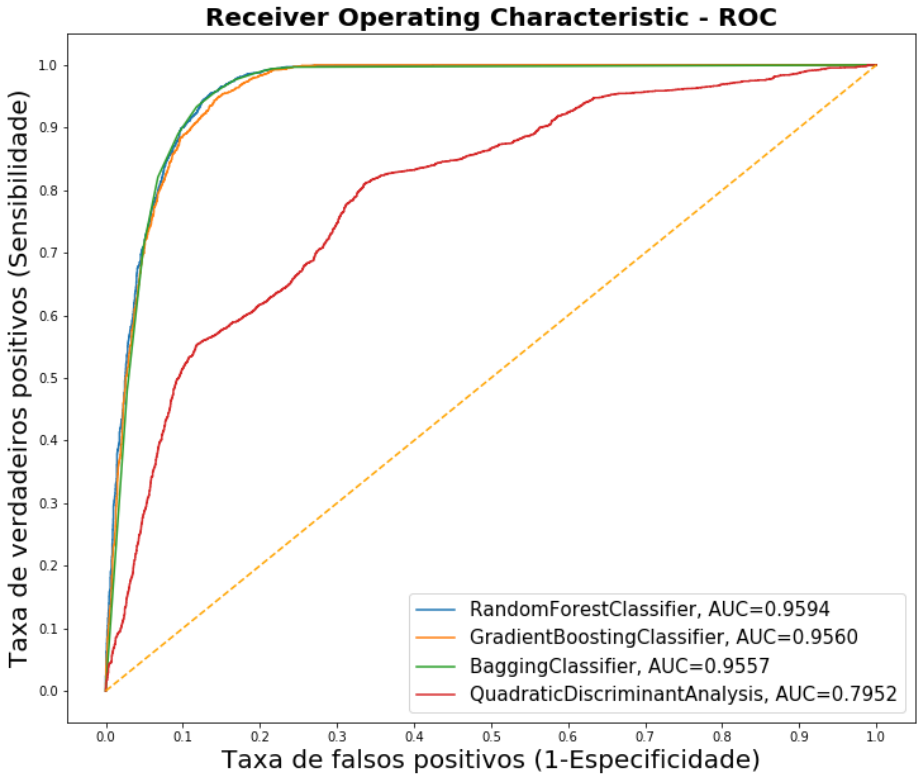
Conforme mostrado no gráfico *ROC* acima, dentre todos os classificadores analisados, os métodos *ensemble RF*, *GB* e *BG* obtiveram as melhores taxas de *AUC* (área sob a curva *ROC*). No geral, esses classificadores apresentam melhor desempenho que as técnicas *RL* e *DA*, métodos estatísticos tradicionais. Foi feito uma análise comparativa de desempenho entre a técnica *random forests – RF* (florestas aleatórias) e *Quadratic Discriminant Analysis – QDA* (análise discriminante quadrática). Informações úteis, com os detalhes dessa análise, são apresentadas a seguir:

1) Curva *ROC* dos algoritmos (*RF*, *GB*, *BG e DA*)

```
In [20]: from sklearn.metrics import roc_curve, roc_auc_score

         classifiers = [RandomForestClassifier(max_depth=9, n_estimators=100,
         random_state=0),
         GradientBoostingClassifier(random_state=1234),BaggingClassifier(random_state=1234
         ), QuadraticDiscriminantAnalysis()]
         result_table = pd.DataFrame(columns=['classifiers', 'tfp','tvp','auc'])
         print ('\033[1m' + 'RESULTADO - Teste:')
         for cls in classifiers:
         model = cls.fit(X_train, y_train)
         yproba = model.predict_proba(X_test)[::,1]
         fpr, tpr, _ = roc_curve(y_test,  yproba)
         auc = roc_auc_score(y_test, yproba)
         result_table = result_table.append({'classifiers':cls.__class__.__name__,
                                             'tfp':fpr,
                                             'tvp':tpr,
                                             'auc':auc}, ignore_index=True)
         result_table.set_index('classifiers', inplace=True)
         fig = plt.figure(figsize=(12,10))
```

```
for i in result_table.index:
plt.plot(result_table.loc[i]['tfp'],
             result_table.loc[i]['tvp'],
             label="{}, AUC={:.4f}".format(i, result_table.loc[i]['auc']))
plt.plot([0,1], [0,1], color='orange', linestyle='--')
plt.xticks(np.arange(0.0, 1.1, step=0.1))
plt.xlabel("Taxa de falsos positivos (1-Especificidade)", fontsize=20)
plt.yticks(np.arange(0.0, 1.1, step=0.1))
plt.ylabel("Taxa de verdadeiros positivos (Sensibilidade)", fontsize=20)
plt.title('Receiver Operating Characteristic - ROC', fontweight='bold',
fontsize=20)
plt.legend(prop={'size':15}, loc='lower right')
plt.show()
```

RESULTADO - Teste:



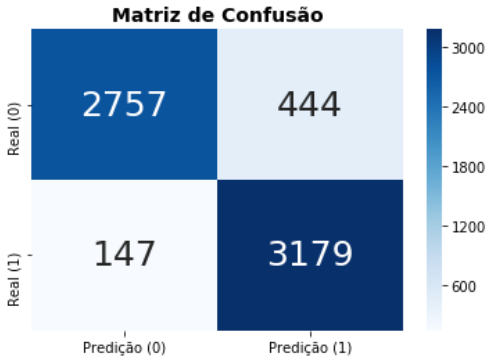
2) *Random forest - RF* (floresta aleatória)

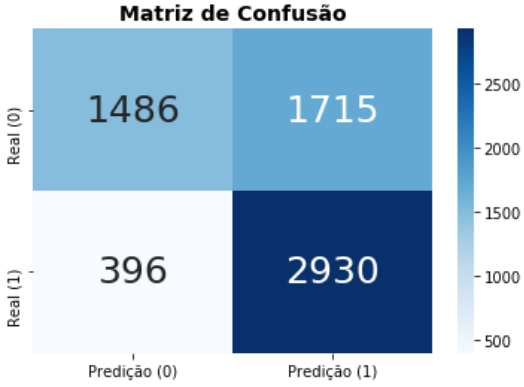
```
In [21]: from sklearn.ensemble import RandomForestClassifier
         RF = RandomForestClassifier(max_depth=9, n_estimators=100,
         random_state=0).fit(X_train, y_train)
         RF_cv_score = cross_val_score(RF, X, y, cv=10, scoring= 'roc_auc')
         cm = confusion_matrix(y_test, RF.predict(X_test))
         cm_matrix = pd.DataFrame(data=cm, columns=['Predição (0)', 'Predição
         (1)'],index=['Real (0)', 'Real (1)'])
         sns.heatmap(cm_matrix, annot=True, annot_kws={"size":25}, fmt='d', cmap='Blues')
         plt.title('Matriz de Confusão', fontweight='bold', fontsize=14)
         print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
         {:.4f}".format(RF.score(X_test, y_test)))
         print("(2) Relatório de classificação:"), print(classification_report(y_test,
         y_pred)),
         print("(3) Pontuação média da AUC:", RF_cv_score.mean()),
         print("(4) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:")
         print('\n')
```

RESULTADO - Teste:

```
(1) Acurácia (ACC): 0.9095
(2) Relatório de classificação:
              precision    recall  f1-score   support
           0       0.48      0.19      0.28      3201
```

```
           1       0.51      0.80      0.62      3326
    accuracy                           0.50      6527
   macro avg       0.49      0.49      0.45      6527
weighted avg       0.49      0.50      0.45      6527

(3) Pontuação média da AUC: 0.9587207346700861
(4) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
```



3) *Discriminant analysis - DA* (análise de discriminante)

```
In [22]: from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
         DA = QuadraticDiscriminantAnalysis().fit(X_train, y_train)
         DA_cv_score = cross_val_score(DA, X, y, cv=10, scoring= 'roc_auc')
         cm = confusion_matrix(y_test, DA.predict(X_test))
         cm_matrix = pd.DataFrame(data=cm, columns=['Predição (0)', 'Predição
         (1)'],index=['Real (0)', 'Real (1)'])
         sns.heatmap(cm_matrix, annot=True, annot_kws={"size":25}, fmt='d', cmap='Blues')
         plt.title('Matriz de Confusão', fontweight='bold', fontsize=14)
         print('\033[1m' + 'RESULTADO - Teste:'), print('\033[0m' + "(1) Acurácia (ACC):
         {:.4f}".format(DA.score(X_test, y_test)))
         print("(2) Relatório de classificação:"), print(classification_report(y_test,
         y_pred)),
         print("(3) Pontuação média da AUC:", DA_cv_score.mean()),
         print("(4) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:")
         print('\n')
         RESULTADO - Teste:
         (1) Acurácia (ACC): 0.6766
         (2) Relatório de classificação:
                       precision    recall  f1-score   support
                   0       0.48      0.19      0.28      3201
                   1       0.51      0.80      0.62      3326
            accuracy                           0.50      6527
           macro avg       0.49      0.49      0.45      6527
        weighted avg       0.49      0.50      0.45      6527

         (3) Pontuação média da AUC: 0.7956942853037121
         (4) Matriz de confusão [C00=VN;C01=FP;C10=FN;C11=VP]:
```