



Universidade de Brasília
Departamento de Estatística

Identificação de alunos com risco de evasão na Universidade de Brasília

Gustavo Durães

Trabalho de Conclusão de Curso apresentado
para obtenção do título de Bacharel em Es-
tatística.

Brasília
2018

Gustavo Durães

Identificação de alunos com risco de evasão na Universidade de Brasília

Orientador:
Prof. Dr. **Donald Matthew Pianto**

Trabalho de Conclusão de Curso apresentado
para obtenção do título de Bacharel em Es-
tatística.

Brasília
2018

Resumo

Identificação de alunos com risco de evasão na Universidade de Brasília

A evasão é um problema consideravelmente grave na Universidade de Brasília, todos os anos milhares de reais são desperdiçados com alunos que eventualmente não concluem a graduação. Tendo isso em mente, esse trabalho tem como objetivo a criação de uma ferramenta iterativa que possa identificar alunos em situação de risco a partir do histórico escolar através da utilização de técnicas estatísticas e de aprendizado de máquinas. A plataforma foi desenvolvida utilizando o software *R* e o pacote *Shiny* e pode ser utilizada por discentes e docentes. A plataforma está disponível no seguinte endereço https://gustavoduraes.shinyapps.io/Shiny_TCC/.

Palavras-chave: *Shiny*, Aprendizado de Máquinas, *R*

Sumário

RESUMO	i
1 INTRODUÇÃO	1
1.1 Objetivos	1
2 SOFTWARE	3
2.1 R	3
2.2 Sistema Operacional	4
3 FORMATAÇÃO DOS DADOS	5
3.1 Obtenção de dados adicionais	7
3.2 Criação do Fluxograma	10
4 ALGORITMO DE ESCOLHA PARA AS MATÉRIAS PRINCIPAIS	10
4.1 Agrupamento de alunos com trajetórias similares	11
5 MODELOS DE CLASSIFICAÇÃO	15
5.1 Modelos de Resposta Multinomial	15
5.2 Aprendizado de máquinas	17
6 APLICAÇÃO DE REDES NEURAI	21
6.1 Rede Neural para classificação de alunos no primeiro semestre	25
6.2 Rede Neural para classificação de alunos no segundo semestre	26
6.3 Rede Neural para classificação de alunos no terceiro semestre	27
6.4 Rede Neural para classificação de alunos no quarto semestre	28
6.5 Rede Neural para classificação de alunos no quinto semestre	29
6.6 Rede Neural para classificação de alunos no sexto semestre	30
6.7 Rede Neural para classificação de alunos no sétimo semestre	31
6.8 Rede Neural para classificação de alunos no oitavo semestre	32
6.9 Considerações	32
7 APLICATIVO SHINY	33

	iii
7.1 Primeira janela	33
7.2 Segunda janela	34
7.3 Terceira janela	34
7.4 Quarta janela	38
7.5 Quinta janela	38
7.6 Sexta e sétima janelas	39
7.7 Considerações	39
8 CONSIDERAÇÕES FINAIS	40
8.1 Trabalhos futuros	40
9 APÊNDICE	41
REFERÊNCIAS	42

Lista de Figuras

1	Exemplo da relação de cursos disponíveis no Matrícula Web.	8
2	Exemplo da página do curso de Estatística.	8
3	Exemplo de seleção do código da habilitação.	9
4	Grupos criados no aplicativo Shiny e suas respectivas frequências por semestre.	13
5	Funcionamento da conexão entre a camada de entrada com 12 variáveis com um neurônio de saída	18
6	Exemplo de Rede neural com 12 variáveis de entrada, duas camadas ocultas e uma camada de saída com 4 classes	19
7	Diferença entre as funções Sigmoide e Tangente hiperbólica.	22
8	Diferença entre as derivadas de primeira ordem das funções Sigmoide e Tangente hiperbólica.	22
9	Rede Neural para classificação de alunos no primeiro semestre.	25
10	Rede Neural para classificação de alunos no segundo semestre.	26
11	Rede Neural para classificação de alunos no terceiro semestre.	27
12	Rede Neural para classificação de alunos no quarto semestre.	28
13	Rede Neural para classificação de alunos no quinto semestre.	29
14	Rede Neural para classificação de alunos no sexto semestre.	30
15	Rede Neural para classificação de alunos no sétimo semestre.	31
16	Rede Neural para classificação de alunos no oitavo semestre.	32
17	Layout da primeira janela do aplicativo.	33
18	Layout da segunda janela do aplicativo	34
19	Exemplo de Layout da terceira janela do aplicativo - Fluxograma	35
20	Exemplo de Layout da terceira janela do aplicativo - Grupos	36
21	Exemplo de Layout da terceira janela do aplicativo - Avaliação da precisão . . .	36
22	Exemplo de Layout da terceira janela do aplicativo - Ajuste dos modelos . . .	37
23	Exemplo de Layout da terceira janela do aplicativo - Ajuste dos modelos . . .	37
24	Exemplo de Layout da quinta janela do aplicativo	38
25	Exemplo de Layout da quinta janela do aplicativo	39

Lista de Tabelas

1	Trajectoria Considerada	12
2	Exemplo de histórico escolar	12
3	Trajectoria formatada utilizada para as análises.	12

1 INTRODUÇÃO

A evasão universitária no sistema educacional público brasileiro é um problema que assola as instituições de ensino superior, causando milhões de reais em prejuízos aos cofres públicos. O presente trabalho tem por objetivo fazer uma análise probabilística e de classificação a fim de agrupar indivíduos em grupos distintos de acordo com sua proficiência. Desta maneira será possível identificar e classificar discentes em situação de risco- ou seja, aqueles que têm alta probabilidade de não concluírem suas habilitações na Universidade de Brasília (UnB).

Os resultados dessa análise serão disponibilizados em um aplicativo com interface interativa e customizável baseada no pacote Shiny, o qual se encontra disponível no software R. O referido aplicativo proporcionará informações individualizadas, baseando-se no histórico de cada estudante.

A análise será baseada no banco de dados fornecido pelo Decanato de Ensino de Graduação (DEG), onde estão contidas as informações acerca de 81.352 alunos distintos do período do primeiro semestre do ano 2000 ao primeiro semestre do ano de 2015. O banco de dados é composto por 16 variáveis, entretanto, nessa análise serão levadas em consideração 7 delas. As variáveis utilizadas oferecem informação sobre a identificação do aluno (composta pela matrícula e CPF), a disciplina cursada, a menção obtida, o período de ingresso do aluno na universidade, o período em que foi cursada a disciplina e a situação referente à conclusão do curso (e.g. Formado, Desligado, Ativo).

1.1 Objetivos

Este trabalho tem como objetivo final criar uma aplicação na plataforma Shiny, utilizando o software R, que tenha funcionalidades para auxiliar a comunidade acadêmica a analisar o perfil de cada habilitação disponível na Universidade de Brasília, através da quantificação da probabilidade de formatura. Pretende-se classificar os estudantes de acordo com situação qualificada como de risco ou não.

Para além disso, espera-se que o resultado final do presente trabalho auxilie a Universidade de Brasília a criar mecanismos que não agregarão nenhum tipo de ônus à mesma - tais como a reestruturação do fluxo - e que poderão diminuir a evasão no ensino superior, reduzindo, desta forma, o rombo existente nos cofres públicos. Finalmente, visa-se

que a ferramenta seja utilizada por pessoas que aspiram ser discentes da UnB, a fim de que consultem as habilitações disponíveis na universidade, levando em consideração, para além do fluxo, outras variáveis, o que provavelmente diminuirá a evasão.

2 SOFTWARE

Todos os procedimentos realizados neste trabalho foram executados na linguagem *R* utilizando a interface *Rstudio*, tanto a linguagem quanto a interface são de uso livre.

2.1 *R*

A utilização do Software *R* foi complementada pelo uso de alguns pacotes, são eles:

- *Tidyverse*, que é um compilado de pacotes utilizados para transformações de dados e representações gráficas. Esse conjunto de pacotes tem como característica peculiar a utilização de uma sintaxe própria e estrutura harmônica entre suas dependências.
- *rvest*, que é um pacote utilizado para coletar dados disponíveis na internet, essa técnica é usualmente chamada de *web scraping*. O pacote foi feito pelo mesmo autor do *Tidyverse* e compartilha de suas características.
- *strungr*, que é um pacote contido no *Tidyverse* e tem como objetivo modificar, identificar e extrair informações a partir da utilização de expressões regulares.
- *igraph*, que é um pacote composto de várias ferramentas de análise de redes, utilizado para construir, analisar e representar graficamente essas estruturas.
- *TraMineR*, que é um pacote utilizado para minerar, descrever e visualizar sequências de estados ou eventos.
- *Tensorflow*, que é um pacote utilizado como *backend* para aplicações que utilizam de computação numérica para executar atividades de aprendizado de máquinas
- *Keras*, que é uma interface de programação de aplicações (API) utilizada para implementar modelos de aprendizagem profunda, utilizando o *Tensorflow* como base.
- *Shiny*, que é um pacote que possibilita a construção de aplicativos interativos.
- *pdftools*, que é um pacote que tem como objetivo ler informações de arquivos no formato *pdf*.

- *kableExtra*, que é um pacote utilizado para criar tabelas customizáveis em formato *html*, muito útil para aplicações no aplicativo *Shiny*.
- *kableExtra*, que é um pacote utilizado para criar tabelas customizáveis em formato *html*, muito útil para aplicações no aplicativo *Shiny*.
- *forcats*, que é um pacote utilizado para a transformação de variáveis categóricas e fatores.
- *plotly*, que é um pacote utilizado para a criação de gráficos interativos, provavelmente o pacote utilizado nesta análise que tem mais sintonia com as funcionalidades do *Shiny*, tendo em vista que o foco do trabalho é a construção de uma plataforma interativa.

2.2 Sistema Operacional

Neste trabalho foram usados dois sistemas operacionais, *Windows 10 Home Edition* e *macOS High Sierra Versão 10.1.36*.

3 FORMATAÇÃO DOS DADOS

Inicialmente, é importante frisar que para a extração e tratamento de dados fornecidos pelo Decanato de Graduação da Universidade de Brasília (DEG) foi utilizado o *Software open source R*. No que diz respeito ao produto dos dados transformados, estes foram implementados no pacote *Shiny* do mesmo software.

O banco de dados utilizado contém 16 variáveis, as quais informam o modo de ingresso na universidade, o sistema de acesso, o curso, turno, habilitação, os respectivos semestres de ingresso e egresso, o motivo do egresso da universidade, as disciplinas cursadas e seus respectivos conceitos recebidos; além de duas variáveis de identificação criptografadas que informam a matrícula e o Código de Pessoa Física do estudante. Por consequência da criptografia, algumas observações foram usadas como semente da randomização e sua informação foi perdida.

Foi criada uma variável que indica o semestre letivo que certo aluno cursou determinada matéria através da combinação das informações do ano de ingresso de cada aluno e do semestre em que a matéria foi cursada.

Para otimizar a análise, disciplinas cursadas no período de verão foram categorizadas como cursadas no primeiro semestre letivo do respectivo ano. Decidiu-se por essa abordagem pois, tendo em vista que foi analisada a trajetória dos estudantes na Universidade, a adição do verão como semestre corrente adicionaria um período a mais a cada ano observado.

Além disso, quaisquer circunstâncias que culminaram na não aprovação em uma disciplina foram categorizadas como não conclusão. Essa transformação na variável corrobora com a parcimônia da representação gráfica da performance dos alunos, dado que resulta em apenas quatro categorias de conceitos possíveis. Seguindo o mesmo raciocínio, quaisquer circunstâncias que resultam na não formatura, salvo o evento de falecimento do estudante, foram consideradas como uma categoria única com rótulo "Não Formatura".

A informação acerca do histórico de cada aluno foi analisada em um formato sequencial, considerando como unidade de tempo o semestre letivo. Entretanto, não foram utilizadas todas as matérias cursadas no período de referência, mas sim as disciplinas chaves que podem ser definidas pelo usuário. Optou-se também pela não utilização das informações

de disciplinas cursadas em outra graduação, dado a impossibilidade da realização da paridade destas disciplinas com a trajetória estabelecida, uma vez que é perdida a noção sequencial do progresso do aluno ao longo do curso.

Os fluxos utilizados foram definidos através das informações disponíveis na plataforma de matrículas da Universidade de Brasília, o Matrícula Web. Os dados referentes aos fluxos foram coletados por meio de um algoritmo que executa a coleta das informações abertamente disponíveis, por intermédio da extração das informações desejadas, utilizando o código fonte de página. Para tal atividade foi utilizado o pacote *rvest* do R. É sabido que alguns cursos obtiveram mudanças recentes em seus currículos e, em virtude disso, poderiam apresentar problemas de incompatibilidade de informações pois, como já foi mencionado, o banco de dados utilizados no presente trabalho contém informações somente até o ano de 2015. Ainda, alguns cursos não apresentam registros de alunos que chegaram a concluir o curso no novo fluxograma, sendo o curso de Estatística um exemplo notório, já que habilitação sofreu uma mudança de currículo que foi efetivada a partir do primeiro semestre de 2014. Com o intuito de contornar essa situação, partindo da suposição que a mudança de currículos e fluxos não afeta o cerne dos cursos, optou-se por escolher quatro matérias chave em cada um dos primeiros quatro semestres de cada um dos cursos da graduação, sendo que a escolha dessas matérias é definida em um primeiro momento por um algoritmo mas também pode ser feita pelo usuário da plataforma.

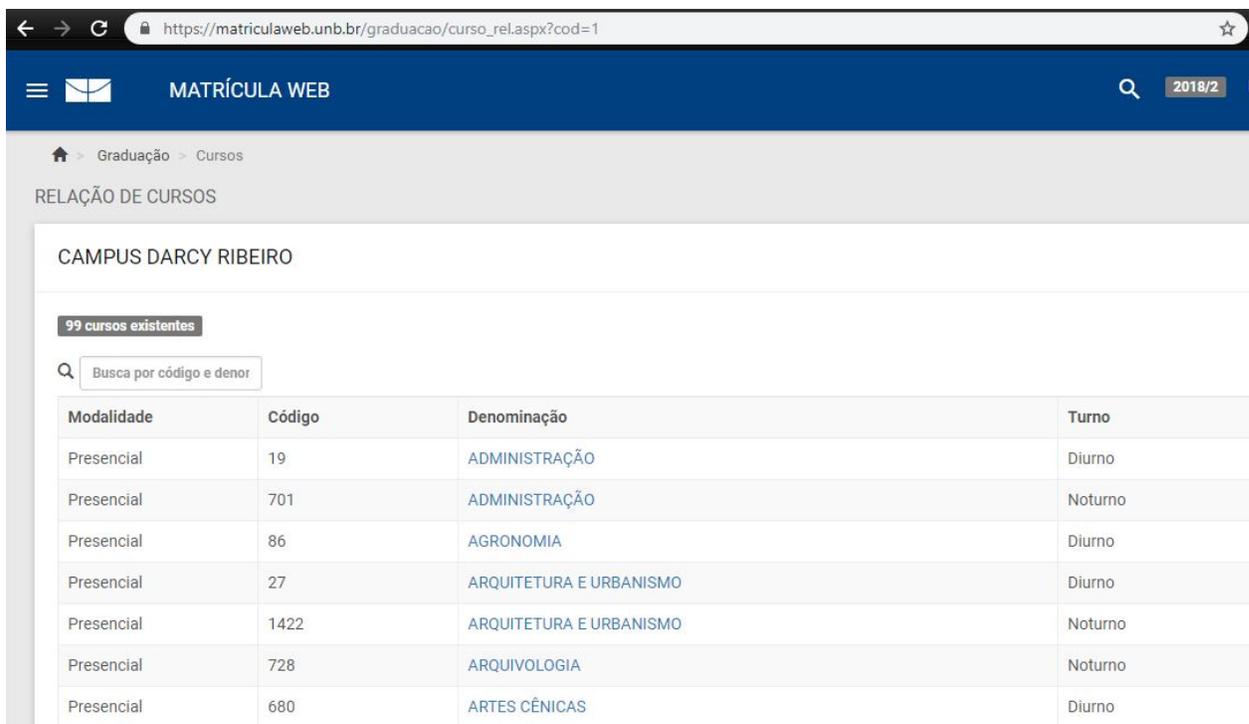
Cada habilitação disponível no nível de Graduação na Universidade de Brasília foi caracterizada por um nome e código. Para os fins deste trabalho, foi considerado o agrupamento de algumas habilitações com características similares. Por exemplo, os discentes matriculados na licenciatura em matemática no período noturno e diurno foram considerados como uma população, entretanto, alunos matriculados no a Bacharelado e Licenciatura em matemática foram analisados separadamente. A respeito do exposto, acredita-se que a realização da análise por agrupamento de habilitações semelhantes foge das possibilidades para a realização do estudo em questão.

3.1 Obtenção de dados adicionais

O banco de dados fornecido pelo Departamento de Ensino de Graduação (DEG) não continha algumas informações necessárias para a análise, tais como o código da matéria, a lista de matérias obrigatórias para cada habilitação e seus respectivos pré requisitos. Ademais, a lista também não possuía informações acerca da oferta de disciplinas. Para obter os dados não fornecidos pelo DEG mas necessários à análise, foi utilizado o pacote *'rvest'* disponível no R, o qual contém funções que coletam os dados diretamente do código *html* da página *web*. Considerando as funções disponíveis no pacote, foi utilizada a função *'read_html()'* para fazer o download e salvar o código *html* de uma determinada página. Após o uso da função dita anteriormente, foi utilizada a função *'html_nodes()'* para extrair a informação desejada de uma seção específica do código *'html'*, em seguida, foi utilizada a função *'html_text()'* para remover elementos da linguagem *html* da informação desejada.

O procedimento de coleta utilizado consistiu em quatro etapas. Primeiramente, foram selecionados todos os cursos listados na plataforma *'Matrícula Web'* disponíveis no campus Darcy Ribeiro. A listagem de cursos se dá em forma de tabela com quatro variáveis, sendo elas: modalidade, composta das categorias "Presencial" e "Distância"; código, composta pelo código do curso- que é diferente do código de habilitação, já que cada curso pode oferecer diferentes tipos de habilitações-; denominação, que denota o nome da cada habilitação e, finalmente, turno- que informa o turno de cada curso.

A partir do código de cada curso foram selecionadas todas as habilitações disponíveis no Campus e, posteriormente, tendo em vista que convenientemente os endereços *url* disponíveis do Matrícula Web são padronizados, foi possível a extração dos respectivos fluxos e currículos. Após selecionadas as habilitações, foram selecionados os fluxos e o currículos para cada uma delas e foi feita a interseção das matérias presentes nos dois bancos de dados resultantes da extração para concatenar um fluxograma construído apenas de disciplinas obrigatórias. A etapa final desse processo foi realizada de forma similar, na qual foi selecionado o código das disciplinas resultantes dos fluxogramas de matérias obrigatórias. Utilizando as técnicas já descritas foi extraído o nome de cada disciplina e seus respectivos pré-requisitos.



https://matriculaweb.unb.br/graduacao/curso_rel.aspx?cod=1

MATRÍCULA WEB

2018/2

Graduação > Cursos

RELAÇÃO DE CURSOS

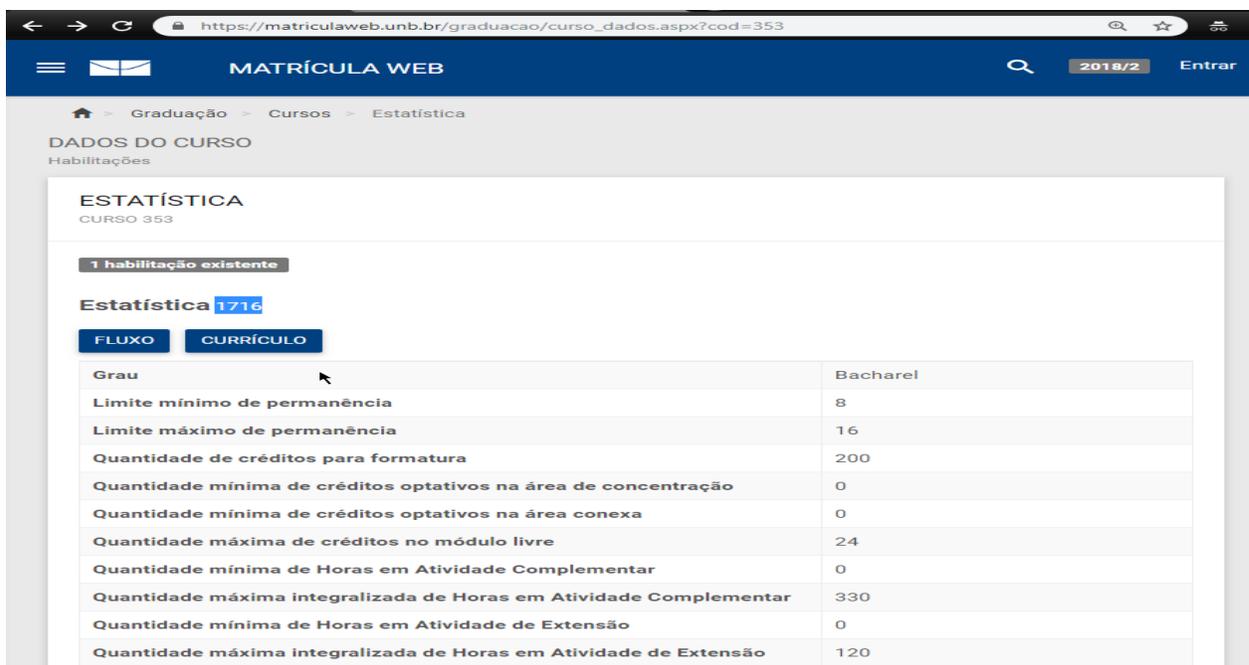
CAMPUS DARCY RIBEIRO

99 cursos existentes

Busca por código e denom

Modalidade	Código	Denominação	Turno
Presencial	19	ADMINISTRAÇÃO	Diurno
Presencial	701	ADMINISTRAÇÃO	Noturno
Presencial	86	AGRONOMIA	Diurno
Presencial	27	ARQUITETURA E URBANISMO	Diurno
Presencial	1422	ARQUITETURA E URBANISMO	Noturno
Presencial	728	ARQUIVOLOGIA	Noturno
Presencial	680	ARTES CÊNICAS	Diurno

Figura 1 – Exemplo da relação de cursos disponíveis no Matrícula Web.



https://matriculaweb.unb.br/graduacao/curso_dados.aspx?cod=353

MATRÍCULA WEB

2018/2 Entrar

Graduação > Cursos > Estatística

DADOS DO CURSO

Habilitações

ESTATÍSTICA

CURSO 353

1 habilitação existente

Estatística 1716

FLUXO CURRÍCULO

Grau	Bacharel
Limite mínimo de permanência	8
Limite máximo de permanência	16
Quantidade de créditos para formatura	200
Quantidade mínima de créditos optativos na área de concentração	0
Quantidade mínima de créditos optativos na área conexa	0
Quantidade máxima de créditos no módulo livre	24
Quantidade mínima de Horas em Atividade Complementar	0
Quantidade máxima integralizada de Horas em Atividade Complementar	330
Quantidade mínima de Horas em Atividade de Extensão	0
Quantidade máxima integralizada de Horas em Atividade de Extensão	120

Figura 2 – Exemplo da página do curso de Estatística.

O exemplo explícito na Figura 2 ilustra um curso que contém apenas uma habilitação, referenciado pelo endereço

'https://matriculaweb.unb.br/graduacao/curso_dados.aspx?cod=19'. O endereço *url* é fixo até a especificação do componente numérico da parte final do endereço, o qual corresponde aos códigos do curso exemplificados na Figura 1.

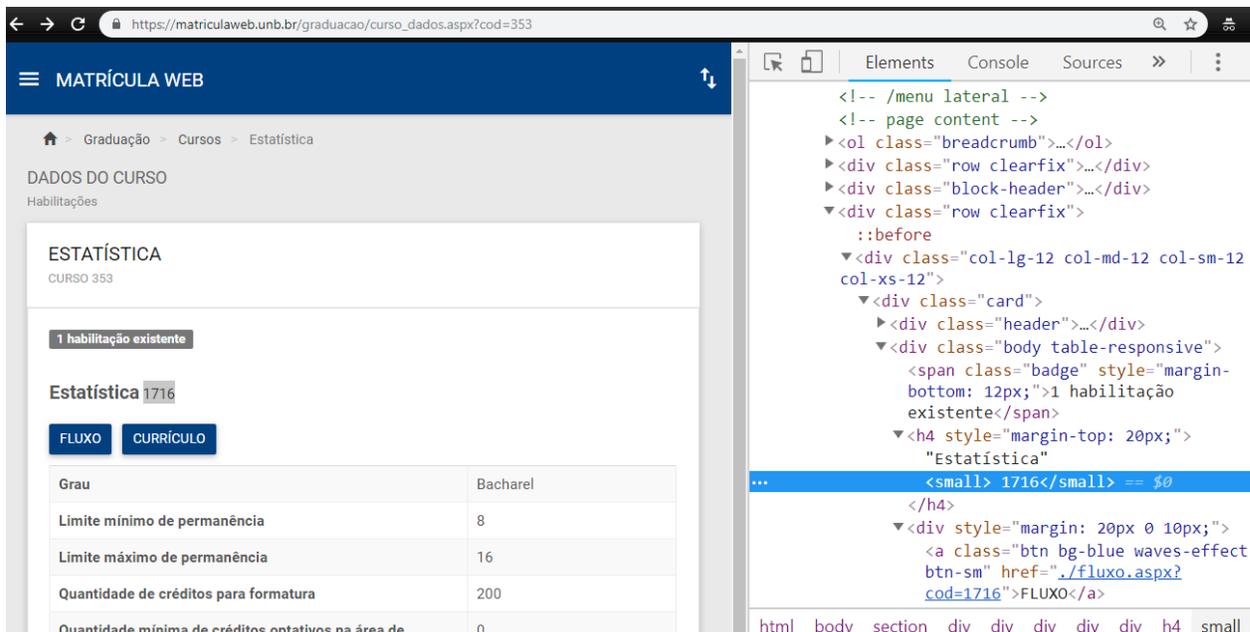


Figura 3 – Exemplo de seleção do código da habilitação.

A partir da seleção do referenciamento do código *html* destacado em azul, é possível extrair o código da habilitação '1716' lendo a página com a função `read_html()`, utilizando a função `'html_nodes()'` tendo o referenciamento como argumento e a função `'html_text()'` para resgatar código. Esse procedimento foi executado para todas as habilitações.

Adicionalmente, quando necessário, foram utilizadas expressões regulares para filtrar a informação dos códigos *html* e organizar o novo banco de dados.

Utilizando esse método foram extraídas listas de disciplinas obrigatórias, seus respectivos pré-requisitos e posição no fluxo proposto para todas as 133 habilitações existentes no campus Darcy Ribeiro da Universidade de Brasília em nível de graduação. Ademais, foi coletada a lista de disciplinas ofertadas por todos os departamentos do campus Darcy Ribeiro contendo informações acerca do horário e dia das respectivas aulas.

3.2 Criação do Fluxograma

A partir das informações coletadas foi possível criar o fluxograma de cada curso para poder ser definida uma sequência de matérias que compõe a trajetória a ser analisada. Uma forma natural de representar essa estrutura é através de um grafo. Podemos definir um grafo como uma estrutura composta por objetos denominados vértices (V_i) e seus respectivos pares ordenados. As conexões entre os arranjos de vértices são denominadas arestas (E_{ij}). Utilizando o pacote *igraph*, é possível construir um grafo que tem como vértices as disciplinas obrigatórias que compõem o fluxo e as arestas são definidas pelas relações diretas de pré-requisitos entre as disciplinas.

4 ALGORITMO DE ESCOLHA PARA AS MATÉRIAS PRINCIPAIS

Como citado no tópico acima, foi definido um algoritmo com o objetivo de definir a trajetória principal a ser analisada. A dimensão da trajetória foi fixada como quatro matérias, entretanto, também foi possível escolher cinco matérias para compor trajetória. A escolha da dimensão quatro para a trajetória analisada foi feita com base no argumento de que a trajetória de disciplinas cursadas por alunos pode ter grande variabilidade, impossibilitando a comparação e, além disso, a modificação de currículos ao longo do tempo também resulta na impossibilidade de comparação.

As operações executadas podem ser sintetizadas a partir do *pseudocódigo* abaixo.

Algoritmo 1: Escolha das matérias principais

Entrada: Grafo do fluxograma

- 1 Mensurar a quantidade de arestas de entrada e saída de todas as disciplinas obrigatórias do fluxo.
- 2 Selecionar as três disciplinas com maior quantidade de arestas.
- 3 Traçar todas as trajetórias possíveis que contêm as disciplinas selecionadas.
- 4 Selecionar as trajetórias de tamanho quatro de forma que exista apenas uma disciplina de cada semestre na trajetória.

Saída: Tronco Principal

Este algoritmo contornou a maioria os problemas explicitados anteriormente, pois o tronco escolhido nunca resultará em um agrupamento com erros já que ele intersecta informações acerca das matérias existentes no banco de dados com informações dos alunos e no banco de dados contendo informações dos fluxos.

4.1 Agrupamento de alunos com trajetórias similares

Após efetuado o tratamento dos dados para obter a forma desejada, foi utilizado um algoritmo de agrupamento divisivo hierárquico através do método da variância mínima de Ward que constrói grupos buscando a variância mínima intra-grupo e a variância máxima entre os grupos. Neste caso específicos foram criados quatro grupos pois havia uma quantidade limitada de alunos e a criação de muitos grupos com poucos elementos acarretaria em pequenas amostras e dificultaria a análise. O agrupamento foi feito utilizando o pacote *TraMineR*, que através da análise das trajetórias em formato sequencial cria clusters com trajetórias similares. Para maximizar a eficiência do agrupamento foi definida uma matriz de custos entre os diferentes conceitos de cada matéria e entre os estados de "Cadeia Finalizada", "Não Formatura" e "Formatura", a matriz de custos definida encontra-se no Apêndice 2 no aplicativo Shiny.

O procedimento para o tratamento dos banco de dados visando adequar a estrutura dos dados á forma requisitada pelo pacote *TraMineR* se deu da seguinte forma: Primeiramente foram selecionadas as quatro disciplinas principais que compõem o tronco escolhido, em seguida, as informações foram agrupadas de acordo com o CPF dos discentes e foi feita a interseção ds matérias cursadas por aluno e as matérias selecionadas no tronco principal. Após feito isso, selecionou-se o semestre em que o aluno atingiu seu estado final no curso, que pode ser de Formatura ou Não Formatura.

Por fim, foram tomadas algumas medidas para completar as lacunas que aparecem naturalmente no banco de dados, tendo em vista que apenas uma fração das informações de cada discente é selecionada. Foi atribuído aos semestres letivos após o estado final um rótulo igual ao estado final, já que o estado final de um aluno é imutável caso não sejam considerados alunos que são reintegrados. Entre o período de conclusão da quarta matéria do tronco até o estado final foi atribuído o rótulo de "Cadeia Finalizada", tendo em vista que o aluno curso e obteve em aprovação em todas as matérias do tronco escolhido. Após

executados estes procedimentos as únicas lacunas restantes foram oriundas de alunos que não cursaram a disciplina no período previsto no fluxo. Para estes casos atribui-se, para os casos nos quais o estudante não tinha conseguido aprovação na última matéria válida cursada, o rótulo de matéria inconcluído, e, caso o discente tivesse conseguido a aprovação mas a lacuna exista foi considerada como a matéria seguinte inconcluído.

Segue abaixo um exemplo do procedimento considerando o histórico de um aluno fictício, descrito na Tabela 2, aluno de estatística, selecionando apenas as matérias descritas na Tabela 1:

Tabela 1 – Trajetória Considerada

Matéria 1	Matéria 2	Matéria 3	Matéria 4
Cálculo 1	Cálculo 2	Cálculo 3	Inferência Estatística

Tabela 2 – Exemplo de histórico escolar

Disciplina	Conceito	Período	Estado Final	Semestre de estado final	CPF
Cálculo 1	MM	1	Formatura	10	X
Cálculo 2	INCONCLUÍDO	2	Formatura	10	X
Cálculo 2	MM	3	Formatura	10	X
Calculo 3	MS	5	Formatura	10	X
Inferência Estatística	MS	7	Formatura	10	X

Após executado o tratamento das informações, obteremos um uma observação da matriz de trajetórias, em forma vetorial, descrita na Tabela 3.

Tabela 3 – Trajetória formatada utilizada para as análises.

CPF	Sem. 1	Sem. 2	Sem.3	Sem. 4	Sem. 5	Sem.6	Sem. 7	Sem. 8	Sem. 9	Sem. 10
X	C1 MM	C2 INC.	C2 MM	C3 INC.	C3 MS	INF INC	INF MS	C.F.	C.F	Formatura

Observações cujo as trajetórias não puderam ser determinadas por esse procedimento não foram consideradas na análise. Além disso, estudantes cujo período em que foi

¹Pra possibilitar a representação tabular, foram feitas abreviações nos nomes das matérias e em alguns rótulos. "Cálculo 1"foi abreviado para "C1", "Cálculo 2"foi abreviado para "C2", "Cálculo 3"foi abreviado para "C3", "Inferência estatística"foi abreviada para "INF", "Inconcluído"foi abreviado para "INC."e "Cadeia Finalizada"foi abreviada para "C.F."e "Semestre"foi abreviado para "Sem."

cursada a disciplina foi anterior ao ingresso, isto é, alunos que estavam cursando uma segunda graduação ou mudaram de curso, também não foram considerados pelo mesmo motivo do caso anterior.

Após a reestruturação dos dados foram realizados alguns procedimentos. Primeiramente foi definida a sequência de estados das disciplinas analisada, o segundo passo do procedimento consistiu em mensurar as diferenças entre as trajetórias - ou sequências- observadas e tomadas duas-a-duas. Para mensurar as dissimilaridades foi utilizado o método *Optimal Matching* (OM) que busca o custo mínimo de seleção e substituição dos estados da trajetória a partir de uma matriz de custos pré determinada levando em consideração ao valor numérico da nota que é atribuída a respectiva menção e a progressão no tempo. A matriz de custos utilizada encontra-se no Apêndice 2 do aplicativo *Shiny*.

A Figura abaixo ilustra um exemplo de quatro grupos criados a partir da trajetória considerada da Tabela 1, considerando a matriz de custos definida anteriormente e o procedimento descrito e exemplificado na Tabela 3. No aplicativo *Shiny* é possível escolher entre a criação de 2 a 4 grupos.

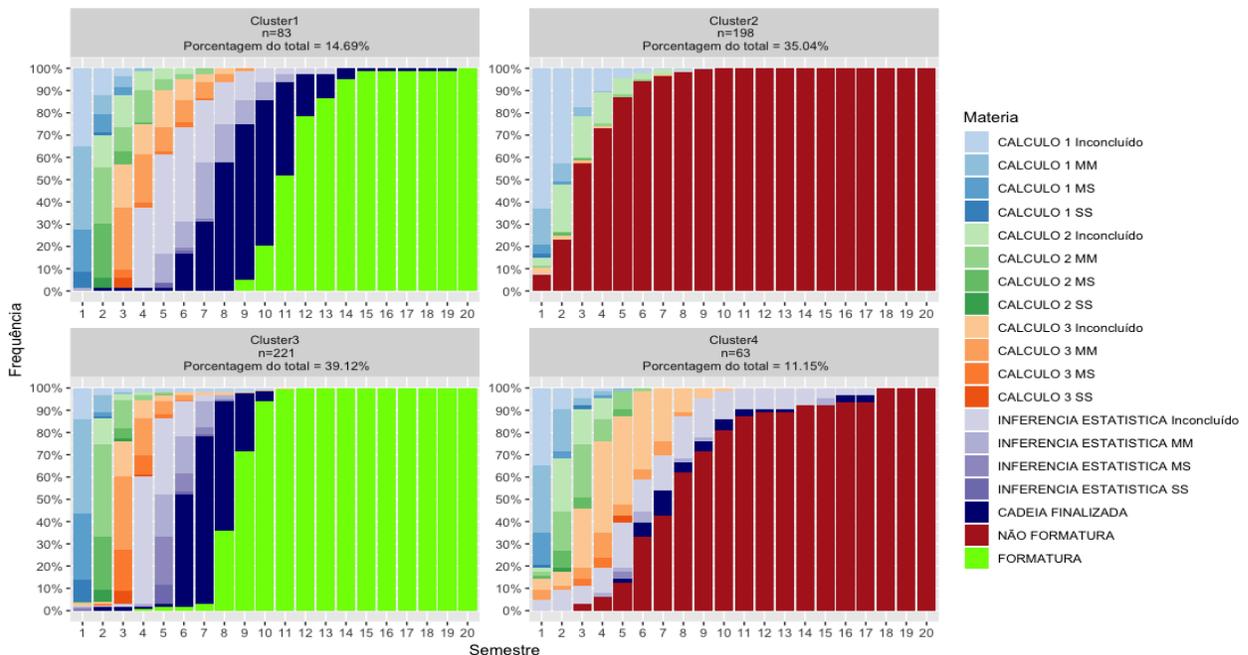


Figura 4 – Grupos criados no aplicativo Shiny e suas respectivas frequências por semestre.

Percebe-se que existe uma clara separação nos grupos de acordo com os es-

tados finais de "Formatura" e a "Não Formatura", isso ocorre pelo fato do "custo" definido previamente entre esses dois estados ser muito alto. A partir dessas observações é natural a consideração de que independentemente do tronco principal selecionado exista sempre uma segregação completa de alunos que se formam ou não.

5 MODELOS DE CLASSIFICAÇÃO

Após o agrupamento das observações do banco de dados foi realizado um procedimento para classificar novas observações de acordo com os grupo pré-estabelecidos anteriormente. Existem dois tipos de modelagem que poderiam ser utilizadas para a classificação, uma delas é a modelagem estatística usual através de modelos estatísticos de Resposta Multinomial e a outra é alguma aplicação de Inteligência Artificial, muito utilizada atualmente para problemas de classificação. Neste estudo foi utilizada a segunda opção de modelagem tendo em vista que a utilização do Modelo Linear Generalizado seria muito mais onerosa, pois teria que ser feita uma análise de diagnóstico para cada modelo gerado e a interpretação dos resultados poderia não ser trivialmente explicada dependendo da função de ligação utilizada. Nas duas seções seguintes há uma breve explicação de cada modelagem, fundamentando a escolha da aplicação utilizada, já que modelos de aprendizado de máquinas utilizam algumas funções matemáticas que podem ser melhor compreendidas no contexto de modelos lineares.

5.1 Modelos de Resposta Multinomial

Um Modelo de Resposta Multinomial é um caso particular de um Modelo Linear Generalizado e tem como resposta variáveis categóricas distintas. Uma definição possível de modelo linear generalizado (Agresti, 2015) é que estes são modelos cujo a variável resposta $f(y)$ segue uma distribuição com forma

$$f(y|\theta) = \exp \{ \theta t(y) - b(\theta) + c(y) \} \quad (1)$$

Onde temos que a função de ligação canônica $g(\mu) = \theta$, e componente sistemático, também denominado de estatística suficiente, $t(y)$. O objetivo do Modelo Linear Generalizado é estimar o valor médio do componente sistemático através da função de ligação.

Aplicando essa definição ao problema atual, que se pode ser descrito através de uma distribuição Multinomial com forma:

$$f(y|\pi_1, \dots, \pi_k) = \frac{n!}{y_1! y_2! \dots y_k!} \exp \left\{ \sum_{k=1}^K y_k \log(\pi_k) \right\} \quad (2)$$

que é uma distribuição que pertence à família exponencial, logo, estaríamos tratando de um Modelo Linear Generalizado com resposta multinomial que possui função de ligação canônica

e função de resposta canônica, respectivamente:

$$g(\pi_k) = \log \left(\frac{\theta_k}{1 - \sum_{k=1}^{K-1} \theta_k} \right) \quad (3)$$

$$\pi_k = \frac{\exp\{\theta_k\}}{\exp \left\{ \sum_{k=1}^K \theta_k \right\}} \quad (4)$$

A equação (4) é conhecida como função *softmax* ou logito multinomial. Além disso, um caso particular da função *softmax* é a função sigmoide, onde são consideradas apenas duas categorias ao invés de K categorias. Ambas as funções são muito utilizadas como funções de ativação em modelos de aprendizado de máquinas.

5.2 Aprendizado de máquinas

Existem uma infinidade de modelos que são ditos como modelos de aprendizado de máquinas, o termo não é bem definido e geralmente varia dependendo da área de aplicação. Usualmente define-se como aprendizado de máquinas como qualquer técnica computacional que permita a interpretação das informações de forma automatizada. A técnica selecionada para classificação neste trabalho é chamada Rede Neural Artificial ou *Artificial Neural Network* (ANN), mais especificamente uma *Feedforward Neural Network*, também conhecido como um Perceptron Multicamadas (*Multilayer Perceptron*). Redes Neurais Artificiais são modelos computacionais que buscam identificar padrões, ou seja, realizar a classificação de itens. A inspiração para a criação de tais modelos veio do funcionamento de redes neurais biológicas mas seu funcionamento é feito através de operações matemáticas e estatísticas, não existindo nenhuma relação com o processo de aprendizado observado na natureza.

Uma rede neural artificial pode ser definida como uma função da forma $f : \mathbb{R}^E \rightarrow \mathbb{R}^S$, onde E é o tamanho do vetor de entrada, ou seja, a quantidade de variáveis que definem os dados observados, que no caso deste trabalho podem ser informações acerca dos alunos presentes no banco de dados e S é o tamanho do vetor de saída, ou seja, a quantidade de grupos que foram criados na etapa de agrupamento. Como os rótulos de cada grupo são conhecidos, é natural a utilização de uma técnica de aprendizado supervisionado.

A figura abaixo representa o funcionamento de uma rede neural trivial com 12 variáveis de entrada e um neurônio de saída, que pode ser ativado ou não. Pode-se considerar a função logística $\sigma(x) = \left(\frac{1}{1 + \exp\{-x\}} \right)$ como função de ativação e o viés $b = 0$ para obtermos um *pseudo* Modelo de Regressão Logística com resposta binária.

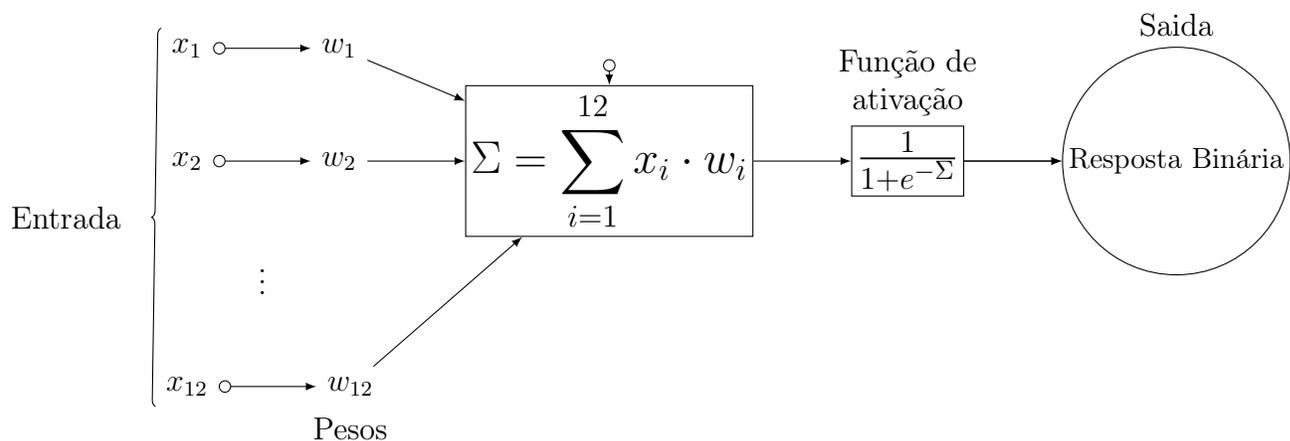
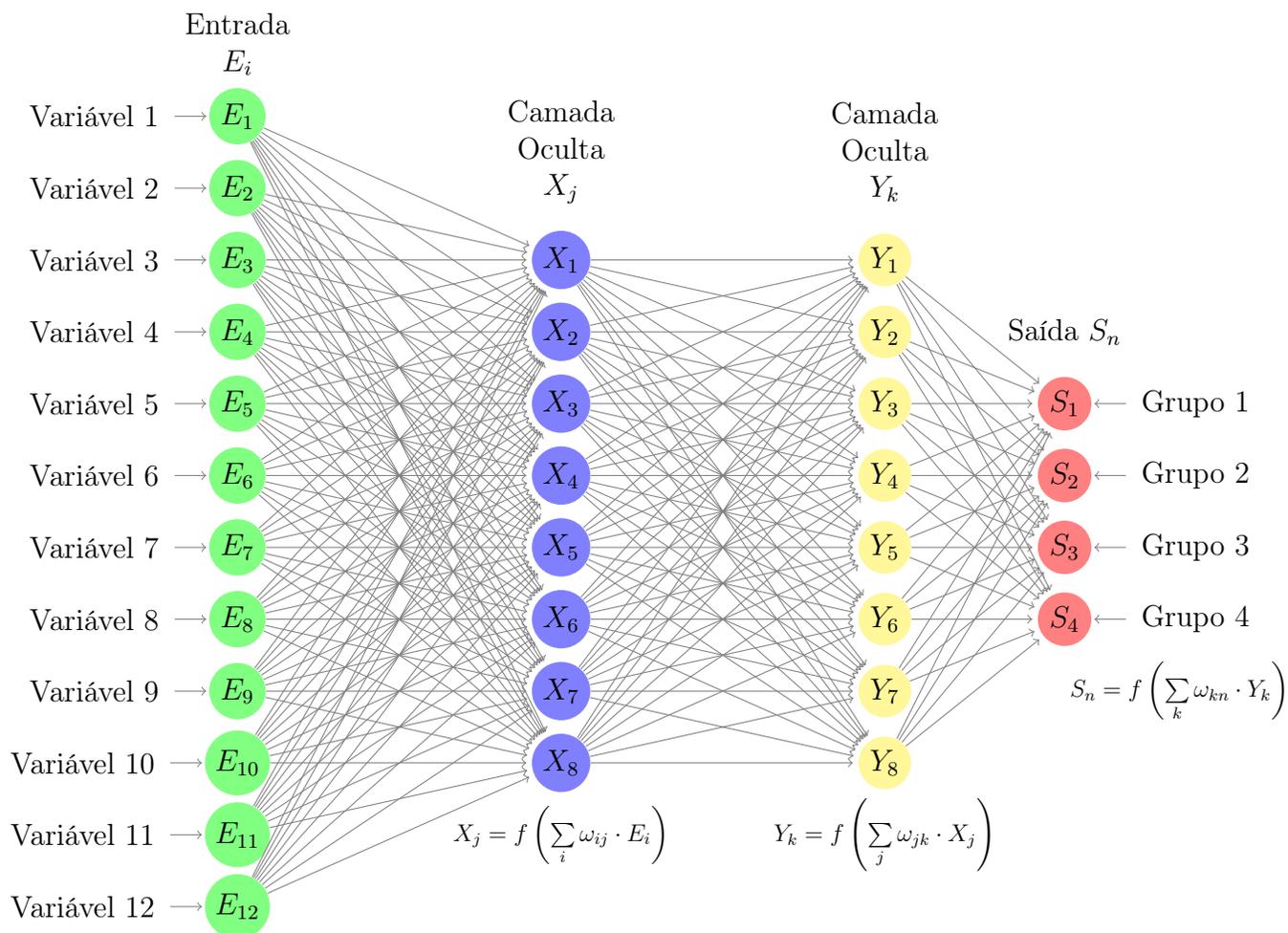


Figura 5 – Funcionamento da conexão entre a camada de entrada com 12 variáveis com um neurônio de saída

Um exemplo de rede neural densamente conectada, caso sejam consideradas 12 variáveis explicativas e quatro grupos definidos na primeira etapa de argumento, pode ser descrito de acordo com a Figura 6 abaixo.



onde ω_{mn} representa o peso atribuído à ligação entres os neurônios m e n

Figura 6 – Exemplo de Rede neural com 12 variáveis de entrada, duas camadas ocultas e uma camada de saída com 4 classes

A construção da Rede Neural se dá primeiramente pela escolha no número de camadas ocultas e suas respectivas quantidades de neurônios, as quais são utilizadas para processar as informações recebidas na entrada e suas dimensões podem ser definidas arbitrariamente. Além disso, para cada camada deve ser definida uma função de ativação apropriada. É importante salientar que quanto mais complexa for a rede maior será a quantidade de recursos computacionais utilizados e o modelo estará mais propenso a apresentar o problema do desaparecimento do gradiente e sobreajuste, já que a maior quantidade de camadas implica em maior quantidade de derivadas para corrigir o erro e quanto mais neurônios existentes na rede neural mais existirá um maior número de combinações possíveis de ativações neurais podendo resultar em uma combinação única para cada observação, resultando em sobreajuste. Também é relevante ser discutido o papel de um eventual fator de viés que é adicionado ao modelo, o viés tem papel análogo ao intercepto em Modelos de Regressão Linear, isto é, pode ser utilizado para efetuar a translação horizontal da função sigmoide afim de melhorar o algoritmo. Após ser definido um modelo apropriado ao objetivo deste trabalho, o problema se resumiu a estimar os pesos ω_i definidos para todas as conexões possíveis afim de minimizar os erros de classificação através do método do gradiente e da retro propagação dos erros de classificação.

6 APLICAÇÃO DE REDES NEURAIS

Neste trabalho foi criado um modelo de Rede Neural Densamente conectada (DNN) para cada semestre, resultando em oito modelos para classificar alunos de cada semestre respectivamente. Posteriormente foram criados modelos para prever a probabilidade de formatura de alunos utilizando o pacote *Keras* (Chollet, F. et al 2015) A função de ativação utilizada na primeira camada da rede neural foi a função sigmoide, que mapeia qualquer valor da reta real no intervalo $(0, 1)$ e é muito utilizada em modelos de classificação. A função da camada de saída foi a função *softmax*, que é uma generalização da função sigmoide e sua utilização é necessária pois o classificador está lidando com diversas possíveis saídas.

A função de ativação utilizada nas demais camadas ocultas foi a Tangente Hiperbólica², ilustradas na Figura 7. A função Tangente Hiperbólica é utilizada por apresentar um derivada, ilustrada na Figura 8, com valor mais alto do que a função sigmoide, implicando em gradientes maiores e o treinamento da rede ocorre mais rapidamente (LeCun, Y., Bottoum L., Orr, G. B., and Muller, 1998). A escolha dessas funções e do número de neurônios foi definida por resultados empíricos e recomendações teóricas disponíveis nos itens de referência deste trabalho.. Não existem critérios puramente estatísticos para comparar as diferenças entre redes neurais e validar suas aplicações. A avaliação da eficácia do modelo se dá pela qualidade da precisão na amostra de validação e treinamento.

Optou-se pela criação de um modelo específico para cada semestre, do primeiro ao oitavo, com objetivo de maximizar a precisão de classificação. Em teoria seria possível criar uma rede geral que funcionasse para todos os semestres letivos, entretanto, seria mais apropriado um modelo feito especificamente para lidar com a correlação entre as menções em cada semestre. A criação deste eventual modelo demandaria maior conhecimento sobre Aprendizado de Máquinas e a possível execução dessa tarefa é um ótimo tópico para ser trabalhada no futuro.

²A Equação da Função Tangente Hiperbólica se relaciona com a função sigmoide $\sigma(x)$ da seguinte forma: $\tanh(x) = 2\sigma(2x) - 1$.

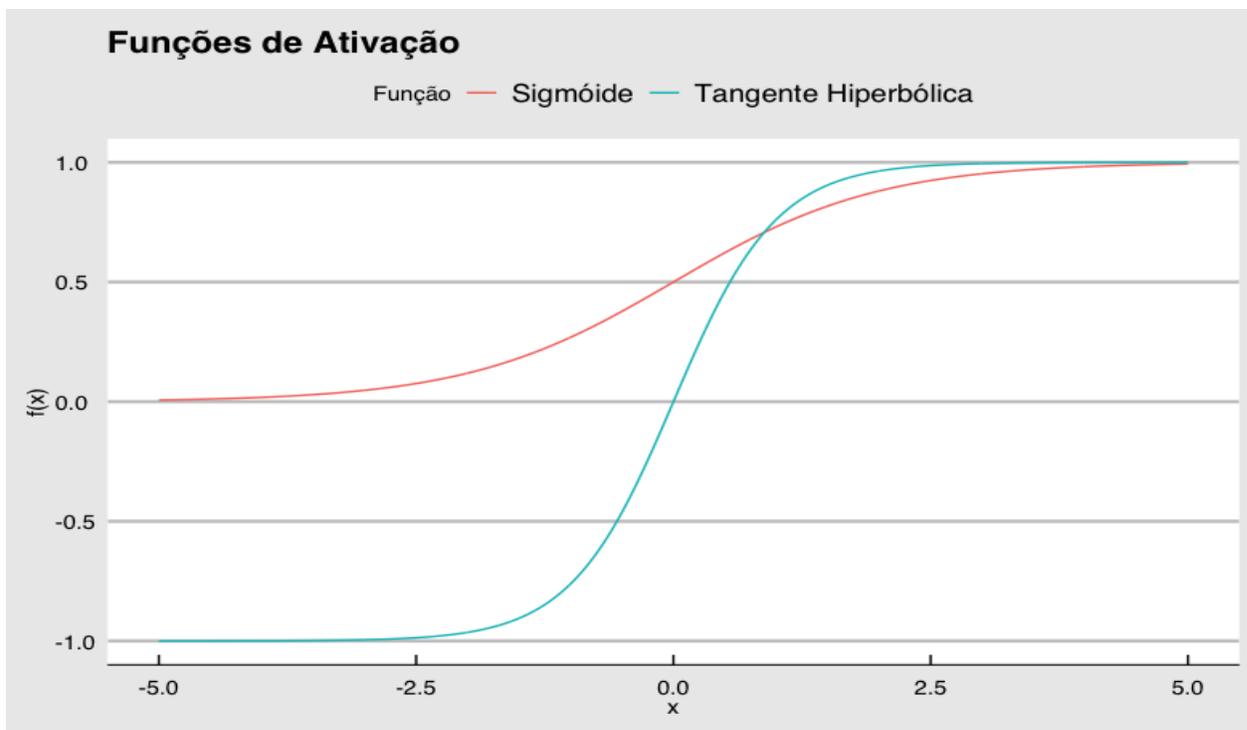


Figura 7 – Diferença entre as funções Sigmóide e Tangente hiperbólica.

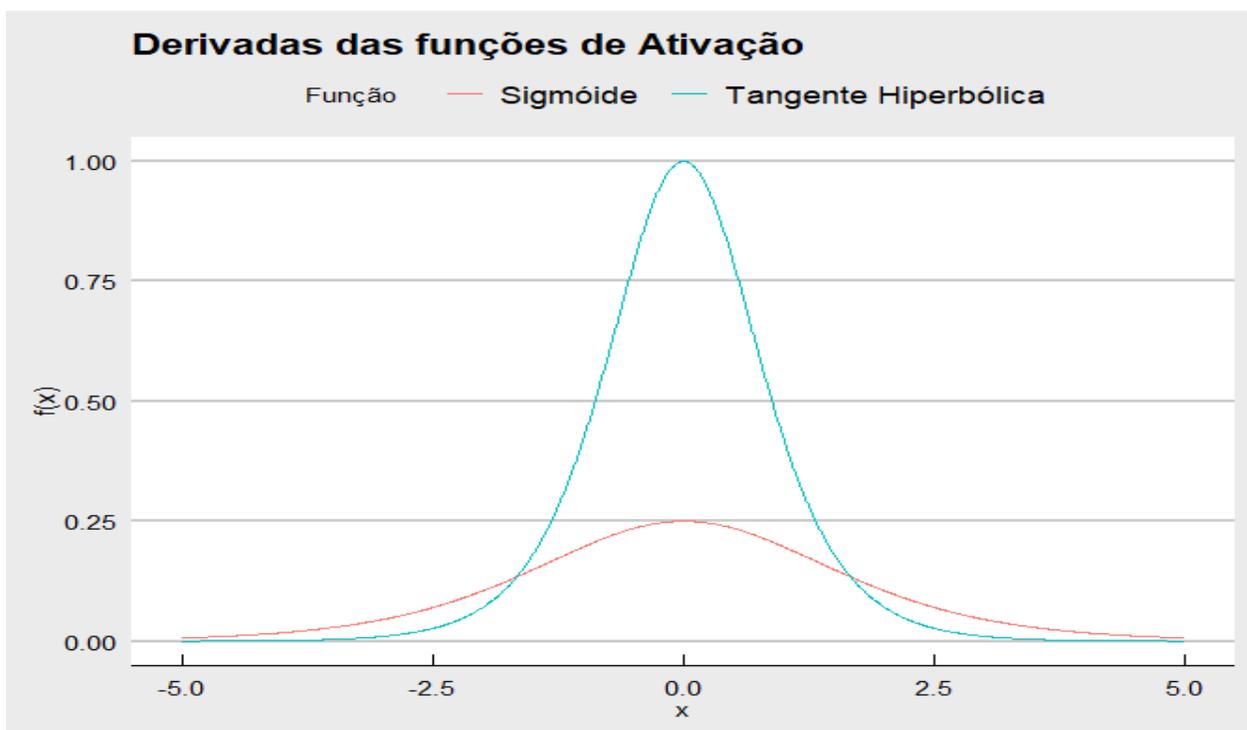


Figura 8 – Diferença entre as derivadas de primeira ordem das funções Sigmóide e Tangente hiperbólica.

Existem alguns parâmetros que podem ser definidos no modelo visando melhorar a performance do algoritmo. Um deles é a taxa de aprendizado, que é um peso aplicado no método do gradiente que tem como objetivo diminuir os 'saltos' do procedimento afim de resolver o problema de desaparecimento do gradiente, que ocorre devido ao fato da correção nos pesos ser feita pelo método do gradiente, que é resultante da derivada parcial da função do erro. Caso esse valor a ser atualizado seja muito pequeno o treinamento da Rede Neural pode parar por completo.

Para solucionar esse problema a o Algoritmo utilizada um otimizador, o caso do presente trabalho foi utilizada o otimizador *RMSprop*, ou *Root Mean Square Propagation*, definido por:

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E(g^2)_{t-1}}} \frac{\partial L}{\partial w} \quad (5)$$

Onde $E(g)$ é a média móvel do gradiente ao quadrado, $\frac{\partial L}{\partial w}$ é a derivada da função de erro em relação ao peso w , η é a taxa de aprendizado e β é o parâmetro de média móvel.

O passo inicial para a configuração do algoritmo é a divisão do banco de dados em duas partes, uma parte correspondente a 80% dos valores do bancos de dados escolhidos de forma aleatória, definida por $\{(\mathbf{x}^n, \mathbf{t}^n : 1 \leq n \leq N)\}$ será selecionada para o treinamento do algoritmo e a fração restante do banco de dados será utilizada para a verificar a adequação do treinamento, após isso são atribuídos pesos iniciais aleatórios ω_{nm} para cada ligações entre o neurônio i e j , $\forall i \neq j$ viabilizando a execução das primeiras iterações do procedimento. Anteriormente foi mencionado que os ajustes nos pesos foram executados de acordo com a retro propagação dos erros através do método do gradiente, em termos matemáticos, o método é descrito da seguinte forma:

$$\Delta\omega = -\eta \cdot \frac{\partial E}{\partial \omega} \quad (6)$$

onde ω é o peso atribuído, L é a função de entropia categórica cruzada definida por:

$$L = - \sum_{c=1}^C y_{\{0,c\}} \ln(p_{\{0,c\}}) \quad (7)$$

onde $p_{\{0,c\}}$ é a probabilidade estimada da classe c e $y_{\{0,c\}}$ é uma variável indicadora que diz se se a classe c atribuída é correta. C é o número de grupos no modelo.

Considerando o fato do erro ser retro propagado no modelo e também analisando que a derivada de primeira ordem da função logística $f(x)$ é $f(x) \cdot (1 - f(x))$, a mudança do peso entre as diversas camadas pode ser ilustrada matematicamente, caso fosse considerada a rede construída na Figura 6:

$$\Delta\omega_{kn} = \eta \cdot \delta_n \cdot Y_k \quad \text{onde} \quad \delta_n = (S_{Correto} - S_n) \cdot S_n \cdot (1 - S_n) \quad (8)$$

$$\Delta\omega_{jk} = \eta \cdot \delta_k \cdot X_j \quad \text{onde} \quad \delta_k = Y_k(1 - Y_k) \sum_{n=1}^8 \omega_{kn} \cdot \delta_n \quad (9)$$

$$\Delta\omega_{ij} = \eta \cdot \delta_j \cdot E_i \quad \text{onde} \quad \delta_j = X_j(1 - X_j) \sum_{k=1}^8 \omega_{jk} \cdot \delta_k \quad (10)$$

Cada Rede Neural criada contém uma camada de entrada composta por um a oito neurônios dependendo do semestre, três camadas ocultas com número variável de neurônios e uma camada de saída com quatro neurônios. Foram utilizados 100 ciclos de treinamento, treinando uma parcela, ou *batch*, de 32 observações do banco de dados por vez, buscando minimizar a entropia cruzada categórica do modelo com um taxa de aprendizado de 0.01. As variáveis categóricas foram convertidas para variáveis numéricas de acordo com os pesos estabelecidos na etapa de agrupamento afim de manter a coerência da análise e facilitar a convergência do classificador. Nos tópicos a seguir, são apresentadas visualizações das Redes Neurais criadas, a cor azul representa a função de ativação *sigmoide*, a cor amarela representa a função de ativação *tangente hiperbólica* e a cor vermelha representa a função de ativação *softmax*. Outro ponto a ser considerado é que o número de grupos gerados para cada curso é variável, entretanto, o número de neurônios na camada de saída é variável de acordo com a quantidade de grupos escolhidos. Nas representações a seguir é considerada a divisão dos discentes em quatro grupos.

6.1 Rede Neural para classificação de alunos no primeiro semestre

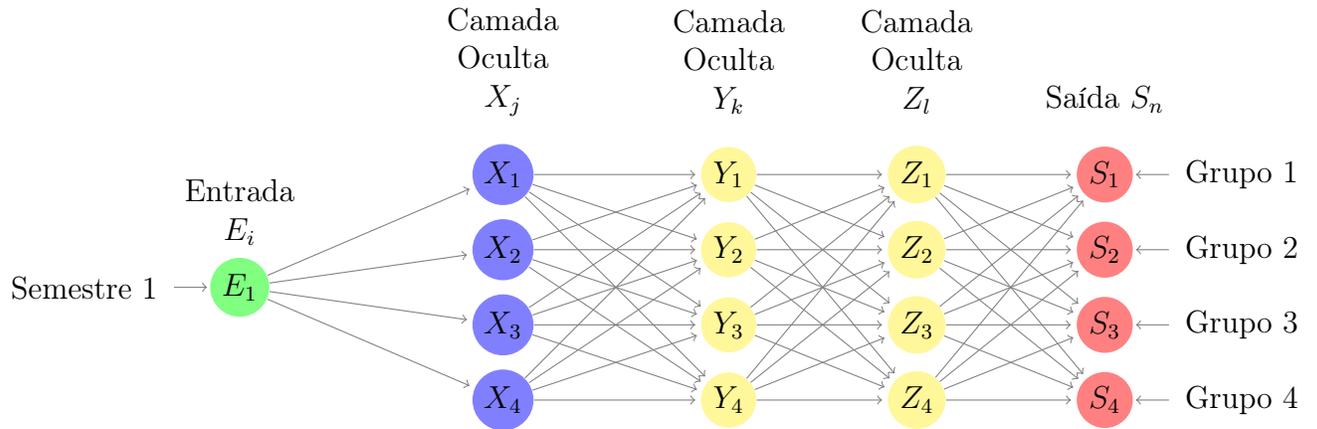


Figura 9 – Rede Neural para classificação de alunos no primeiro semestre.

Neste modelo são estimados 68 parâmetros, 16 desses parâmetros são os respectivos interceptos das cada camadas ocultas e da camada de saída e os outros 52 parâmetros são pesos entre as camadas densamente conectadas.

6.2 Rede Neural para classificação de alunos no segundo semestre

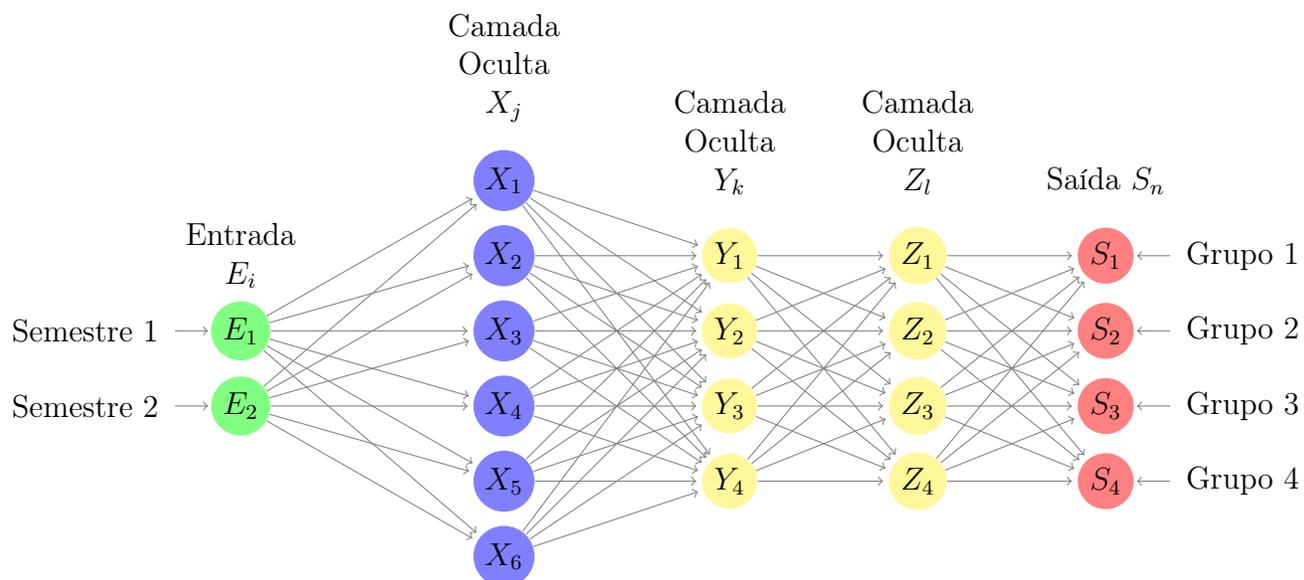


Figura 10 – Rede Neural para classificação de alunos no segundo semestre.

Neste modelo são estimados 80 parâmetros, 18 desses parâmetros são os respectivos interceptos das cada camadas ocultas e da camada de saída e os outros 62 parâmetros são pesos entre as camadas densamente conectadas.

6.3 Rede Neural para classificação de alunos no terceiro semestre

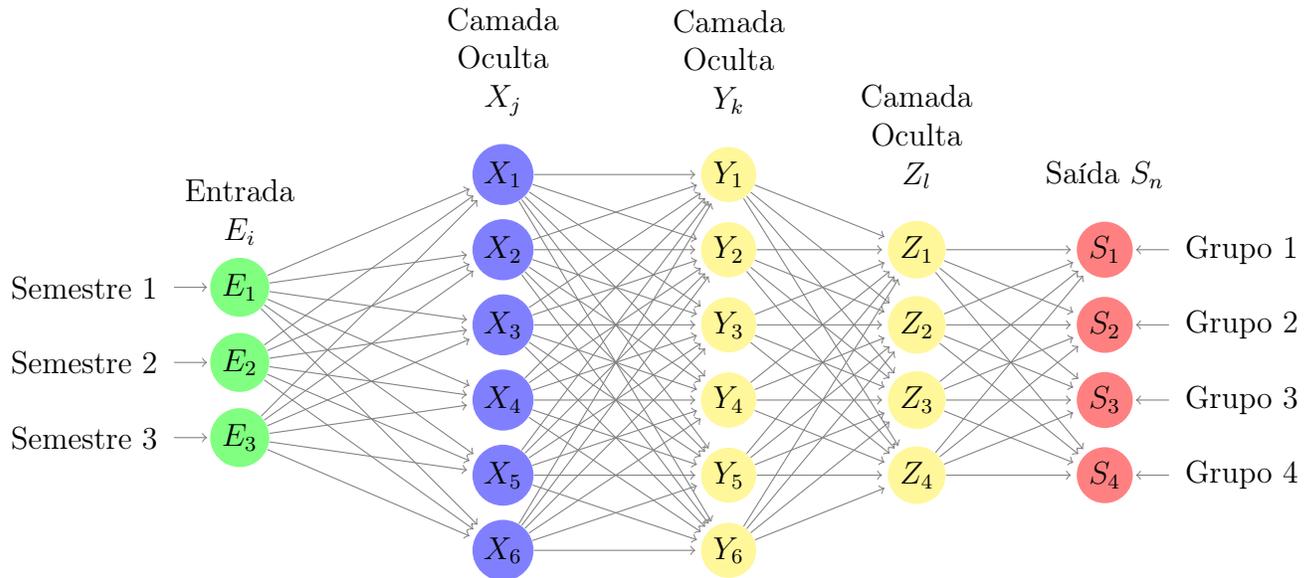


Figura 11 – Rede Neural para classificação de alunos no terceiro semestre.

Neste modelo são estimados 102 parâmetros, 20 desses parâmetros são os respectivos interceptos das cada camadas ocultas e da camada de saída e os outros 82 parâmetros são pesos entre as camadas densamente conectadas.

6.4 Rede Neural para classificação de alunos no quarto semestre

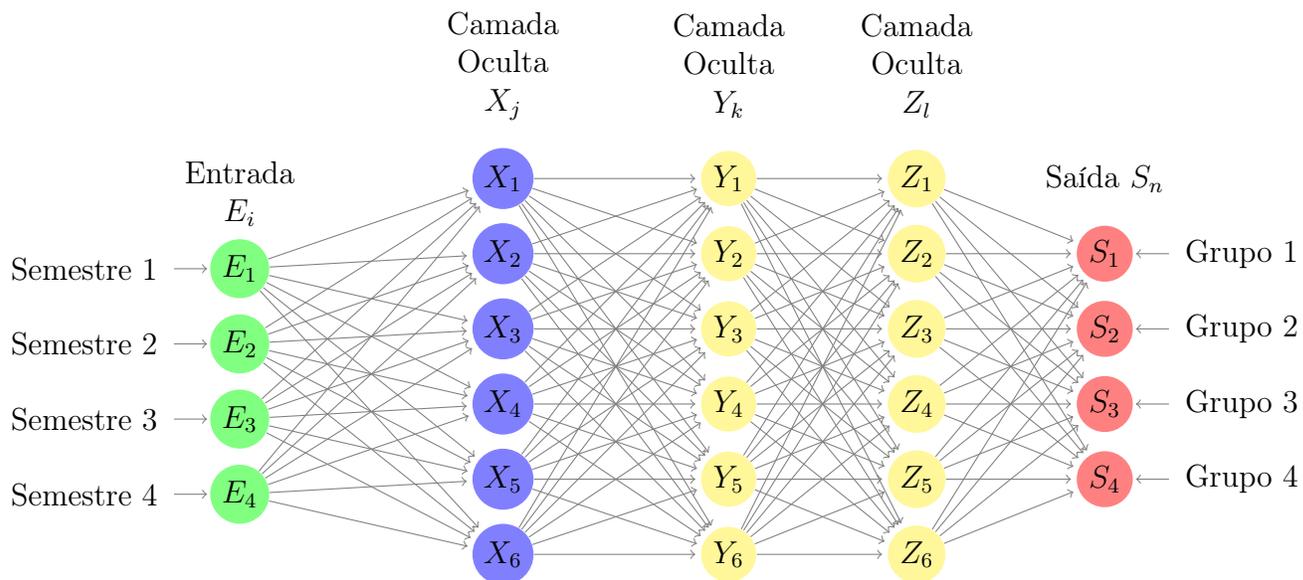


Figura 12 – Rede Neural para classificação de alunos no quarto semestre.

Neste modelo são estimados 124 parâmetros, 22 desses parâmetros são os respectivos interceptos das cada camadas ocultas e da camada de saída e os outros 102 parâmetros são pesos entre as camadas densamente conectadas.

6.5 Rede Neural para classificação de alunos no quinto semestre

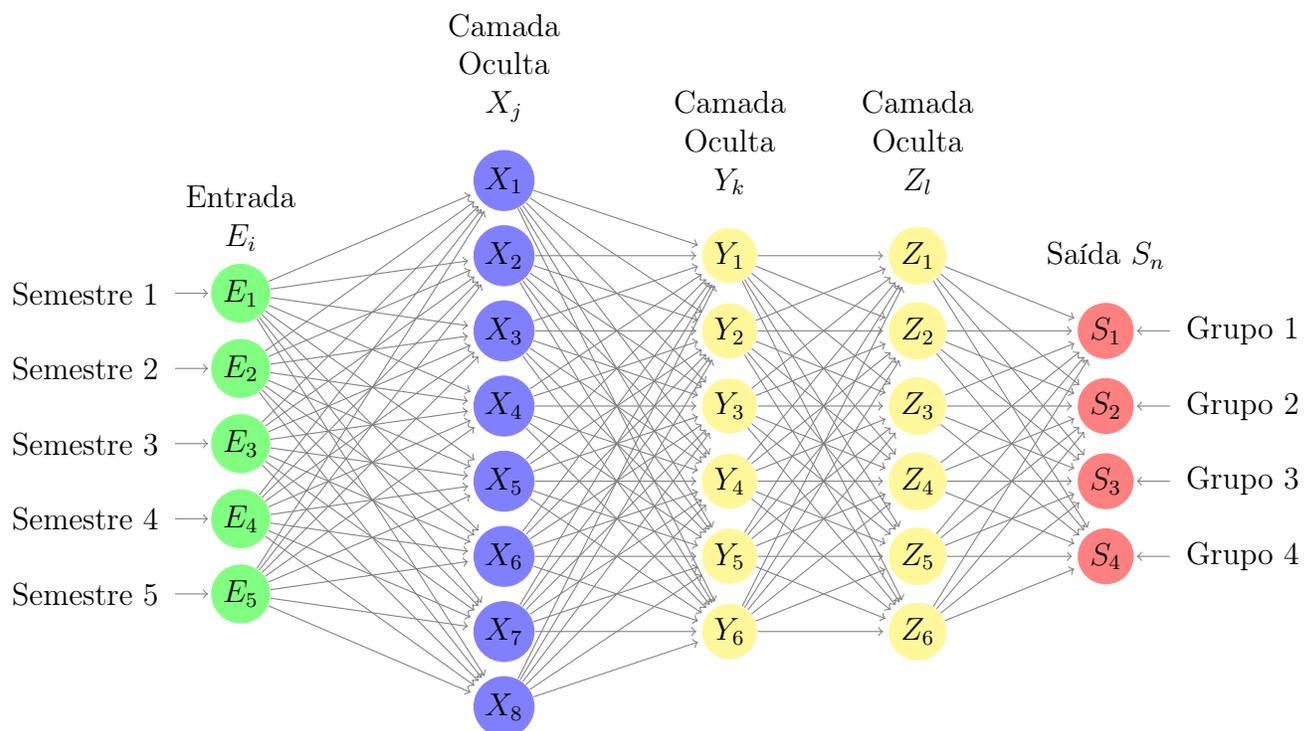


Figura 13 – Rede Neural para classificação de alunos no quinto semestre.

Neste modelo são estimados 140 parâmetros, 24 desses parâmetros são os respectivos interceptos das cada camadas ocultas e da camada de saída e os outros 116 parâmetros são pesos entre as camadas densamente conectadas.

6.6 Rede Neural para classificação de alunos no sexto semestre

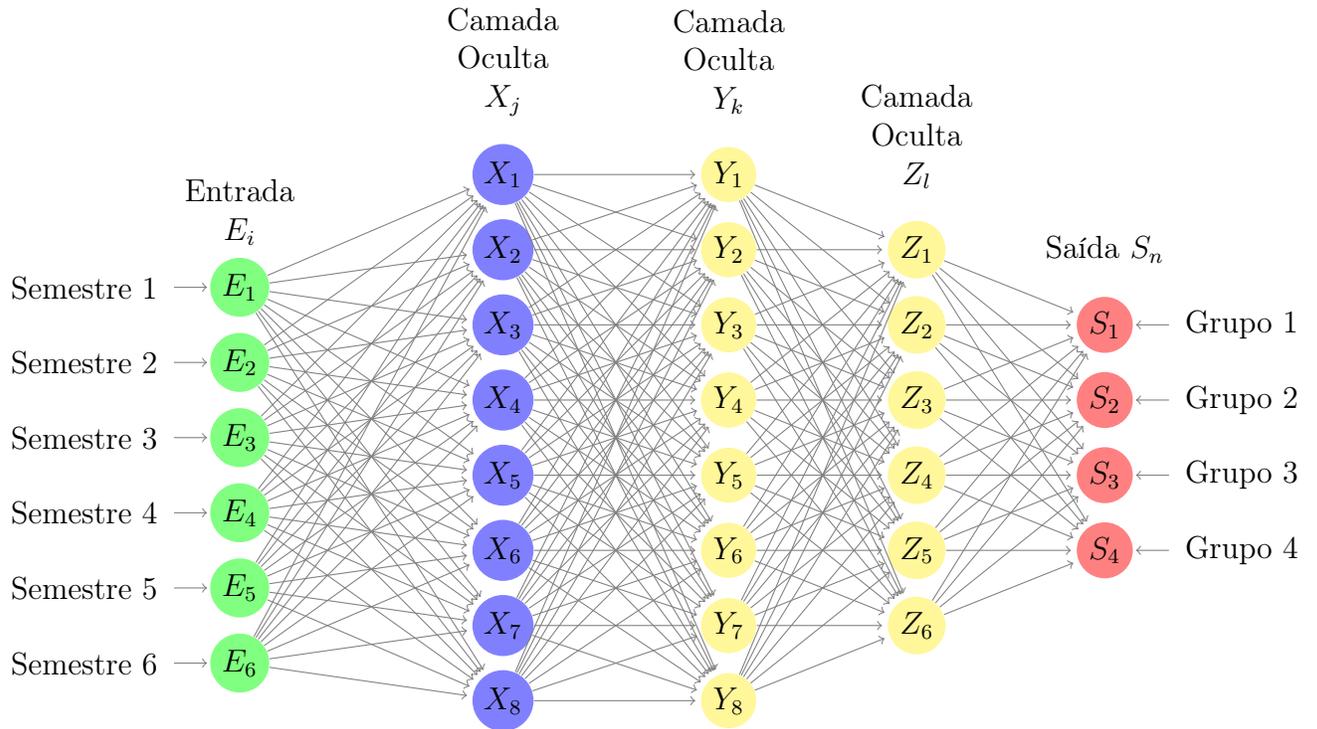


Figura 14 – Rede Neural para classificação de alunos no sexto semestre.

Neste modelo são estimados 170 parâmetros, 26 desses parâmetros são os respectivos interceptos das cada camadas ocultas e da camada de saída e os outros 144 parâmetros são pesos entre as camadas densamente conectadas.

6.7 Rede Neural para classificação de alunos no sétimo semestre

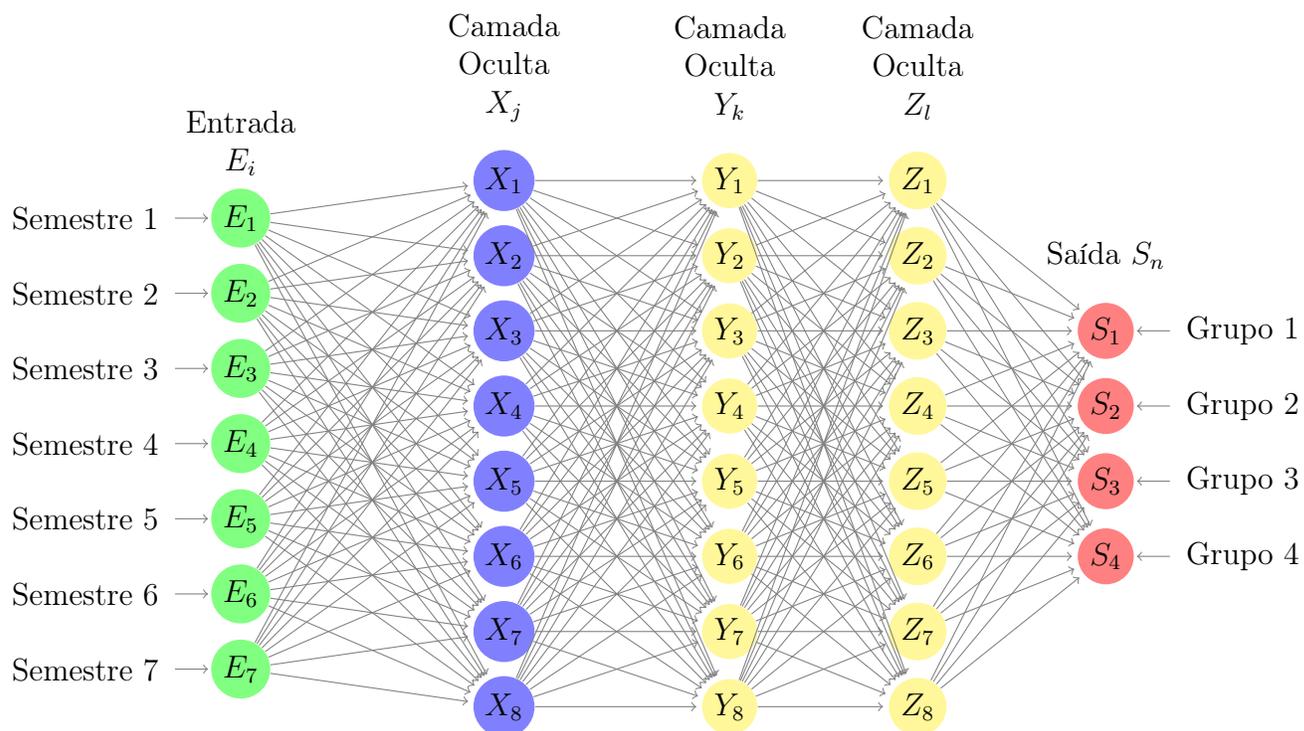


Figura 15 – Rede Neural para classificação de alunos no sétimo semestre.

Neste modelo são estimados 196 parâmetros, 28 desses parâmetros são os respectivos interceptos das cada camadas ocultas e da camada de saída e os outros 168 parâmetros são pesos entre as camadas densamente conectadas.

6.8 Rede Neural para classificação de alunos no oitavo semestre

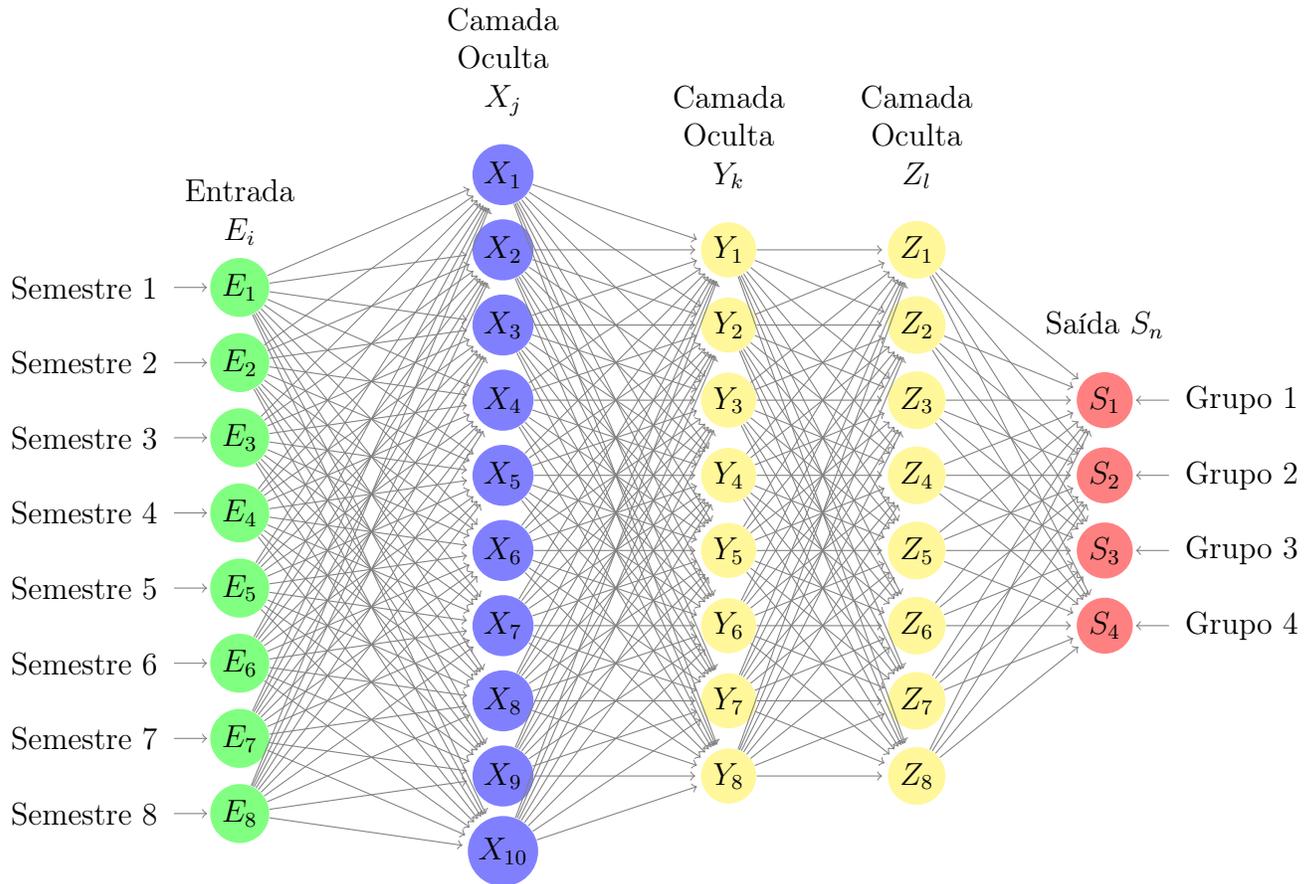


Figura 16 – Rede Neural para classificação de alunos no oitavo semestre.

Neste modelo são estimados 216 parâmetros, 30 desses parâmetros são os respectivos interceptos das cada camadas ocultas e da camada de saída e os outros 186 parâmetros são pesos entre as camadas densamente conectadas.

6.9 Considerações

É importante ressaltar que o número crescente de neurônios foi necessário, pois o número de combinações possíveis de menções cresce a cada novo semestre considerado e o modelo necessitou de uma maior quantidade desses elementos para aumentar o número de combinações possíveis de ativações dos neurônios e assim ter maior precisão. A partir da classificação do modelo, foi possível obter a probabilidade de cada indivíduo pertencer a um dos grupos através da função `predict_proba()` do pacote *Keras*.

7 APLICATIVO SHINY

Como já foi mencionado brevemente antes, uma forma inteligente e elegante de concatenar todos os procedimentos já descritos neste relatório e possibilitar a utilização é a elaboração de uma ferramenta interativa que pode ser facilmente disponibilizada ao público interessado. Tendo isso em vista, foi decidido criar um aplicativo iterativo utilizando o pacote Shiny do software R, pois sua plataforma base (*R*) também é o software utilizado para realizar todos os procedimentos descritos acima, possibilitando a plena utilização de todas as ferramentas desenvolvidas.

O aplicativo *Shiny* consiste de duas partes, uma delas é o servidor ou *server* onde são executados todos as funções, o agrupamento e a classificação. Além disso, também são registrados os elementos que possibilitam as operações iterativas do aplicativo. A outra parte é a interface de usuário ou *User Interface* (UI). Na UI é programado todo o *layout* que será utilizado pelo usuário do aplicativo. Cada aspecto da UI tem que ser necessariamente programado, o que permite a construção de elementos customizados de infinitas formas, mas, requer um conhecimento considerável da linguagem *R*, das funcionalidades do pacote *Shiny* e até mesmo alguns aspectos da linguagem *html*.

As funcionalidades do aplicativo foram distribuídas em seis janelas.

7.1 Primeira janela

A primeira janela apresenta uma breve descrição do aplicativo, de seu criador e orientador.

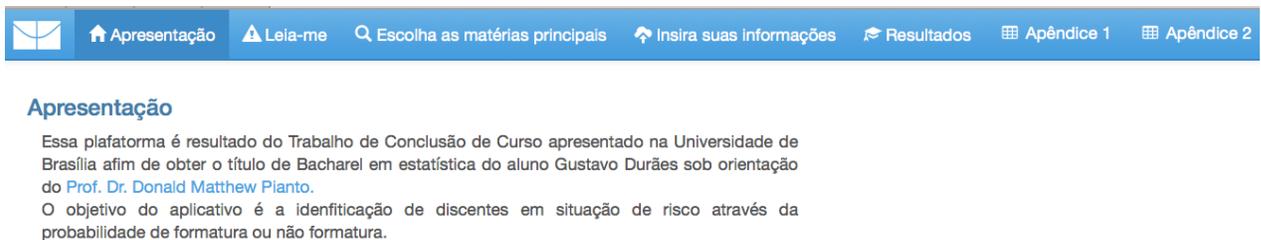


Figura 17 – Layout da primeira janela do aplicativo.

7.2 Segunda janela

A segunda janela contém instruções de uso detalhadas para a utilização da plataforma.

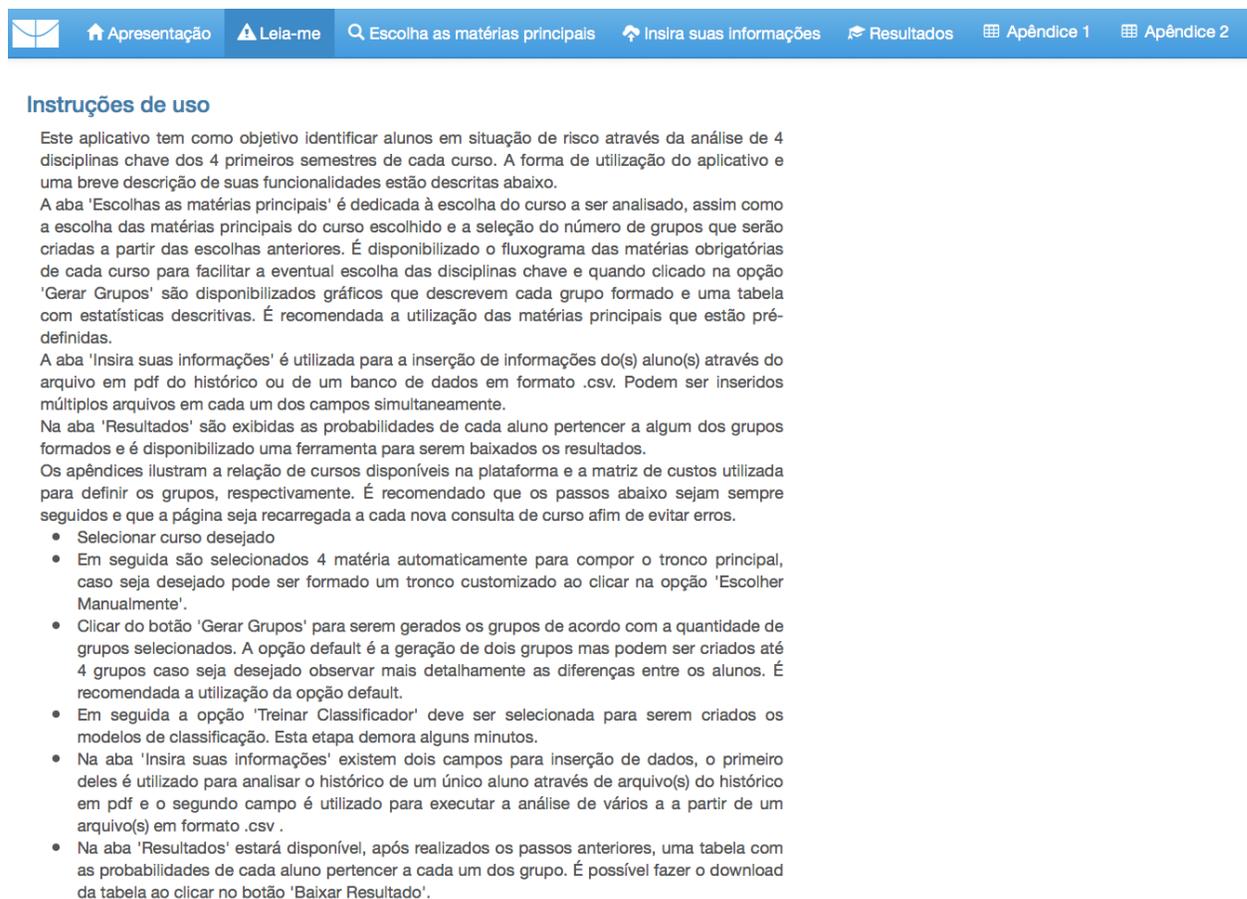


Figura 18 – Layout da segunda janela do aplicativo

7.3 Terceira janela

A terceira janela é dedicada a escolha das matérias principais, criação dos grupos, de acordo com a seção e a criação dos modelos de classificação, de acordo com o capítulo cinco deste trabalho. Como ferramentas de auxilio, são exibidos o fluxo do curso selecionado, um gráfico de barras descrevendo cada grupo, estatísticas resumo de cada grupo, indicando o semestre médio até o estado final (formatura ou não formatura) e proporção de menções em cada grupo. Além disso, após executado o treinamento das redes neurais, são exibidas tabelas que ilustram a classificação feita pelo modelo e a classe real, denominadas matrizes de

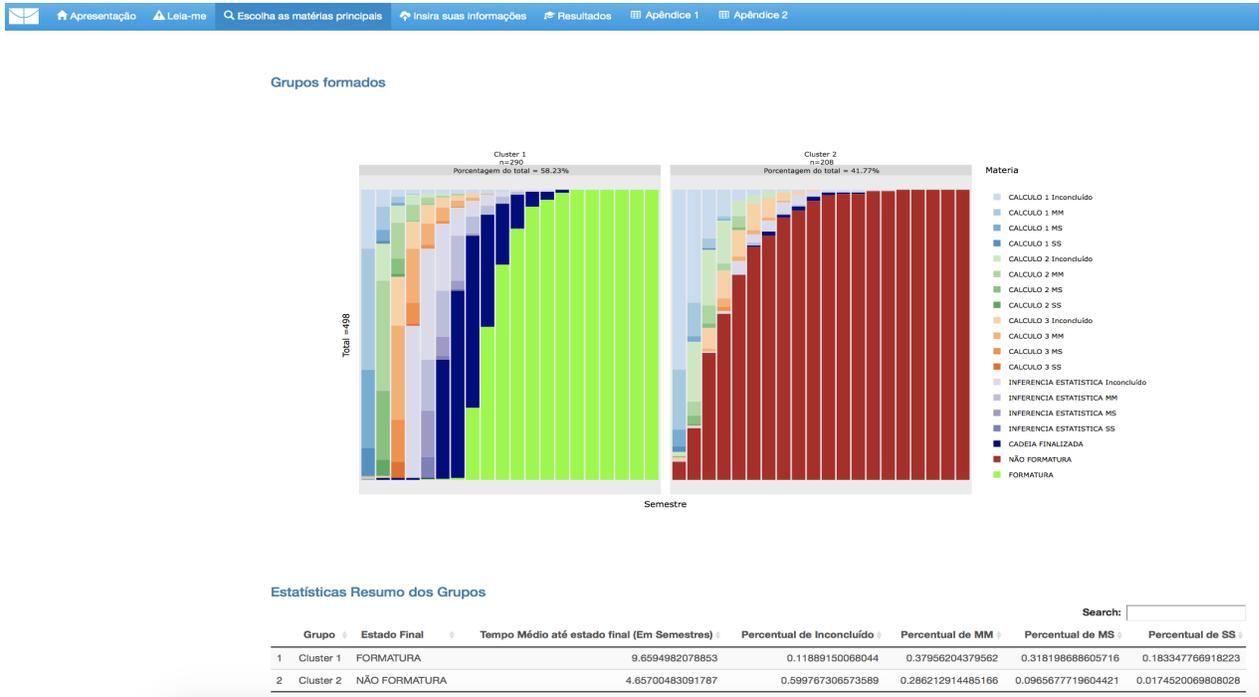


Figura 20 – Exemplo de Layout da terceira janela do aplicativo - Grupos

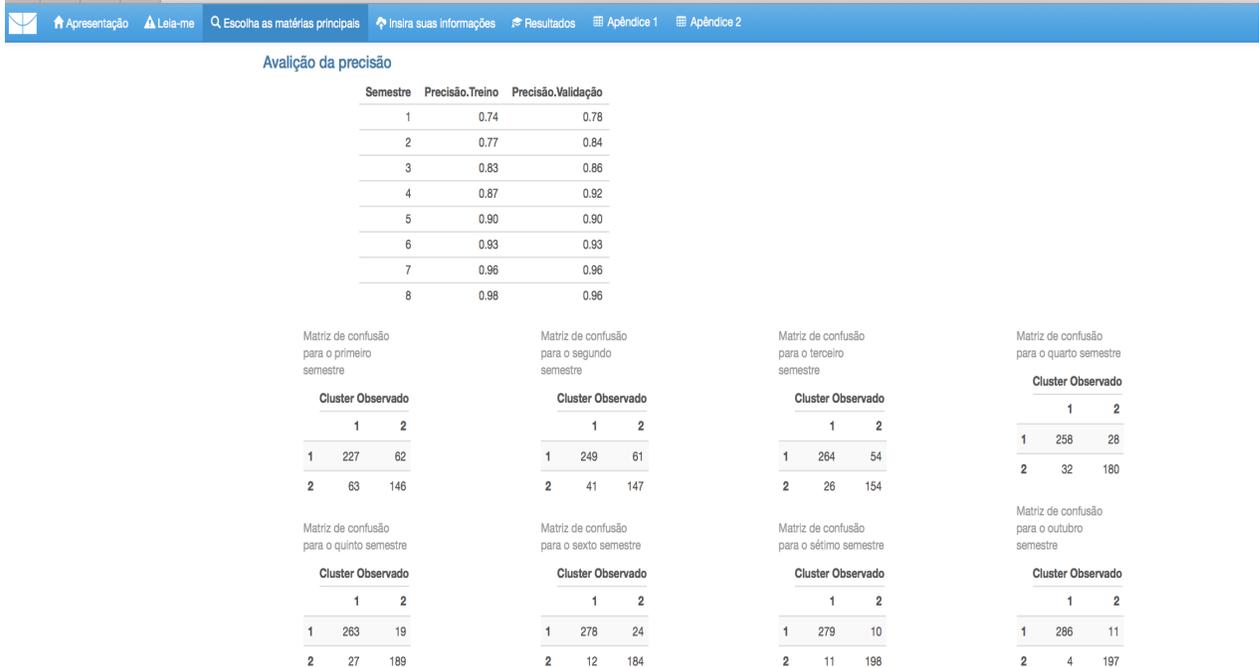


Figura 21 – Exemplo de Layout da terceira janela do aplicativo - Avaliação da precisão

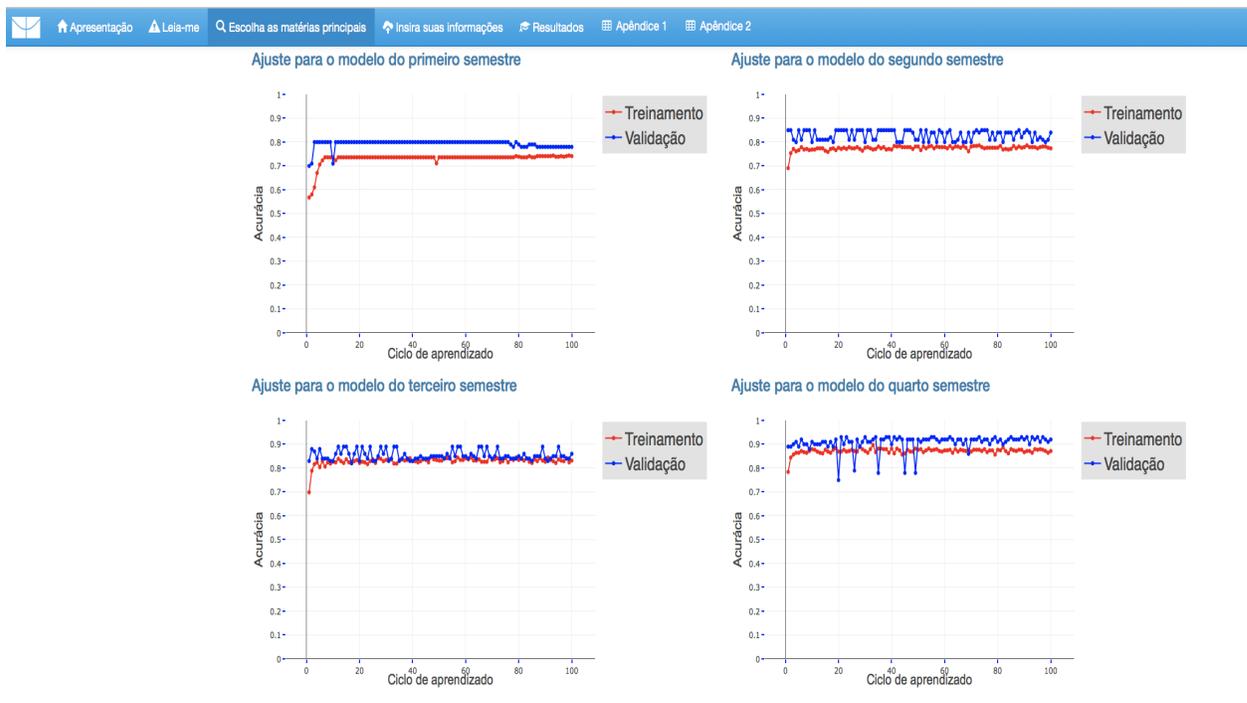


Figura 22 – Exemplo de Layout da terceira janela do aplicativo - Ajuste dos modelos

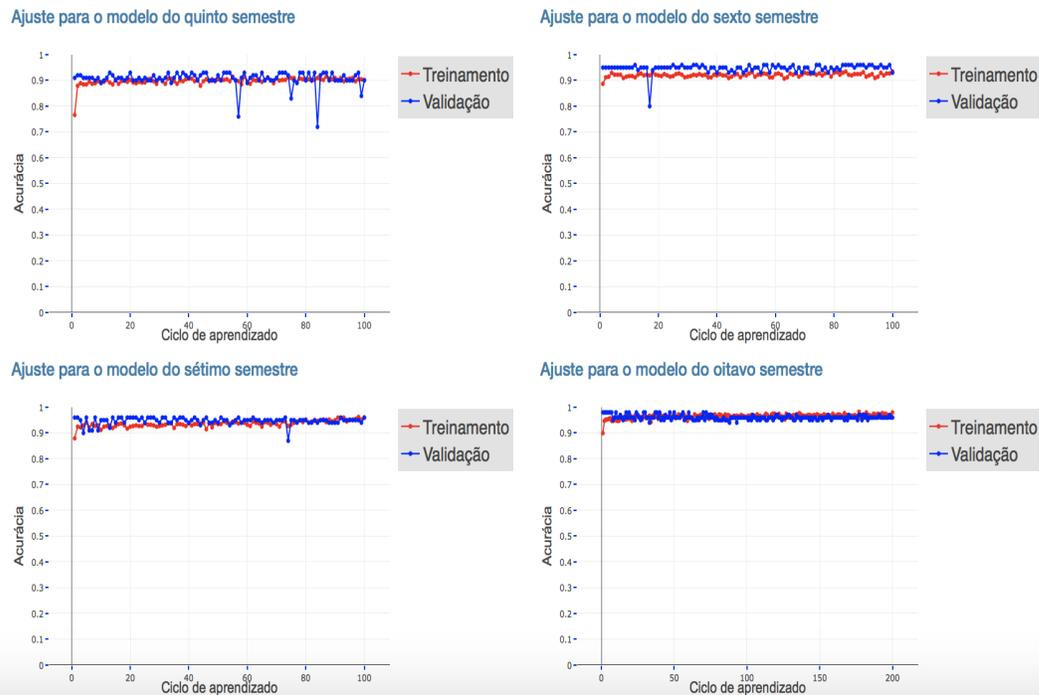


Figura 23 – Exemplo de Layout da terceira janela do aplicativo - Ajuste dos modelos

7.4 Quarta janela

A quarta janela é utilizada para a inserção das informações do histórico dos discentes. É possível inserir dados através do histórico escolar padrão fornecido pela Universidade de Brasília em formato *pdf* ou através de um banco de dados em formato *csv*. Caso seja optado inserir o banco de dados em formato *csv* o arquivo deve conter quatro colunas. A primeira coluna deverá conter a matrícula dos alunos e ser nomeada *Matrícula*. A segunda coluna deverá conter o nome das disciplinas que o aluno já cursou e ser nomeada *Disciplina*. A terceira coluna deverá conter a menção que o aluno recebeu em cada disciplina e ser nomeada *Menção*. A quarta coluna deverá conter um valor numérico que indica em qual semestre o aluno cursou a disciplina e ser nomeada *Período*, é importante que essa variável seja representada por valores inteiros. Os dados inseridos são transformados para o formato sequencial.

A imagem a seguir ilustra a inserção de um conjunto de dados em formato *csv*.

The screenshot displays the application's data entry interface. On the left, there are two sections: 'Histórico' with a 'Selecionar' button and 'Arquivo em formato em .pdf' text, and 'Banco de dados' with a 'Selecionar' button and 'exemplo.csv' text. Below these is an 'Upload complete' button and instructions about the CSV format. The main area is divided into two tables:

Dados Inseridos

Matrícula	Disciplina	Menção	Período
1 XXXX	CALCULO 1	MM	1
2 XXXX	ESTATISTICA EXPLORATORIA	MM	1
3 XXXX	INTRODUCAO A PROBABILIDADE	MM	1
4 XXXX	CALCULO 2	MM	2
5 XXXX	CALCULO 3	MM	3
6 XXXX	INFERENCIA ESTATISTICA	MM	5
7 YYYY	CALCULO 1	MM	1

Dados Transformados

Matrícula	Semestre 1	Semestre 2	Semestre 3	Semestre 4	Semestre 5	Semestre 6	Semestre 7	Semestre 8	Semestre 9
1 XXXX	CALCULO 1 MM	CALCULO 2 MM	CALCULO 3 MM	INFERENCIA ESTATISTICA Inconcluido	INFERENCIA ESTATISTICA MM				
2 YYYY	CALCULO 1 MM								

Figura 24 – Exemplo de Layout da quinta janela do aplicativo

7.5 Quinta janela

A quinta janela do aplicativo é utilizada para exibir os resultados da classificação dos discentes cujo os dados foram inseridos na quarta janela. Os são exibidas as probabilidade de cada um dos alunos pertencerem aos grupos gerados na terceira janela.

	Matrícula	Probabilidade de FORMATURA (Cluster 1)	Probabilidade de NÃO FORMATURA (Cluster 2)
1	XXXX	0.992303907871246	0.00769601808860898
2	YYYY	0.114596828818321	0.885403156280518

Figura 25 – Exemplo de Layout da quinta janela do aplicativo

7.6 Sexta e sétima janelas

As duas janelas finais do aplicativo contém os apêndices que exibem informações sobre os cursos disponíveis no aplicativo e a matriz de custos utilizada para quantificar as diferenças entre as combinações de menções e disciplinas.

7.7 Considerações

O aplicativo foi construído com o objetivo de ser o mais abrangente possível, visando possibilitar a análise de todos os cursos da Universidade de Brasília, entretanto, podem haver erros dada a complexidade da aplicação. Como todos os testes na etapa de desenvolvimento foram feitos observando a seleção dos cursos de Matemática e Estatística a aplicação funciona muito bem para estes cursos, entretanto, caso fosse adotado o uso intensivo da aplicação por outros cursos seria necessário realizar pequenas correções que inevitavelmente serão necessárias.

8 CONSIDERAÇÕES FINAIS

Os resultados alcançados neste trabalho foram a par das expectativas, a construção da plataforma e todo o arcabouço que a sustenta proporcionam informações que podem ser de grande ajuda para a solução do problema proposto. Dada a complexidade da aplicação é necessário constante acompanhamento e correção de eventuais erros na solução proposta.

8.1 Trabalhos futuros

Existem várias aplicações muito interessantes que podem ser feita a partir deste trabalho. Uma atualização do banco de dados contendo o histórico dos discentes adicionado do cruzamento com dados socioeconômicos, poderia proporcionar identificação de alunos em situação de risco antes mesmo de ser cursada algum matéria na universidade, maximizando a eficiência de qualquer medida que poderia ser tomada afim de diminuir a evasão Universitária. Outra aplicação interessante é trabalhar em conjunto com departamentos que demonstrem interesse na utilização da plataforma, afim de melhor definir os troncos principais de acordo com as características únicas de cada curso.

Além disso, é possível a utilização de outras modelagens com resposta multinomial afim de melhorar o algoritmo classificação, tendo em vista que a aplicação de aprendizado de máquinas é razoavelmente demorada e é difícil interpretar como o modelo lida com as informações.

9 APÊNDICE

Todos os códigos utilizados neste trabalho estão disponíveis no endereço: <https://github.com/gustavoduraes/TCC>. Os bancos de dados que contém informações pessoais dos alunos não está disponível por motivo de segurança e isso impede que o código seja reproduzido.

O aplicativo Shiny está disponível no endereço https://gustavoduraes.shinyapps.io/Shiny_TCC/ e a matriz de custos e relação de cursos utilizadas encontram-se nas duas últimas janelas do aplicativo.

REFERÊNCIAS

- Agresti, A. (2015). *Foundations of Linear and Generalized Linear Models*. Wiley Series in Probability and Statistics. Wiley.
- Chollet, F. et al. (2015). Keras. <https://keras.io>.
- DAHAL, P. Classification and loss evaluation - softmax and cross entropy loss. Disponível em <https://deepnotes.io/softmax-crossentropy1>.
- Dancho, M. (2018). Tensorflow for r: Deep learning with keras to predict customer churn.
- Gabadinho, A., Ritschard, G., Müller, N. S., and Studer, M. (2011). Analyzing and visualizing state sequences in R with TraMineR. *Journal of Statistical Software*, 40(4):1–37.
- LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. (1998). Efficient backprop. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 9–50, London, UK, UK. Springer-Verlag.
- Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective*. Adaptive computation and machine learning. MIT Press.
- Renals, S. Machine learning practical lectures. Notas de aula disponíveis em <http://www.inf.ed.ac.uk/teaching/courses/mlp/lectures-2018.html>.
- RStudio, I. (2013). *Easy web applications in R*.
- Vehbi Olgac A Karlik, B. (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence And Expert Systems (IJAE)*.
- Wikipedia contributors (2018). Graph (discrete mathematics) — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Graph_\(discrete_mathematics\)&oldid=853815909](https://en.wikipedia.org/w/index.php?title=Graph_(discrete_mathematics)&oldid=853815909).