TRABALHO DE CONCLUSÃO DE CURSO

AVALIAÇÃO DO IMPACTO DOS DADOS DA PANDEMIA EM MODELOS DE MACHINE LEARNING NO MERCADO DE RENDA VARIÁVEL

ANDRÉ HENRIQUE REIS GOMES

Brasília, 20 de dezembro de 2023

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

AVALIAÇÃO DO IMPACTO DOS DADOS DA PANDEMIA EM MODELOS DE MACHINE LEARNING NO MERCADO DE RENDA VARIÁVEL

ANDRÉ HENRIQUE REIS GOMES

Orientador: PROF. DR. EDSON MINTSU HUNG , ENE/UNB

TRABALHO DE CONCLUSÃO DE CURSO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO - XXX/AAAA
BRASÍLIA, 20 DE DEZEMBRO DE 2023.

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

AVALIAÇÃO DO IMPACTO DOS DADOS DA PANDEMIA EM MODELOS DE MACHINE LEARNING NO MERCADO DE RENDA VARIÁVEL

ANDRÉ HENRIQUE REIS GOMES

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRA ELÉTRICA.

APROVADA POR:

Prof. Dr. Edson Mintsu Hung , ENE/UnB
Orientador

Prof. Dr. Geovany Araújo Borges , ENE/UnB
Examinador interno

Prof. Dr. Renam Castro da Silva, D.Sc UFRJ
Examinador externo

BRASÍLIA, 20 DE DEZEMBRO DE 2023.

FICHA CATALOGRÁFICA
ANDRÉ HENRIQUE REIS GOMES
AVALIAÇÃO DO IMPACTO DOS DADOS DA PANDEMIA EM MODELOS DE MACHINE LEARNING NO MERCADO DE RENDA VARIÁVEL
2023xv, 73p., 201x297 mm
(ENE/FT/UnB, Engenheira Elétrica, Engenharia Elétrica, 2023)
Trabalho de Conclusão de Curso - Universidade de Brasília
Faculdade de Tecnologia - Departamento de Engenharia Elétrica

REFERÊNCIA BIBLIOGRÁFICA

ANDRÉ HENRIQUE REIS GOMES (2023) AVALIAÇÃO DO IMPACTO DOS DADOS DA PANDEMIA EM MODELOS DE MACHINE LEARNING NO MERCADO DE RENDA VARIÁVEL. Trabalho de Conclusão de Curso em Engenharia Elétrica, Publicação xxx/AAAA, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 73p.

CESSÃO DE DIREITOS

AUTOR: ANDRÉ HENRIQUE REIS GOMES
TÍTULO: AVALIAÇÃO DO IMPACTO DOS DADOS DA PANDEMIA EM MODELOS DE MACHINE LEARNING NO MERCADO DE RENDA VARIÁVEL.
GRAU: Engenheira Elétrica ANO: 2023

_____

ANDRÉ HENRIQUE REIS GOMES
andrehrgomes@gmail.com

Agradecimentos

Gostaria de, primeiramente, agradecer a minha família que muito me apoiou durante a graduação, sempre com muito carinho e amor.

Um destaque especial aos meus pais que são a inspiração máxima da minha vida.

A minha namorada pelo companheirismo em todo esse período de graduação e amor nos anos que passaram e os que virão.

Aos meus amigos que sempre foram uma rede de apoio, trazendo felicidade e contribuindo com momentos memoráveis, marcando essa época como uma das melhores da minha vida.

Por último, gostaria de agradecer ao meu avô Elmano pelo incondicional apoio e orgulho que ele tinha de mim. Sua falta sempre será sentida, assim como o amor que o senhor tinha por nós.

Resumo

O presente trabalho se dispôs a discorrer sobre os impactos que os dados da pandemia teriam em modelos de *machine learning*, no contexto do mercado de renda variável. Foram avaliados, principalmente, modelos de *Boosting*, sendo eles o *LightGBM*, o *CatBoost* e o *XGBoost*. Além disso, foi simulado um modelo de ensemble com o próprio *LightGBM*, um *Double Ensemble*. A base para a modelagem foi uma biblioteca desenvolvida pela *Microsoft* chamada Qlib, que serve como ferramenta para a extração de dados, modelagem e avaliação do desempenho dos modelos.

Para a modelagem dos 3 algoritmos citados, foram realizados muitos testes e muitas mudanças de configurações, desde o processamento dos dados até a otimização de hiperparâmetros. Essas alterações permitiram melhorar o desempenho dos modelos, demonstrando que esses tipos de modelos que utilizam *boosting*, desempenham muito bem nessa situação proposta.

Por fim, foi visto o impacto dos dados da pandemia, quando colocados no dataset de teste e quando colocados no dataset de treino e validação, mostrando que a depender da característica do bloco de dados utilizados o impacto pode ser mais severo em um ou em outro, no caso, foram utilizados papéis que compõe o Ibovespa e papéis que compõe a bolsa brasileira (B3) como um todo.

Palavras-chave: Aprendizado de Máquina, Árvores de Decisão , Investimentos Quantitativos, Testes de Dataset, Qlib, Mercado Financeiro.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

Capítulo 1

Introdução

1.1 Motivação

No presente ano de 2023, tem-se notado um altíssimo aumento do interesse da população brasileira e mundial em inteligência artificial e em investimentos de modo geral. De acordo com pesquisa realizada pela *McKinsey & Company*, um terço dos entrevistados afirma que suas companhias já utilizam inteligência artificial em algum processo recorrente e 40% dos entrevistados afirmaram que suas empresas aumentaram o investimentos em inteligência artificial no último ano. Além disso, teve-se a explosão de IA generativas no público comum, não se limitando a círculos e tecnologia. O maior expoente dessa popularização foi o ChatGPT, que teve quando lançado o recorde de usuários cadastrados, com 100 milhões em 2 meses de utilização.

Em relação a investimentos, nota-se um crescimento constante do número de investidores ano a ano, com crescimento de 34% na modalidade de renda fixa e 23% em renda variável, de acordo com dados da própria B3, de junho/22 a junho/2023. O número total de investidores chegou em junho/2023 a 17,6 milhões de pessoas, com um potencial de crescer muito mais.

1.2 Objetivos

Aliando esses dois temas atuais e com a sinergia que eles apresentam, pensou-se no presente trabalho de analisar o desempenho de diferentes modelos de inteligência artificial, quando impactados por variações extremas do mercado, como a pandemia. Com isso, analisou-se como os modelos se comportam quando as alterações estão no dataset de teste e como eles se comportam quando as alterações estão nos datasets de treino e validação.

1.3 Limitações dos Experimentos

Neste trabalho foram realizados alguns experimentos que simulam desempenhos de portfólios montados pelos modelos de machine learning, porém essas simulações contém várias limitações. Os testes realizados só levam em consideração características inatas dos papéis, como preço e volume de negociação, excluindo da análise indicadores econômicos, como taxa de juros básica e PIB, além de eventos do mercado que são impossíveis prever. Tais eventos podem ser qualquer grande notícia ou acontecimento que gere medo nos investidores e acrescente risco as operações, como início de guerras, que podem ter impacto na economia global, como o conflito Rússia-Ucrânia, e acusações e julgamentos de corrupção, que foram constantes nos últimos 15 anos no Brasil. Esses eventos são componentes externos que não entraram na modelagem para as simulações da situação proposta.

1.4 Estrutura do Trabalho

O trabalho contém 5 capítulos, sendo estruturado da seguinte forma.

- **Capítulo 2:** foram abordados alguns pontos de fundamentação teórica, que auxiliarão a compreensão dos experimentos realizados e na interpretação dos resultados.
- **Capítulo 3:** foram descritos os procedimentos dos experimentos, um pouco do passo a passo do que foi feito para execução e melhoria dos modelos.
- **Capítulo 4:** foram apresentados e analisados os resultados dos modelos, sendo discutidos os impactos que os dados da pandemia tem nos diferentes datasets. Também foi descrito um pequeno tutorial, com dicas que podem auxiliar trabalhos futuros e algumas recomendações de caminhos para futuras pesquisas.
- **Capítulo 5:** serviu como a conclusão do trabalho, com reflexões finais.

Capítulo 2

Fundamentação Teórica

2.1 Machine Learning

2.1.1 Conceitos Gerais

Machine learning] é uma forma de programação em que a máquina simularia o aprendizado humano, gerando resultados cada vez mais precisos com uma grande quantidade de dados. Algoritmos de *machine learning* tem diversas aplicações no mundo atual, como: recomendação de conteúdo em redes sociais e *streaming*, detecção de fraudes em transações bancárias, classificação de imagens, classificação de spam, análises de negócios, análises de ações e muitas outras aplicações.

Pode-se descrever 3 métodos de *machine learning*: aprendizado supervisionado, o não-supervisionado e o semissupervisionado. No aprendizado supervisionado, o conjunto de dados é rotulado anteriormente, de forma que o algoritmo se ajusta até chegar numa eficácia adequada, tornando-se capaz de trabalhar com bons resultados no problema proposto. O funcionamento desses algoritmos pode ser separado em 3 componentes, processo de decisão, função de erro e processo de otimização. O processo de decisão é basicamente a estipulação de um padrão observado pelo algoritmo no conjunto de dados, já a função de erro, avalia os resultados obtidos por esse padrão e confere a precisão com os rótulos dos dados. O processo de otimização é a atualização das pontuações do modelo a fim de melhorar sua precisão, esse processo é repetido até que a maior precisão possível seja encontrada. No algoritmo não-supervisionado os dados não são rotulados, então cabe ao algoritmo identificar padrões e semelhanças sem conhecer a "resposta", não tendo a ajuda de uma classificação humana. No semissupervisionado, a diferença está na quantidade de dados rotulados, utilizando uma pequena parte rotulados e uma maior parte que não contém rótulos.

É importante ressaltar que existem diversos tipos de algoritmos de machine learning e cada um tem um certo tipo de problema que é mais indicado. Existem problemas, como a classificação de spam, que pedem um algoritmo com aprendizado supervisionado, por

exemplo.

2.1.1.1 Divisão e Tratamento dos Dados

Como foi citado, os dados podem ser rotulados ou não rotulados, mas outro passo relevante no tratamento desses dados é a divisão em blocos: treino, validação e teste.

Os dados de treino são a maior parte do conjunto de dados disponível e são usados para o treinamento do modelo escolhido, como citado anteriormente para encontrar o padrão nesses dados.

O bloco de validação já tem sua funcionalidade ligada a alteração e otimização de hiperparâmetros, que por sua vez, são atributos que controlam o treinamento do algoritmo e possibilitam uma generalização, evitando problemas que o algoritmo só funciona para o conjunto de dados proposto (*overfitting*) e problemas que a precisão é muito baixa (*underfitting*).

O dataset de teste já é uma avaliação do modelo e é utilizado para verificar a capacidade de trabalhar com dados que não viu antes.

2.1.2 Tipos de Algoritmos e Backpropagation

Existem vários tipos de algoritmos sendo alguns extremamente comuns para as aplicações citadas acima e, principalmente, para a análise de produtos do mercado financeiro.

2.1.2.1 Regressão Linear

É um algoritmo mais utilizado para a predição de valores, observando uma relação linear entre os dados. Os inputs nesse algoritmo são analisados para traçar uma relação com alguma variável como resultado.

2.1.2.2 Regressão Logística

Esse algoritmo faz previsões de forma binária, tendo como resultado uma probabilidade de ocorrência de um evento. A regressão logística se diferencia da regressão linear pela definição da variável resultante como uma probabilidade e por precisar de um dataset maior.

2.1.2.3 Redes Neurais

As redes neurais têm esse nome devido a tentativa de simular o cérebro humano, com vários neurônios que realizam o processamento da informação. Esses "neurônios"são chamados de nós e fazem o processamento interligado dos dados recebidos. Cada nó recebe

uma informação e atribui um peso a ela, de acordo com a sua influência na saída, após isso é feita uma soma ponderada dos sinais e a saída depende se esse valor excede um certo limite.

A correção dos pesos em cada um dos nós pode ser feita de acordo com o único exemplo apresentado (padrão) ou pelo erro médio considerando um ciclo com N exemplos (*batch*).

Vale ressaltar um algoritmo muito comum que é o MLP (*Multi Layer Perceptron*), que nada mais é do que uma rede neural com várias camadas (grupos de nós).

2.1.2.4 Backpropagation

O *backpropagation* é a propagação "para trás"do erro do algoritmo, avaliando o gradiente descendente da função de erro. A função de erro é o erro entre a saída obtida e a saída esperada (considerando um algoritmo de aprendizagem supervisionada), considerando o bloco de inputs e os parâmetros utilizados. Já o gradiente descendente é uma forma de minimizar funções diferencias e quando aplicado a função de erro, é possível achar o ponto de erro mínimo da função, realizando a correção dos pesos. Por fim, a propagação "para trás"acontece porque a primeira camada a ter seus pesos corrigidos é a última, tendo a primeira camada como a última a ser corrigida, fazendo com que a correção seja propagada de camada em camada.

2.1.3 Algoritmos de Árvores de Decisão

Os algoritmos de árvores de decisão são amplamente utilizados para problemas de regressão, sendo possível, fazer previsões com base em algumas regras de decisão. Seu funcionamento se dá, como o nome diz, com um formato de "árvore", tendo ramos e folhas, sendo formada por uma sequência de nós, que são os pontos de decisão. Os nós podem ser nós internos, que representam atributos de decisão, assim, a partir deles o dado pode ser direcionado para um ramo diferente da árvore e nós terminais, que são os nós de resposta (folhas).

O algoritmo funciona com 3 principais etapas, sendo elas a definição de um critério de parada, a avaliação dos scores dos atributos e a definição de classes nos nós. A regra de parada é uma condição que determina se aquele é um nó de resposta ou se ainda tem mais caminhos (ramos), as classes são atribuições feitas nos nós e o score é basicamente uma classificação, que identifica os atributos mais relevantes para o problema em questão. Os modelos de árvore de decisão também podem ser usados para problemas de regressão, em que o resultado é um valor numérico. Para isso, são utilizados os desvios padrão dos valores da variável target, para cada variável de predição, assim, uma variável com um pequeno desvio padrão está mais próxima da média, fazendo com que ele tenha uma importância grande na estruturação da árvore.

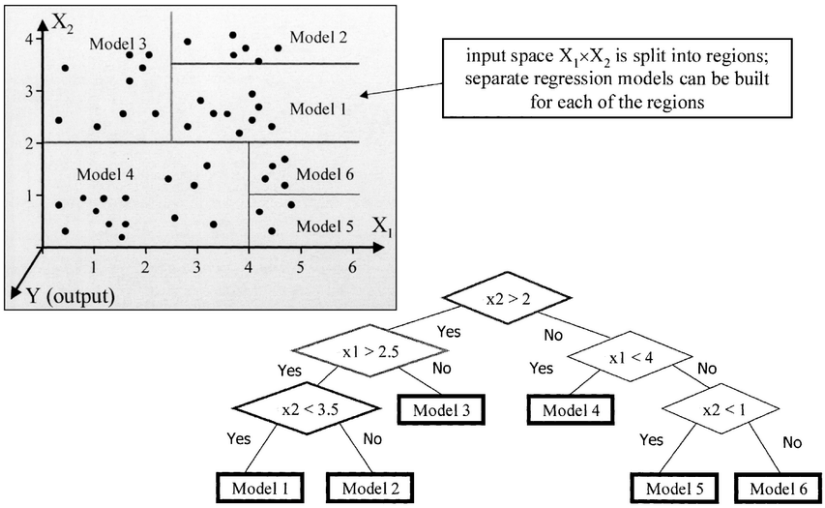

Figura 2.1: Exemplo de Árvore de Decisão

No trabalho, foram desenvolvidas várias simulações que utilizam árvores de decisão, mais especificamente técnicas de *boosting*, que são uma classificação de ensemble. Existem vários tipos de ensemble, como *Bagging*, que utiliza alguns modelos, treinados por uma parte do dataset, que devem ter características parecidas, e assim obter a média dos valores dos resultados dos modelos. Um exemplo de *Bagging* são as *Ramdom Forests*. Outro modelo muito importante são os de *Boosting*, que serão amplamente discutidos na sequência. Os algoritmos de *boosting*, treinam modelos um depois do outro, realizando alterações para melhoria dos modelos seguintes, assim, fazendo com que modelos com desempenho ruim ou simples tenham resultados melhores, já que os erros cometidos nos modelos são utilizados para correções e ajustes de peso nos seguintes.

2.1.3.1 Modelos de Boosting

No presente trabalho foram utilizados 3 modelos de *boosting*, todos pertencentes a classe de *Gradient Boosting*, o *LightGBM*, o *CatBoost* e o *XGBoost*. Os modelos de *Gradient Boosting* são formados por várias árvores de decisão e cada árvore é influenciada pela anterior, com a alteração de parâmetros, visando minimizar os erros. Essa alteração se dá pela descida do gradiente, que nada mais é que um método para minimização de uma função, no caso desses modelos, minimizando a função de erro.

Alguns dos erros mais comuns usados em problemas de regressão são: o erro quadrático médio (MSE), a raiz do erro quadrático médio (RMSE) e o erro absoluto médio (MAE). Foram utilizados nas simulações o MSE e o RMSE, que têm as definições demonstradas a seguir.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.1}$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.2}$$

Destaca-se que o objetivo do gradiente descendente é minimizar essas funções de erro, assim melhorando as árvores seguintes, levando a melhores resultados.

2.1.4 Modelos Utilizados

Como descrito na seção anterior, foram utilizados algoritmos de Boosting, sendo que, cada um deles tem origens e parâmetros diferentes. A seguir será explicado um pouco de cada modelo.

2.1.4.1 LightGBM

O *LightGBM* é um modelo desenvolvido pela *Microsoft*, com foco em acelerar o treinamento e trabalhar bem com uma grande quantidade de dados. Isso acontece porque, dentre outras especificidades, o *LightGBM* discretiza resultados contínuos, fazendo com que ele melhore a eficiência computacional, quando comparado com outros modelos.

Importante destacar quais são os hiperparâmetros do modelo *LightGBM*, sendo trabalhados nas simulações 8 deles.

- Loss: Esse parâmetro defini qual é a função de erro utilizada.
- Colsample_bytree: Esse parâmetro define quais características serão amostradas para construir cada árvore. Essa variável pode influenciar em complexidade e diversidade de cada árvore.
- Learning_rate: É a taxa de aprendizado do modelo, ela controla o tamanho da etapa de cada iteração para chegar no mínimo da função.
- Subsample: É o grupo de dados oriundos da divisão do dataset, que serão amostrados em árvores diferentes. Essa variável também pode influenciar na diversidade do modelo.
- Lambda_l1 e Lambda_l2: São as variáveis de regularização, responsáveis por penalizar os pesos dos atributos. A principal função da regularização é evitar *overfitting* e aumentar a generalização do modelo, mas pode ser prejudicial, a depender do valor, fazendo com que o modelo não se adapte aos dados propostos.
- Max_depth: É o tamanho de cada árvore, considerando o início até seu nó terminal mais longo. Esse também é um parâmetro que afeta na complexidade do modelo.

- Num_leaves: Esse parâmetro representa o número máximo de nós terminais por árvore. Também interfere na complexidade do modelo.
- Num_threads: É o número de instruções que um processo pode ter. Esse valor depende das características do computador utilizado para o treinamento.

2.1.4.2 CatBoost

O *CatBoost* é um modelo que trabalha muito bem com grandes quantidades de dados e teve seu desenvolvimento muito ligado a proteção contra *overfitting*. O *CatBoost* tem uma característica importante que é o *Ordered Boosting*, uma variação do *Gradient Boosting* dos outros modelos, em que o algoritmo cria uma ordem das classes baseada na suas importâncias para o modelo.

Vale destacar que vários hiperparâmetros são comuns ou similares aos modelos, logo serão explicado só os específicos do *CatBoost*. Os hiperparâmetros comuns com o outro modelo utilizados são: *Loss, Thread Count, Learning Rate, Depth (Max_depth), Subsalmple, Colsample Bylevel (Colsample_bytree) e L2 Leaf Reg (Lambda_l2).*

- Grow Policy: Esse parâmetro define a estratégia de crescimento da árvore, podendo assumir os valores: SymmetricTree, Depthwise, Lossguide.
- Bootstrap Type: É uma definição das amostragens dos dados que influencia na regularização e velocidade de processamento.
- Iterations: Essa variável define o número máximo de iterações de treinamento.

2.1.4.3 XGBoost

O XGBoost é um algoritmo de excelente desempenho em várias competições de machine learning e seu nome vem de Extreme Gradient Boosting.

Serão explicados os hiperparâmetros que ainda não foram descritos nos outros modelos, mas estão presentes também: Learning Rate, Objective (Loss), Max Depth, Subsample, Colsample_bytree, Reg_Alpha e Lambda, N_jobs (Num_threads) e Num_rounds (Iterations).

1. Booster: Define qual tipo de booster será utilizado, como o Gradient Boosting.
2. N_estimators: Esse parâmetros descreve o número de árvores do modelo.
3. Random_state: Essa variável serve para definir alguns números pseudoaleatórios na construção do modelo, como a inicialização de pesos. Esse parâmetro permite a reprodução do modelo,
4. Num_parallel_tree: É o número de árvores paralelas construídas no modelo.

5. Early_stopping_rounds: Define um valor de iterações que, caso não haja melhora no desempenho do modelo nesse valor, o treinamento se encerra.

6. Eval_metric: É qual critério é analisado para parar o treinamento no número de iterações dem melhora do early_stopping_rounds.

7. Features: Define como as features devem ser tratadas no treinamento do modelo, como por exemplo, aplicando todas as features em todas as situações ou deixando o modelo definir automaticamente quais aplicar.

2.1.4.4 Double Ensemble

O método de ensemble já foi explicado anteriormente e todos os algoritmos de boosting estão dentro do modelo de ensemble, porém esse 4º modelo é um double ensemble. Assim, o modelo é formado por uma lógica de boosting, mas com as "árvores fracas"sendo algum algoritmo, no caso, o LightGBM.

Os hipeparâmetros comuns desse modelo foram os seguintes: Loss, Alpha1 e Alpha2 (Lambda), epochs (iterações) e early_stopping_rounds.

1. "base_model": Define qual modelo será utilizado no ensemble, no caso, LightGBM.

2. "num_models": Define o número de modelos no ensemble, quantos LightGBM serão treinados em sequência.

3. "enable_sr"e "enable_fs": São definições para fracionamento dos dados entre os modelos e fracionamento das características avaliadas.

4. "bins_sr"e "bins_fs": Número de fracionamentos que vai para cada modelo.

5. "decay": É uma taxa que diminui a importânica de alguns modelos treinados, a fim de melhorar o desempenho.

6. "sample_ratios": Representa quanto dos dados será utilizado em cada modelo.

7. "sub_weights": Esse parâmetro define os pesos de cada modelo na decisão final.

2.1.5 Biblioteca QLIB

Para os testes e simulações foi utilizada a biblioteca Qlib, que é uma plataforma orientada por IA para investimentos quantitativos, desenvolvida pela Microsoft. Ela foi criada para facilitar o uso de IA em simulações de investimentos e auxiliar no desenvolvimento de pesquisas na área. A biblioteca oferece uma grande quantidade de recursos, desde scripts para downloads de dados para as simulações, até personalização de modelos sofisticados com avançadas técnicas de apuração.

Descreve-se algumas das funcionalidades disponíveis pela Qlib, aplicadas nos experimentos, utilizando o exemplo do código abaixo.

Listagem 2.1: Código utilizando a Qlib

```
1
2  if __name__ == "__main__":
3
4      provider_uri = "~/.qlib/qlib_data/br_data"
5      GetData().qlib_data(target_dir=provider_uri, region=REG_US,
           exists_skip=True)
6      qlib.init(provider_uri=provider_uri, region=REG_US)
7
8      market = "all"
9      benchmark = "^bvsp"
10
11     ###################################
12     # train model
13     ###################################
14     data_handler_config = {
15         "start_time": "2008-01-01",
16         "end_time": "2022-02-24",
17         "fit_start_time": "2008-01-01",
18         "fit_end_time": "2016-12-31",
19         "instruments": market,
20         "infer_processors": [],
21         "learn_processors": [
22             {"class": "Fillna"},
23             {
24                 "class": "CSRankNorm",
25                 "kwargs": {"fields_group": "label"}
26             }
27         ],
28         "label": ["Ref($close, -1) / $close"]
29     }
30
31     # Modified model configuration
32     model_config = {
33         "class": "XGBModel",
34         "module_path": "qlib.contrib.model.xgboost",
35         "kwargs": {
36             "booster": "gbtree",
37             "objective": "reg:squarederror",
38             "max_depth": 6,
39             "learning_rate": 0.03,
40             "n_estimators": 500,
41             "subsample": 0.8,
42             "colsample_bytree": 0.8,
43             "reg_alpha": 0.1,
44             "reg_lambda": 0.1,
```

```
            "random_state": 42,
            "verbosity": 0,
            "n_jobs": 4,
            "num_parallel_tree": 4,
            "early_stopping_rounds": 20,
            "eval_metric": ["rmse"],
            "num_round": 1500,
            "features": "auto",
            "model_type": "xgb",
        } ,
    }

    task = {
        "model": model_config ,
        "dataset": {
            "class": "DatasetH",
            "module_path": "qlib.data.dataset",
            "kwargs": {
                "handler": {
                    "class": "Alpha158",
                    "module_path": "qlib.contrib.data.handler",
                    "kwargs": data_handler_config ,
                },
                "segments": {
                    "train": ("2008-01-01", "2016-12-31"),
                    "valid": ("2017-01-01", "2020-12-31"),
                    "test": ("2021-01-01", "2022-02-24"),
                },
            },
        },
    }

    # model initiation
    model = init_instance_by_config(task["model"])
    dataset = init_instance_by_config(task["dataset"])

    port_analysis_config = {
        "executor": {
            "class": "SimulatorExecutor",
            "module_path": "qlib.backtest.executor",
            "kwargs": {
                "time_per_step": "day",
                "generate_portfolio_metrics": True ,
            },
        },
        "strategy": {
            "class": "TopkDropoutStrategy",
            "module_path": "qlib.contrib.strategy.signal_strategy",
            "kwargs": {
```

```
                "signal": (model, dataset),
                "topk": 20,
                "n_drop": 2,
            },
        },
        "backtest": {
            "start_time": "2021-01-01",
            "end_time": "2022-02-24",
            "account": 100000000,
            "benchmark": benchmark,
            "exchange_kwargs": {
                "freq": "day",
                "limit_threshold": 0.095,
                "deal_price": "close",
                "open_cost": 0.0005,
                "close_cost": 0.0015,
                "min_cost": 5,
            },
        },
    }

    example_df = dataset.prepare("train")
    print(example_df.head())

    # start exp
    with R.start(experiment_name="workflow"):
        R.log_params(**flatten_dict(CSI300_GBDT_TASK))
        model.fit(dataset)
        R.save_objects(**{"params.pkl": model})

        # prediction
        recorder = R.get_recorder()
        sr = SignalRecord(model, dataset, recorder)
        sr.generate()

        # Signal Analysis
        sar = SigAnaRecord(recorder)
        sar.generate()

        par = PortAnaRecord(recorder, port_analysis_config, "day")
        par.generate()
```

O código é estruturado por 6 principais funções: extração dos dados, tratamento dos dados, configurações do modelo, definição da tarefa, configuração da análise do modelo por backtest e execução das tarefas.

Na extração dos dados tem-se a a utilização da função "get_data"para recuperar um dataset previamente selecionado em "provider_uri". De posse dos dados, é possível dar início a simulação, definindo qual mercado o modelo trabalhará, no caso para dados brasileiros,

IBOV ou B3, e qual o benchmark a ser adotado.

No "data_handler_config"os dados passam por um tratamento, sendo definidas as datas para inicialização do modelo, além do uso de processadores que trabalham bem esses dados. Por último é definido o parâmetro "label"que auxiliará nas predições do modelo.

A próxima etapa é a da definição do modelo em "model_config". É aqui que é definido qual modelo será utilizado, com o caminho do script do modelo presente na biblioteca, além dos hiperparâmetros para o seu devido funcionamento.

A tarefa, representada em "task", é o que será executado, é aonde acontece a conexão do dataset trabalhado com o modelo definido, além da segmentação do dataset em dados de treino, validação e teste.

No "port_analysis_config"são definidas as métricas para análise do portfólio e backtest. Aqui é definida a estratégia, o período de teste e os valores dos custos de operação, que são considerados. A estratégia utilizada em todas as simulações foi a "TopkDropout", que consiste em selecionar um número de ações "Topk"e realizar a cada período, uma troca de um valor "n_drop"das piores ações ranqueadas, por outras melhores posicionadas.

Por fim, são executados comando de início do experimento, aplicando o modelo e salvando os resultados. Como resultado da execução, são gerados resultados como o retorno anualizado e o max drawdown, para o benchmark, o retorno do excesso do portfólio sem custos e com custos. Entenda-se como excesso o valor que passa do benchmark.

2.2 Mercado Financeiro

2.2.1 Bolsa de Valores e Instituições do Mercado Financeiro

A Bolsa de Valores é um dos principais pilares do mercado financeiro e é o ambiente aonde são negociados ativos financeiros, como ações, títulos públicos e até commodities. No Brasil, a bolsa de valores é a B3, responsável por toda a estrutura de negociação e por alguns dos principais índices do mercado, como o Ibovespa, que será melhor explicado na seção 2.2.3.

Além da B3 existem várias outras instituições relevantes para o mercado financeiro brasileiro, com destaque a CVM e as corretoras. A CVM (Comissão de Valores Mobiliários) é a grande instituição reguladora do mercado no Brasil e tem como objetivo fiscalizar, normatizar, disciplinar e desenvolver o mercado de valores mobiliários no país. Já as corretoras são instituições responsáveis por intermediar as operações entre os investidores e tomadores de recurso (pessoa física ou jurídica que necessita de recursos). Para se investir na Bolsa de Valores é necessário conhecer as normas e regulações da CVM, a fim de se ter conhecimento necessário sobre as regras de atuação no mercado, e abrir uma conta numa corretoras de títulos e valores mobiliários (CTVM) para que seja possível realizar negociações.

2.2.2 Mercado de Ações

As ações são papéis que representam uma pequena porcentagem de uma companhia com capital aberto. As ações aqui no Brasil são negociadas na B3 e podem ser compradas por meio de uma corretora, como explicado na seção anterior. Existem diversos tipos de ações, mas se destacam as ordinárias (dígito 3) e preferenciais (dígito 4). As ordinárias garantem direito a voto aos investidores e as preferenciais possuem prioridade no recebimento de dividendos e de valores em caso de liquidação da empresa, Alguns dos exemplos mais famosos são VALE3 e ITSA4. Vale citar, que dividendos são uma forma de distribuição de lucros da empresa para os seus acionistas, sendo uma parte relevante dos rendimentos de um investidor, somada com a valorização do papel.

O número de investidores no Brasil vem crescendo, com cerca de 4.6 milhões de investidores no mercado variável ao final de 2022, e com isso alguns conceitos se tornaram mais relevantes no mercado brasileiro, como o de montagem de carteira e diversificação. A montagem da carteira, nada mais é do que a seleção dos ativos para investimentos. A carteira deve ser selecionada de acordo com o perfil do investidor e seus objetivos com o investimentos. Um fator muito relevante para a definição de uma carteira é seu balanceamento por meio da diversificação. A diversificação consiste em uma forma de mitigação dos riscos, como uma analogia popular de nunca colocar todos os ovos na mesma cesta, porque se cair todos quebram. O mesmo vale para os papéis, nunca concentrar todo seu investimento em uma empresa só, ou até mesmo no mesmo setor, porque se cair, todo o seu investimento é afe-

tado. O ideal é encontrar um equilíbrio em que se uma ação cair, tem outra que está subindo e assim compensa a perda.

2.2.3 Índices e Parâmetros Relevantes

Essa seção contém alguns índices e parâmetros relevantes para o trabalho em questão, mas também para a avaliação de ações no geral. Uma ideia muito importante para se entender os investimentos no mercado de ações á a de *Risco x Retorno*. Esse conceito pode ser definido pelo retorno sendo a expectativa de receita a partir de um determinada operação e o risco como uma medida da incerteza ou volatilidade do retorno.

Para se avaliar o desempenho de um ativo é comum compará-lo com um *benchmark*. Um *benchmark* é um índice usado, além da avaliação de desempenho, para a comparação de diferentes ativos. Alguns dos *benchmarks* mais populares do mercado são o Ibovespa e o CDI (explicados abaixo). Seguem alguns outros parâmtros importantes para o desenvolvimento do presente trabalho:

1. **Ibovespa (IBOV)**: O Ibovespa é o principal índice de referência do mercado de ações brasileiro. Ele representa a média ponderada do desempenho das ações mais negociadas na B3 (Brasil, Bolsa, Balcão) e reflete as variações da cotação dessas ações ao longo do tempo. Como citado, é um dos principais *benchmarks* do Brasil.

2. **Taxa Básica de Juros**: A Taxa Básica de Juros ou Taxa Selic é a taxa de juros determinada pelo Banco Central para controlar a oferta de moeda e a inflação. Ela serve como referência para diversas operações financeiras, como empréstimos, financiamentos e investimentos.

3. **Certificado de Depósito Interbancário (CDI)**: O CDI é uma taxa utilizada como referência para remuneração de investimentos de renda fixa no Brasil. Ele representa a taxa de juros praticada em empréstimos entre bancos e também é um dos principais *benchmarks* do Brasil. Além disso, é ligada a Selic, sendo cerca de 0.10% abaixo da Selic.

4. **Volume:** O volume de ações se refere à quantidade de ações negociadas em um determinado período de tempo. É uma medida de liquidez e atividade do mercado, indicando o número de transações realizadas.

5. **Máximo DrawnDown:** É a maior perda registrada em um investimento durante um determinado período de tempo. Ele mede o pico de desvalorização máxima em relação ao valor máximo anterior.

6. **Mínimo Drawn Down:** É a menor perda registrada em um investimento durante um determinado período de tempo. Ele mede o ponto de menor desvalorização em relação ao valor máximo anterior.

7. **Beta:** O Beta é uma medida de volatilidade relativa de um ativo em relação ao mercado como um todo, no caso do Brasil, considera-se a volatilidade do mercado como o IBOV. Ele indica a sensibilidade do preço do ativo às oscilações do mercado. Um beta igual a 1 significa que o ativo se move em linha com o mercado, enquanto um beta maior que 1 indica maior volatilidade e um beta menor que 1 indica menor volatilidade.

8. **EBITDA:** O EBITDA é uma métrica financeira que representa o lucro operacional de uma empresa antes dos juros, impostos, depreciação e amortização (earnings before interests, taxes, depreciation e amortization) . É utilizado para avaliar a saúde financeira e a geração de caixa operacional de uma empresa.

9. **Dividend Yield:** Representa a relação entre os dividendos (distribuição de lucros para os acionistas) pagos por uma empresa e o preço de suas ações.

2.3 Desempenho do IBOV

Nessa seção, serão avaliados o desempenho do principal índice da bolsa brasileira, o IBOV. A gráfico abaixo, foi extraído do Yahoo Finanças e mostra o desempenho do IBOV de 2016 a 2023.

Figura 2.2: Desempenho do IBOV - https://br.financas.yahoo.com/chart/%5EBVSP

É possível notar que o índice, formado por papéis de renda variável, passa por uma volatilidade considerável, porém do ano de 2016 a 2020, a tendência era de crescimento do indicador. No ano de 2020, houve um rápido crescimento da pandemia no mundo inteiro e no Brasil, notificou-se o primeiro caso de infecção em 26/02. Nas próximas duas semanas, com o caos aumentando no mundo inteiro e a infecção se espalhando no Brasil, o mercado começou a cair muito, devido ao receio dos investidores do que estava por vir, até que no dia 12/03/2022, a bolsa registrou a maior queda desde 2008, com o acionamento de dois *circuit breakers*, mecanismo utilizados pela B3 para a interrupção de negociações devido a bruscas quedas. A partir dessa data, a queda continuou levando a bolsa a derreter no primeiro trimestre de 2020, registrando assim, uma anomalia no bloco de dados do valor do IBOV.

Capítulo 3

Estudo Empírico

Nessa seção, serão passados os precedimentos que marcaram esse trabalho, desde a obtenção de dados, a escolha e utilização dos modelos até a melhoria de hiperparâmetros e adaptações para melhoria dos resultados. Serão colocadas algumas dificuldades enfrentadas e quais soluções foram encontradas, a fim de orientar replicações e trabalhos futuros.

3.1 Obtenção dos Dados

Após a introdução com as ferramentas e estudos necessários para a compreensão dos objetivos do trabalho, se deu início a primeira parte dos experimentos: A obtenção dos dados. Uma das partes principais da solução de um problema com IA é o dataset que é utilizado para treinamento.

A própria biblioteca Qlib contém alguns scripts capazes de importar dados dos mercados americano, chinês e brasileiro. Esses scripts possibilitam o download dos dados, com a seleção da região, por exemplo para os dados dos Estados Unidos (US), utilizamos –region us, no comando para executar o script. O exemplo abaixo conta com o diretório em que os dados serão salvos em target dir.

Os dados que são salvos veem na mesma estrutura, independente da região selecionada. São baixados 3 pastas com os dados: *calendars, features e instruments*. Em calenders, estão os dias em que foram coletados os dados, em features, estão as features avaliadas de cada ação e dos respectivos índices, que são: open, close, high, low, volume, change e factor. Na parte final de instruments temos as datas que cada papel estava ativo, seja na bolsa como um todo, seja em um índice específico como o S&P500 (US) e o IBOV (BR).

Os dados do mercado brasileiro foram os mais amplamente trabalhados no presente estudo, sendo avaliados o uso do índice Ibovespa e os papéis que o compõe. Abaixo, como exemplo, estão os 20 papéis com maior participação no índice IBOV no mês de dezembro/2023:

Código	Ação	Part. (%)
VALE3	VALE	14,766
PETR4	PETROBRAS	7,214
ITUB4	ITAUUNIBANCO	6,428
PETR3	PETROBRAS	4,467
BBDC4	BRADESCO	3,731
B3SA3	B3	3,579
ELET3	ELETROBRAS	3,393
BBAS3	BRASIL	3,262
ABEV3	AMBEV S/A	2,951
RENT3	LOCALIZA	2,611
WEGE3	WEG	2,579
ITSA4	ITAUSA	2,35
PRIO3	PETRORIO	1,815
BPAC11	BTGP BANCO	1,799
EQTL3	EQUATORIAL	1,692
SUZB3	SUZANO S.A.	1,691
RADL3	RAIADROGASIL	1,685
RDOR3	REDE D OR	1,632

Um ponto importante sobre os dados é que o dataset da bolsa brasileira tem mais falhas, quando comparado com os de US e CH. Essas falhas acontecem porque o mercado brasileiro é bem menor e menos desenvolvido que esses países, fazendo com que os vários papéis sejam muito recentes ou que tenham sido encerrados ou alterados nos últimos 15 anos. Tais falhas inspiraram algumas alterações que serão melhor explicadas na seção 3.3.2, visando o melhor tratamento e melhoria dos resultados.

3.2 Utilização Qlib - US e CN

Após a obtenção dos dados, foi iniciado o período de simulações. De acordo com pesquisas e sendo alguns dos algoritmos mais indicados para a análise de problemas como o em questão, foram selecionados o XGBoost, Catboost e LightGBM. Com esse algoritmos, inicialmente, foram simuladas situações com os dados dos Estados Unidos e da China.

As primeiras simulações foram realizadas com arquivos YAML (uma linguagem de serialização de dados) , já que a Qlib fornece uma possibilidade de execução por meio do uso da função *qrun*. Abaixo, segue o código de uma simulação com o modelo XGBoost, importante notar as funções utilizadas, que foram previamente explicadas na seção de Fundamentação Teórica.

Listagem 3.1: YAML com simulção para o dados americanos

```yaml

qlib_init:
    provider_uri: "~/.qlib/qlib_data/us_data"
    region: us
market: &market SP500
benchmark: &benchmark ^dji
data_handler_config: &data_handler_config
    start_time: 2008-01-01
    end_time: 2019-12-31
    fit_start_time: 2008-01-01
    fit_end_time: 2014-12-31
    instruments: *market
port_analysis_config: &port_analysis_config
    strategy:
        class: TopkDropoutStrategy
        module_path: qlib.contrib.strategy
        kwargs:
            topk: 20
            n_drop: 2
            signal: <PRED>
    backtest:
        start_time: 2017-01-01
        end_time: 2019-12-31
        account: 100000000
        benchmark: *benchmark
        limit_threshold: 0.095
        deal_price: close
        open_cost: 0.0005
        close_cost: 0.0015
        min_cost: 5

task:
    model:
        class: XGBModel
        module_path: qlib.contrib.model.xgboost
        kwargs:
            booster: gbtree
            objective: reg:squarederror
            max_depth: 6
            learning_rate: 0.05
            n_estimators: 100
            subsample: 0.9
            colsample_bytree: 0.9
            reg_alpha: 0.1
            reg_lambda: 1.0
            random_state: 42
            verbosity: 0
            n_jobs: 4
            num_parallel_tree: 4
```

```
            early_stopping_rounds: null
            eval_metric: [rmse]
            num_round: 100
            features: all
            model_type: xgb
    dataset:
        class: DatasetH
        module_path: qlib.data.dataset
        kwargs:
            handler:
                class: Alpha158
                module_path: qlib.contrib.data.handler
                kwargs: *data_handler_config
            segments:
                train: [2008-01-01, 2014-12-31]
                valid: [2015-01-01, 2016-12-31]
                test: [2017-01-01, 2019-12-31]
    record:
        - class: SignalRecord
          module_path: qlib.workflow.record_temp
          kwargs: {}
        - class: PortAnaRecord
          module_path: qlib.workflow.record_temp
          kwargs:
              config: *port_analysis_config
```

3.3 Dados do Mercado Brasileiro e Alterações

O objeto de maior estudo desse trabalho é o mercado brasileiro, com isso, finalizados os testes com os dados americanos e chineses, foi feita uma adaptação para arquivos python e não mais em YAML. Os arquivos python permitem mais liberdade para testes e prints de averiguação de como está sendo desenvolvido o modelo.

Para utilização dos dados brasileiros foram necessárias algumas alterações na função path do script. A primeira delas alterando o "provider_uri"que traz o local dos arquivos, sendo alterado pra pasta que traz os dados do Brasil, depois a alteração do market, aqui consideraremos o os papéis que compuseram o Ibovespa (no código determinado por "ibov") e o benchmark, o próprio índice Ibovespa (no código determinado por "b̂vsp". Para as simulações com os dados de todos os papéis que compõe a B3 foi necessária a alteração do market de "ibov"para "all". Para a interpretação dos resultados, considerou-se "B3"como a representação dos papéis de toda a bolsa brasileira.

Assim foram realizados as primeiras simulações usando os modelos LightGBM, Catboost e XGBoost, posteriormente foi simulado também um modelo de Ensemble, com o LightGBM. O trabalho foi dividido em 2 experimentos, o primeiro que visa analisar os im-

pactos dos dados da pandemia (ano de 2020) no dataset de teste, em cada um dos modelos, e o segundo que visa analisar os impactos desses dados no dataset de treino.

O primeiro experimento teve o dataset dividido com treinamento e validação até o final de 2017, com variação do período em que os dados eram treinados, sendo antes da pandemia ou considerando todo o período disponível. Já o segundo experimento, a variação se deu na inclusão ou não do ano de 2020 nos dados de treino e validação, com o período de teste sendo de 2021 até início de 2022. Todos os modelos foram simulados com os dados do Ibovespa e da B3 como um todo. Essa divisão do dataset será mais explorada na seção 3.3.1.

Assim como a divisão do dataset, foram realizadas outras alterações visando a melhoria dos modelos, com a alteração da estratégia e a mudança dos processadores, que com a utilização dos dados brasileiros, que tinham várias falhas, se mostraram muito importantes. Por fim, foram realizados vários testes para a otimização dos hiperparâmetros de cada modelo, na tentativa de melhorar os respectivos desempenhos.

3.3.1 Divisão do Dataset

Os dados utilizados nas simulações dos dados brasileiros foram de janeiro de 2008 até fevereiro de 2022. Os datasets foram divididos considerando os 2 principais experimentos realizados, sendo eles:

- Experimento 1 - Dataset considerando o impacto dos dados da pandemia no dataset de teste. Segue a divisão dos dados, sendo os dados para treino e validação iguais para as duas partes do experimento e a diferença no período de teste.
  - Parte 1 - Teste Pré-Pandemia
    - Dados de treino: 2008-01-01 até 2014-12-31
    - Dados de validação: 2015-01-01 até 2016-12-31
    - Dados de teste: 2017-01-01 até 2019-12-31
  - Parte 2 - Teste com a Pandemia
    - Dados de treino: 2008-01-01 até 2014-12-31
    - Dados de validação: 2015-01-01 até 2016-12-31
    - Dados de teste: 2017-01-01 até 2022-02-24
- Experimento 2 - Dataset considerando o impacto da pandemia no dataset de treino e validação. Segue a divisão dos dados, sendo os dados para teste iguais para as duas partes do experimento e a diferença no período de treino e validação.
  - Parte 1 - Não incluindo 2020
    - Dados de treino: 2008-01-01 até 2016-12-31
    - Dados de validação: 2017-01-01 até 2019-12-31

    - Dados de teste: 2021-01-01 até 2022-02-24
- Parte 2 - Incluindo 2020
    - Dados de treino: 2008-01-01 até 2016-12-31
    - Dados de validação: 2017-01-01 até 2020-12-31
    - Dados de teste: 2021-01-01 até 2022-02-24

3.3.2 Estratégia e Processadores

Para a melhoria dos modelos com dados do Brasil, foram alterados vários parâmetros e alguns dos principais foram alterações na estratégia e em alguns dos processadores.

A estratégia utilizada foi a TopKDropoutStartegy, que como explicado anteriormente, consiste em uma estratégia que seleciona um número "Topk"específico de ações para compor a carteira e um número "Drop"que são as ações a serem trocadas a cada período de negociação. As primeiras simulações foram com 50 papéis sendo trocados os 5 com pior ranking após uma nova avaliação, ou seja, "Topk"igual a 50 e "Drop"igual a 5. Vários testes foram realizados, visando diminuir o tamanho da carteira, assim aumentando a concentração, mas potencialmente aumentando os ganhos. Os exemplos disponíveis na Qlib trazem muito esse valor de 50/5, mas com os índices do exterior que tem muito mais ações, já o IBOV conta com cerca de 85 papéis, assim, fazendo com que haja a necessidade de diminuir um pouco a quantidade de papéis na carteira. Ao final de vários testes, com carteiras mais concentradas e menos concentradas, a estratégia escolhida foi com o "Topk"igual a 20 e "Drop"igual a 2, apesar de uma pequena variação dependendo do modelo, essa foi a combinação que melhor funcionou com os dados brasileiros.

Em relação aos processadores, também foi um mudança que trouxe impactos relevantes para os resultados. Primeiro, uma necessidade dos dados brasileiros era um tratamento melhor, muitos dados como "NaN"(Not a Number), devido a falhas nos dados e, principalmente, o mercado brasileiro ser "pequeno", com ações novas, muitas ações que sofrem alguma mudança e algumas que são encerradas. Para isso, foram utilizados dois tipos de processadores "DropnaLabel"e "Fillna", o primeiro, retirava dos dados os papéis que tinham algum valor como "NaN", já o segundo preenchia o espaço com o valor 0. A partir de simulações foi verificado que o "Fillna"apresentava melhores resultados e por isso foi escolhido para a comparação dos modelos.

O segundo tipo de processador que fez uma diferença considerável no resultado foram alguns learn processors: RobustZScoreNorm e CSRankNorm. As primeiras simulações utilizaram o primeiro tipo desse processador, realizando a normalização Z-score, não obteve resultados bons, assim, realizou-se a troca para o CSRankNorm que teve resultados substancialmente superiores, se mostrando mais adequado para o dataset em questão.

3.3.3 Melhoria de Hiperparâmetros

Para melhora dos modelos foi necessária a otimização dos hiperparâmetros. Em cada um dos modelos, vários testes foram realizados até que chegasse nos melhores desempenhos observados. Para o modelo do LightGBM, esses foram os valores finais para os hiperparâmetros:

Listagem 3.2: Hiperparâmetros do modelo de LightGBM

```
1
2  model_config = {
3          "class": "LGBModel",
4          "module_path": "qlib.contrib.model.gbdt",
5          "kwargs": {
6              "loss": "mse",
7              "colsample_bytree": 0.7,
8              "learning_rate": 0.01,
9              "subsample": 0.7,
10             "lambda_l1": 5,
11             "lambda_l2": 5,
12             "max_depth": 5,
13             "num_leaves": 50,
14             "num_threads": 20
15         },
16     }
```

As alterações nesse primeiro modelo foram realizadas, principalmente, visando evitar o overfitting e pra diminuir a complexidade do modelo com a redução de alguns parâmetros iniciais. Dos iniciais, foram diminuídos todos os parâmetros com exceção do "num_threads". Os valores de lambda, foram reduzidos para dar mais flexibilidade ao modelo e permitir uma menor regularização, já a "learning_rate"foi reduzida visando uma melhor generalização do modelo e os outros hiperparâmetros foram reduzidos para diminuir a complexidade do modelo.

Para o modelo CatBoost foram definidos os seguintes parâmetros:

Listagem 3.3: Hiperparâmetros do modelo de Catboost

```
1
2  model_config = {
3          "class": "CatBoostModel",
4              "module_path": "qlib.contrib.model.catboost_model",
5              "kwargs": {
6                  "loss": "RMSE",
7                  "thread_count": 20,
8                  "grow_policy": "SymmetricTree",
9                  "bootstrap_type": "Bernoulli",
10                 "learning_rate": 0.1,
11                 "iterations": 2000,
12                 "depth": 6,
```

```
                "subsample": 0.8,
                "colsample_bylevel": 0.8,
                "l2_leaf_reg": 3

        },
    }
```

Nesse segundo modelo, as alterações mais impactantes foram no aumento da quantidade de iterações, na "grow_policy"que começou como DepthWise e mostrou um resultado melhor com SymmetricTree. Além dessas adaptações, foi adicionado um valor de regularização visando trabalhar com a flexibilidade do modelo, podendo ajudar a evitar um overfitting.

O terceiro modelo utilizado foi o XGBoost, esses foram os hiperparâmetros finais:

Listagem 3.4: Hiperparâmetros do modelo de XGBoost

```

model_config = {
        "class": "XGBModel",
        "module_path": "qlib.contrib.model.xgboost",
        "kwargs": {
            "booster": "gbtree",
            "objective": "reg:squarederror",
            "max_depth": 6,
            "learning_rate": 0.03,
            "n_estimators": 500,
            "subsample": 0.8,
            "colsample_bytree": 0.8,
            "reg_alpha": 0.1,
            "reg_lambda": 0.1,
            "random_state": 42,
            "verbosity": 0,
            "n_jobs": 4,
            "num_parallel_tree": 4,
            "early_stopping_rounds": 20,
            "eval_metric": ["rmse"],
            "num_round": 1500,
            "features": "auto",
            "model_type": "xgb",
        } ,
    }
```

Nesse terceiro modelo foram feitas algumas mudanças para diminuir a regularização e aumentar o número de árvores, além de alterar a quantidade de iterações e da "learning_rate". Os valores de "subsample"e "colsample_bytree"foram reduzidos para compensar um pouco da regularização diminuída na "reg_alpha"e "reg_lambda", o "num_estimators", "num_parallel_trees"e "num_rounds"foram aumentados visando aumentar a diversidade e aprendiado das árvores.

O quarto e último modelo utilizou a técnica de Ensemble, explicada anteriormente. O

modelo utilizado foi o LightGBM.

Listagem 3.5: Hiperparâmetros do modelo de Ensemble

```
model_config = {
        "class": "DEnsembleModel",
        "module_path": "qlib.contrib.model.double_ensemble",
        "kwargs": {
            "base_model": "gbm",
            "loss": "mse",
            "num_models": 6,
            "enable_sr": True,
            "enable_fs": True,
            "alpha1": 1.0,
            "alpha2": 1.0,
            "bins_sr": 10,
            "bins_fs": 5,
            "decay": 0.95,
            "sample_ratios": [0.8, 0.7, 0.6, 0.5, 0.4],
            "sub_weights": [1, 1, 1, 1, 1, 1],
            "epochs": 150,
            "early_stopping_rounds": 20
        },
    }
```

Para a melhora desse modelo, foi diminuído o "decay", que aumenta a importância de modelos antigos no resultado, foi aumentado o número de "epochs"para tentar aumentar a aprendizagem e adicionado um "early_stopping_rounds"para evitar overfitting e melhorar o desempenho.

Vale lembrar que essas alterações foram realizadas por meio de testes, almejando melhores desempenhos, porém podem existir combinações melhores. O foco aqui é avaliar como modelos com bom desempenho são impactados pela pandemia, que foi um evento inesperado e externo ao desempenho dos papéis.

Capítulo 4

Análise dos resultados

Nessa seção, serão apresentados os resultados dos experimentos, acompanhados de uma análise que explica um pouco das relações de desempenho de cada modelo. Na seção 4.1, serão mostrados os resultados de uma simulação com os dados americanos, utilizando um dos modelos, o CatBoost. Essa simulação vai servir como exemplo de como os resultados devem ser interpretados e como faremos as análises dos dados brasileiros.

4.1 Simulações US e CN

Os primeiros testes realizados com dados chineses e americanos apresentaram resultados diversos. Os resultados das simulações é medido via backtest e traz três grupos de resultados: os resultados do benchmark selecionado, o resultado do modelo sem considerar o custo das operações e o resultado incluindo custo das operações. As principais métricas observadas foram o "annualized_return"que indica o desempenho do portfólio no período testado e o "max_drawndown", que indica a maior queda da carteira no período.

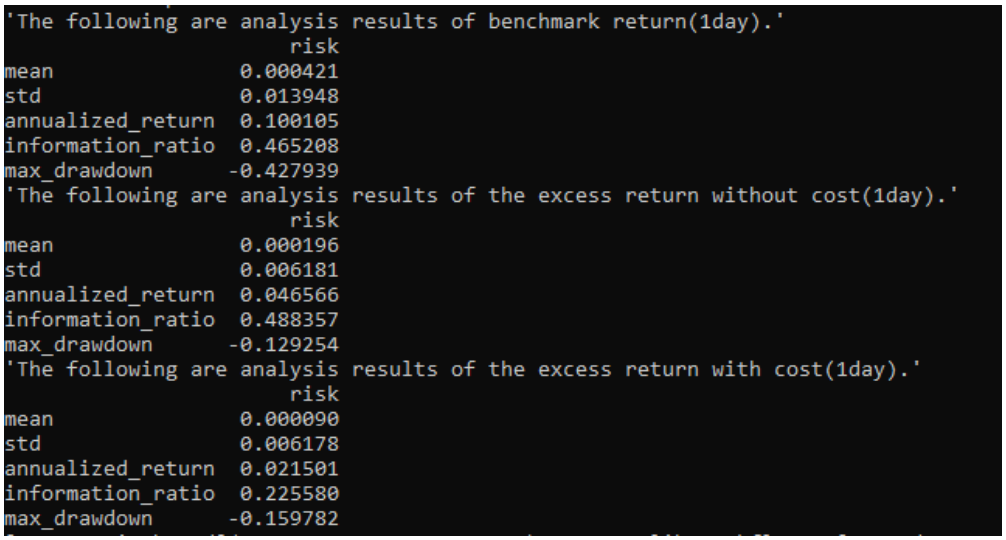

Figura 4.1: Resultados CatBoost - US

As análises foram focadas na comparação dos resultados do benchmark com os resultados do portfólio já considerando os custos.

Inicia-se as análises com o "annualized_return", em que tivemos o benchmark (Índice Dow Jones) com retorno de 0.1001 e o retorno do portfólio com 0.0215. Para interpretar esse resultado utiliza-se a seguinte expressão:

$$return\_real = return\_benchmark * 100 * (1 + return\_portfolio) \tag{4.1}$$

em que o retorno do benchmark já é apresentado com a porcentagem, sendo necessária somente a multiplicação por 100, que no caso resulta em um retorno de 10,01%. Já o retorno do portfólio que é exibido, é uma parcela do retorno do benchmark. No caso temos 0.0215, substituindo na equação acima temos:

$$return\_real = 10.01 * (1.0215) \tag{4.2}$$

$$return\_real = 10.225\% \tag{4.3}$$

Assim, o retorno do modelo foi 0.215% acima do benchmark, com um retorno total de 10.225%. Entenda-se que quanto maior o retorno, melhor o resultado, já que, na prática o rendimento do portfólio seria maior.

Para a análise do Max Drawdown (MDD), o racional é quase o mesmo, com uma pequena mudança de sinal, descrita na equação abaixo.

$$MDD\_real = MDD\_benchmark * 100 * (1 - MDD\_portfolio) \tag{4.4}$$

Assim, no caso desta seção, tem-se que o Max Drawdown do benchmarck foi -0.4279, ou -42.79%, lembrando que os dados de teste englobavam a pandemia. Já o modelo teve um Max Drawndown de -0.1597, substituindo na equação:

$$MDD\_real = -42.79 * (1.1597) \tag{4.5}$$

$$MDD\_real = -49.62\% \tag{4.6}$$

Interpreta-se o resultado do MDD como, quanto maior melhor, já que representa a maior queda em um período. Se o MDD estiver mais próximo de zero, representa pouco risco e uma boa notícia para o investidor.

Com esses dois indicadores, é possível avaliar e comparar os diferentes modelos, de acordo com desempenho, risco e volatilidade. Nas próximas seções, serão dispostos os re-

sultados dos experimentos observando, principalmente, esses indicadores amplamente abordados, comparando os resultados dos benchmarks e do modelo considerando os custos.

4.2 Experimento 1 - Parte I

Nessa primeira parte, foi avaliado o desempenho dos modelos, testando-os antes da pandemia, até o final de 2019. O desempenho do benchmark avaliado (Ibovespa) está descrito na imagem 4.2.

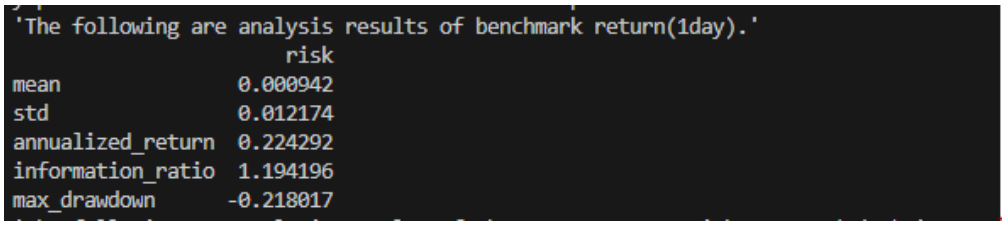

Figura 4.2: Benchmark - IBOV até 2019

Foi observado que o retorno anualizado do período foi de 22.42% e o Max Drawdown de -21.80%. Possuindo esse dados, foi possível comparar com os resultados dos diversos modelos.

Na avaliação dos modelos detalham-se 2 resultados, o primeiro considerando somente os papéis que compõe o IBOV e o segundo considerando todos os papéis da B3. Os dados desse benchmark servirão para todos as simulações dessa parte I.

4.2.1 LightGBM

Nesse primeiro modelo, verifica-se os resultados do modelo LightGBM.

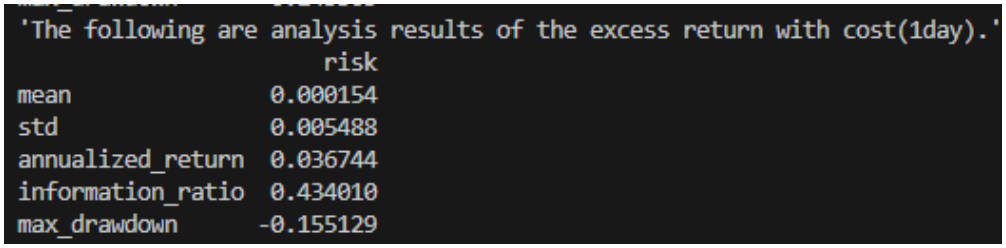

Figura 4.3: LightGBM com IBOV até 2019

Com dados até o final do ano de 2019, modelo de com LightGBM teve um desempenho interessante de +0.0367, o que realizando as contas da equação 4.1, tem-se que o retorno do modelo é de 23.25%, ou seja, um ganho de +0.82% acima do benchmark. O MDD teve como resultado do modelo -25.18%, 3.38% menor que o observado no benchmark.

A seguir confere-se os resultados com os dados da B3.

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean                0.000584
std                 0.010767
annualized_return   0.138935
information_ratio   0.836464
max_drawdown       -0.192879
```

Figura 4.4: LightGBM com B3 até 2019

Já na simulação usando os dados da B3 como um todo, o modelo teve um desempenho de +0.1389, o que é melhor que com os dados do IBOV, totalizando um retorno de 25.54%, 3.11% acima do benchmark, o que já representa um ganho muito relevante. O MDD do modelo foi de - 26.00%, sendo assim, 4.2% abaixo do Ibovespa.

4.2.2 CatBoost

O segundo modelo foi o CatBoost e assim foram expostos seus resultados:

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean                0.000162
std                 0.005462
annualized_return   0.038568
information_ratio   0.457730
max_drawdown       -0.126674
```

Figura 4.5: CatBoost com IBOV até 2019

No mesmo período, o CatBoost teve desempenho ligeiramente superior ao LightGBM, com retorno de 0.0385 sobre o benchmark, que totaliza 23.29%, um aumento de 0.86% do Ibovespa. Já na análise de risco, teve-se o MDD também um pouco melhor que o do LightGBM, com um total de -24.56%, representando um diferencial de -2.75%.

A seguir confere-se os resultados com os dados da B3.

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean                0.000154
std                 0.008906
annualized_return   0.036764
information_ratio   0.267579
max_drawdown       -0.239853
```

Figura 4.6: CatBoost com B3 até 2019

O CatBoost não apresentou melhora com os dados da B3, tendo inclusive uma leve piora para um desempenho de 23.52%, um aumento de 0.823%, bem próximo do desempenho da

primeira simulação do LightGBM. Já no MDD, notou-se também uma piora, com resultado de -27.028%, uma diminuição do benchmark de 5.227%.

Percebeu-se que o modelo CatBoost, teve desempenho um pouco melhor que o LightGBM com os dados do IBOV, mas com a B3 um desempenho bem pior.

4.2.3 XGBoost

O terceiro modelo utilizado foi o XGBoost. Seguem os resultados para o IBOV.

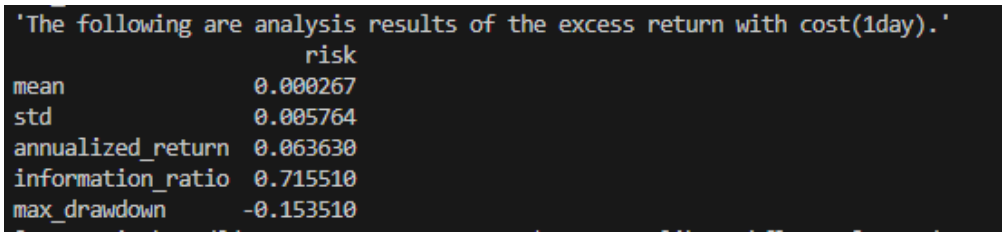

Figura 4.7: XGBoost com IBOV até 2019

Esse foi o modelo com o melhor desempenho no cenário de papéis do IBOV, com um retorno total de 23.85%, um aumento de 1.43% do benchmark, que já é um bom retorno. O MDD também teve um bom resultado, comparando com os outros modelos, com um valor total de -25.146%, pior que o Ibovespa -3.34%.

A seguir confere-se os resultados com os dados da B3.

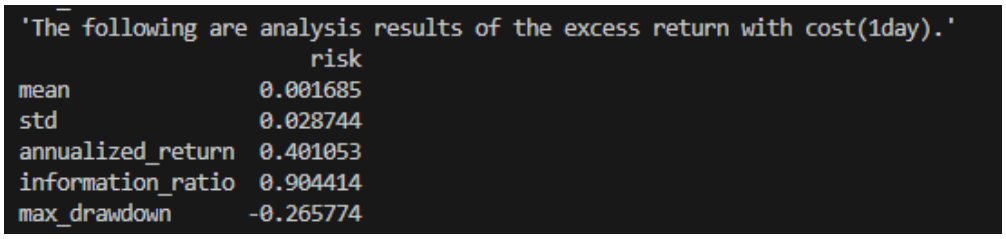

Figura 4.8: XGBoost com B3 até 2019

Para os dados da B3, o XGBoost teve como resultado um retorno anualizado de 0.401, muito acima que os outros modelos, totalizando um retorno de 31.42%. O aumento em relação ao benchmark foi de 8.99%, reapresentando um excelente retorno. O MDD apresentou uma queda um pouco maior que os outros modelos de -0.2657, totalizando -27.59%, o menor valor registrado nessa parte I.

4.2.4 Ensemble

O modelo utilizando o método de Ensemble com o LightGBM teve os seguintes resultados.

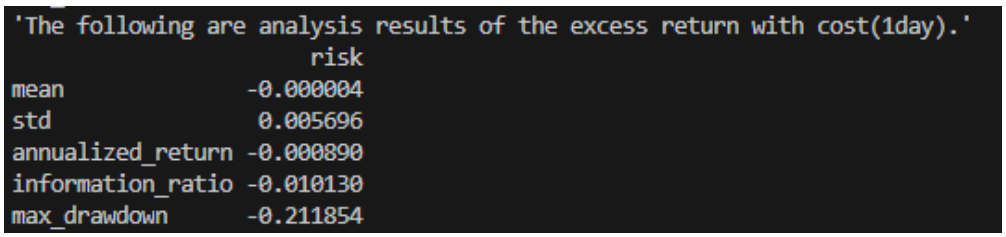

Figura 4.9: Ensemble com IBOV até 2019

Com os papéis do IBOV, o modelo co ensemble teve o pior resultado deste cenário, com um retorno totalizando 22.40%, acompanhando o benchmark. O MDD teve também o pior resultado, com o menor valor dos dados do IBOV com -26.41%, uma variação de -4.61%.

Com os dados da B3, esse modelo conseguiu desempenhar melhor, conforme demostrado a seguir.

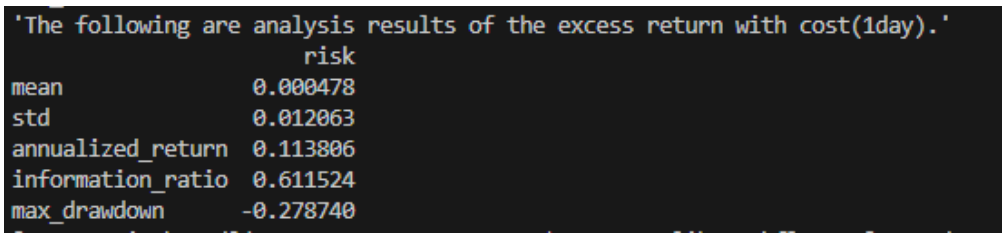

Figura 4.10: Ensemble com B3 até 2019

O retorno superior ao visto na outra simulação foi de 0.1138, que resulta em um retorno anual de 24.98%, um diferencial muito bom de 2.55%. Já o MDD foi o menor deste cenário com 27.87%, um diferencial de relevantes -6.07%.

4.2.5 Compilado dos Resultados Experimento 1 - Parte I

Nessa seção, encontra-se a tabela de comparação dos resultados dessa primeira parte.

Tabela 4.1: Compilado de Resultados com Teste até 2019

Modelo	Fonte de Dados	Retorno Total (%)	Δ de Retorno (%)	MDD (%)	Δ de MDD (%)
LightGBM	IBOV	23,25	0,82	-25,18	-3,38
	B3	25,54	3,11	-26,00	-4,20
CatBoost	IBOV	23,29	0,86	-24,56	-2,76
	B3	23,52	0,83	-27,03	-5,23
XGBoost	IBOV	23,85	1,42	-25,15	-3,35
	B3	31,42	8,99	-27,59	-5,79
Ensemble	IBOV	22,41	-0,02	-26,41	-4,61
	B3	24,98	2,55	-27,87	-6,07

Verificou-se nessa primeira parte do primeiro experimento que os modelos tiveram um bom desempenho em todas as situações. O XGBoost foi o modelo que apresentou o melhor desempenho nesse cenário, que contava com um dataset com menos imperfeições, já que a pandemia não fez parte dessa primeira parte. Além do destaque ao XGBoost, notou-se que quase todos os modelos, com a exceção do CatBoost, tiveram uma melhora nas simulações com os dados da B3. Uma hipótese para essa ocorrência é a maior quantidade e diversidade dos dados, o que pode favorecer esses tipos de algoritmo de árvore, principalmente e boosting.

4.3 Experimento 1 - Parte II

Nessa segunda parte do experimento, foram simulados cenários considerando o período da pandemia, mantendo-se os mesmos dados de teste e validação, mas com o período de teste se estendendo até 2022. O objetivo ao final desse experimento é avaliar os impactos da pandemia, que representa um grande outlier nos dados, um evento inesperado e externo as informações analisadas nesses modelos, têm nos desempenhos dos modelos avaliados e comparar com o que foi visto com os dados de pré-pandemia.

O Ibovespa teve os seguintes resultados no novo período de teste, que considera a pandemia.

```
'The following are analysis results of benchmark return(1day).'
                      risk
mean              0.000646
std               0.016555
annualized_return  0.153844
information_ratio  0.602377
max_drawdown      -0.575330
```

Figura 4.11: Benchmark - IBOV até 2022

Já é possível identificar um primeiro sinal muito relevante, o indicador de risco avaliado, o Max Drawdown teve o valor de -57.35, o que quer dizer quem em determinado período, as perdas do índice foram de -57.35%. Além disso, o retorno anualizado do período também foi pior, com -15.38%.

4.3.1 LightGBM

O modelo LightGBM com o IBOV, sendo testado no período da pandemia teve os seguintes resultados.

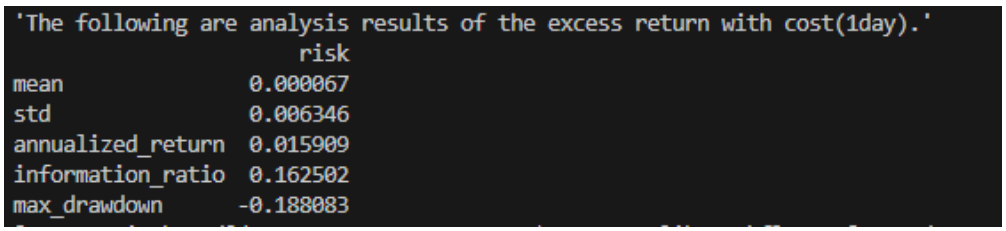

Figura 4.12: LightGBM com IBOV até 2022

Essa primeira simulação teve um resultado interessante, com 0.0159, mas que representa um acréscimo e 0.244% no retorno do benchmark, totalizando 15.62%. O MDD foi bem considerável de -68.34%, uma queda de -10.82%.

Seguem os resultados do modelo com os dados da B3.

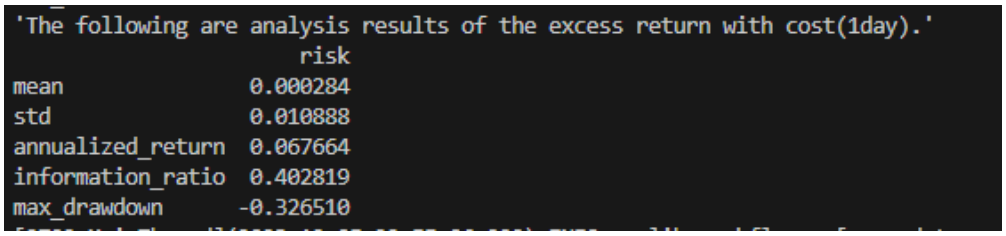

Figura 4.13: LightGBM com B3 até 2022

Nota-se que os o retorno foi melhor, sendo até superior ao modelo CatBoost da parte I. O retorno total foi de 16.42%, gerando um acréscimo de 1.03% no benchmark. O MDD teve o menor valor das simulações, com -0.3265, resultando em -76.31%, uma variação de -18.78%.

4.3.2 CatBoost

Foi-se simulado o CatBoost com o período da pandemia

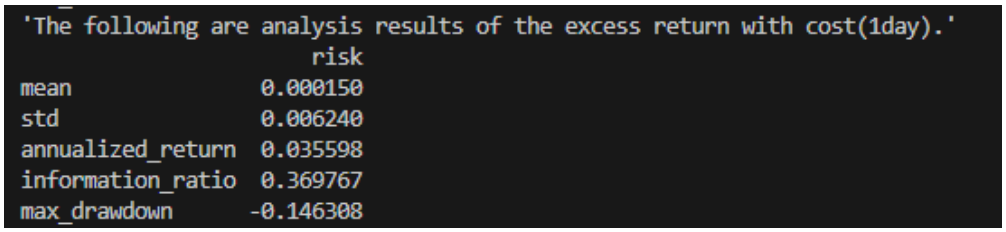

Figura 4.14: CatBoost com IBOV até 2022

O retorno desse modelo foi bem parecido com o resultado da parte I do experimento, tendo 0.0355, totalizando 15.92%, um acréscimo de 0.54% no benchmark. Já o MDD foi proporcionalmente baixo, de -0.1463, porém totalizando um alto valor de -65.94%, representando uma variação de -8.41%.

Verifica-se a seguir o resultado com os dados da B3.

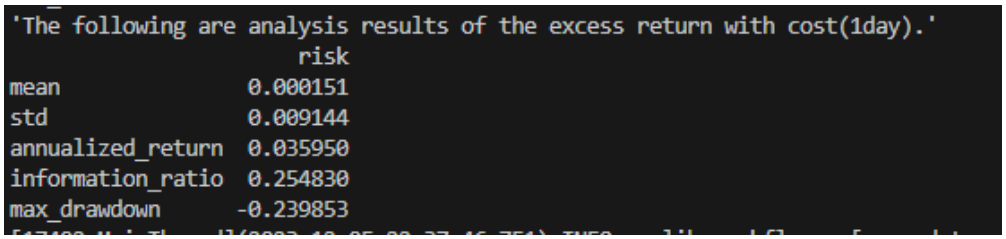

Figura 4.15: CatBoost com B3 até 2022

O resultado com o dataset da B3 foi bem similar ao visto nas outras simulações do CatBoost, com 0.0359 de retorno, resultando em 15.93% de retorno anualizado, com variação de 0.55%. O MDD caiu -0.2398, gerando uma queda total de -71.32%, o que representa uma variação de -13.76%.

4.3.3 XGBoost

Demonstra-se os resultados do modelo XGBoost, que teve o melhor desempenho na parte I.

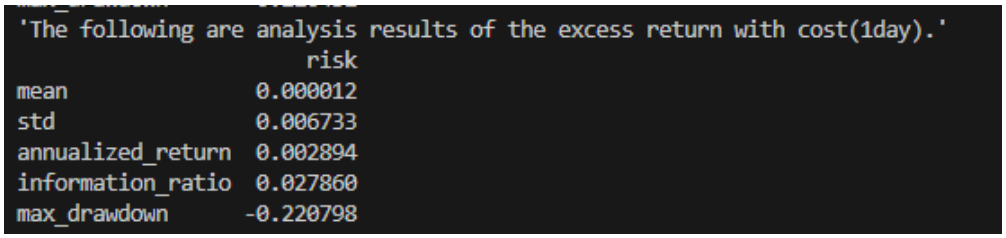

Figura 4.16: XGBoost com IBOV até 2022

O retorno do XGBoost nesse cenário de pandemia, praticamente acompanhou o benchmark, com total de 15.42%, variando 0.04%. Já o MDD ficou numa média esperada, de -0.2207, fazendo o MDD do modelo chegar a -70.22%, variando -12.69%.

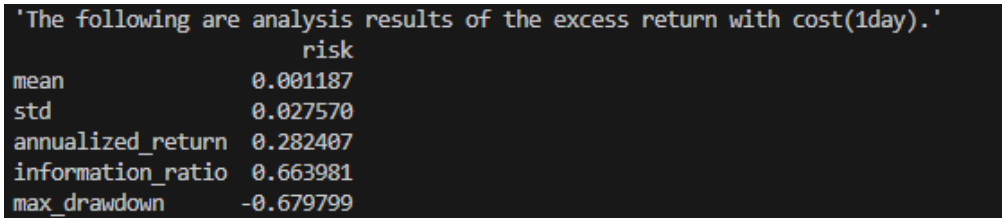

Figura 4.17: XGBoost com B3 até 2022

Com o dataset de todos os papéis da B3, o XGBoost entregou um excelente resultado, com um retorno alto de 19.72%, ficando acima do benchmark 4.43%. Porém o risco foi muito alto, em determinado momento da simulação a perda chegou a ser de -96.63%, um diferença de -39.10% do benchmark. O retorno foi alto, mas o risco foi o maio simulado.

4.3.4 Ensemble

Com o método esemble, teve-se os seguintes resultados.

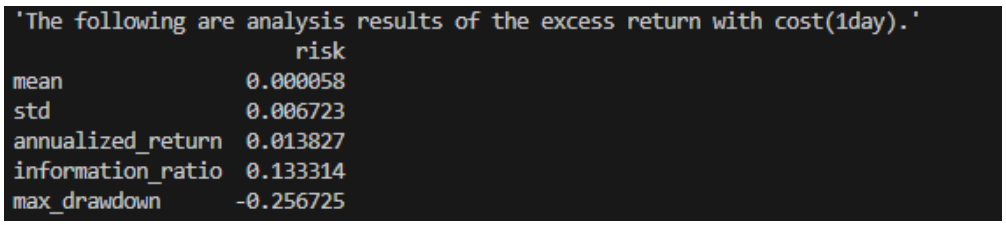

Figura 4.18: Ensemble com IBOV até 2022

O método de ensemble trouxe um retorno de 0.0159, que representa 15.62%, um ganho de 0.244%, bem próximo do desempenho do LightGBM nesse cenário. O MDD foi de -72.30%, com variação de -14.77%, mais uma vez uma perda máxima muito alta.

A seguir, checa-se com os dados da B3.

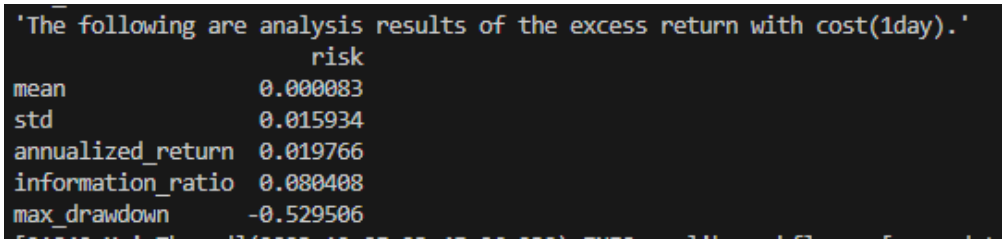

Figura 4.19: Ensemble com B3 até 2022

Para os dados da B3, o retorno anualizado foi bem estável, similar a simulação com os dados do IBOV, com retorno total de 15.68%, variando 0.3%. O MDD foi de -87.99%, com variação de -30.46%, um dos menores valores das simulações.

4.3.5 Compilado de Resultados Experimento 1 - Parte II

A Tabela 2 compara os resultados das simulações dessa segunda parte.

Tabela 4.2: Compilado de Resultados com Testes na Pandemia

Modelo	Fonte de Dados	Retorno Total (%)	Δ de Retorno (%)	MDD (%)	Δ de MDD (%)
LightGBM	IBOV	15,62	0,24	-68,34	-10,81
	B3	16,42	1,04	-76,31	-18,78
CatBoost	IBOV	15,92	0,54	-65,94	-8,41
	B3	15,93	0,55	-71,32	-13,79
XGBoost	IBOV	15,42	0,04	-70,22	-12,69
	B3	19,72	4,34	-96,63	-39,10
Ensemble	IBOV	15,62	0,24	-72,30	-14,77
	B3	15,68	0,30	-87,99	-30,46

Nessa parte, pode-se analisar que devido aos efeitos da pandemia o max drawdown dos modelos foi muito alto, demonstrando um cenário de maior risco e volatilidade, o que já era esperado.

Um destaque dessa simulação foi a consistência do modelo CatBoost, em diferentes cenários, com datasets diferentes e períodos de teste diferentes, ele entregou resultados bem próximos de retorno. Um possível explicação para esse desempenho consistente pode ser uma característica do CatBoost que permite a otimização automática de hiperparâmetros e a combinação de features, o que pode tornar o modelo mais flexível a entradas diferentes, tornando-o mais adaptável.

Apesar do destaque para o desempenho do CatBoost, o melhor retorno aconteceu com o XGBoost com os dados da B3, assim como na parte I.

4.3.6 Conclusões Experimento 1

Foi observado que o desempenho dos modelos foi superior, de forma geral, no período pré-pandemia, como esperado. No teste com a pandemia os retornos diminuíram consideravelmente, com exceção do bom desempenho do CatBoost, que manteve uma regularidade que é admirável devido a diversidade dos testes. Pode-se concluir que a pandemia, que configura um outlier no dataset, afeta os desempenhos dos modelos e adiciona muito risco aos portfólios, porém a expectativa era que o impacto fosse muito maior que o observado, já que todos os modelos superaram o benchmark. Além disso, uma possível conclusão desse primeiro experimento é a importância da quantidade de dados e de um grau de diversidade, o que possibilita os modelos a trabalhar um pouco melhor.

Com isso, conclui-se que os modelos treinados apresentam uma certa resistência a situações adversas e são capazes de entregar um desempenho razoável.

4.4 Experimento 2 - Parte I

No segundo experimento, foi avaliado o impacto da inclusão dos dados da pandemia nos datasets de treino e validação. O período de teste foi o mesmo para todas as simulações, 2021 até fevereiro de 2022.

O benchmark nesse período teve o seguinte desempenho.

```
'The following are analysis results of benchmark return(1day).'
                        risk
mean               -0.000029
std                 0.012439
annualized_return  -0.006810
information_ratio  -0.035490
max_drawdown       -0.244532
```

Figura 4.20: Benchmark - IBOV de 2021 a 2022

Para a análise do retorno com um benchmark negativo, como o registrado de -0.68%, deve-se inverter o sinal do componente do retorno do portfólio da 4.1. Um exemplo seria, que um retorno de +0.1 representaria uma variação de 0.068%, trazendo o retorno total para -0.612%. O MDD do período foi de -24.45%, bem menor que o observado na parte II do primeiro experimento.

Para a realização da comparação, nessa primeira parte, os dados foram testados no período "pós-pandemia", com treino e validação até 2019, já na segunda parte, foram treinados e validados incluindo o ao de 2020.

4.4.1 LightGBM

No LightGBM, já foi possível avaliar que os modelos foram impactados pelas mudanças realizadas no dataset.

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean               -0.000233
std                 0.006747
annualized_return  -0.055392
information_ratio  -0.532142
max_drawdown       -0.143198
```

Figura 4.21: LightGBM com treino e validação do IBOV até 2019

Na primeira simulação desse cenário, teve-se o retorno anualizado negativo, de -0.055, fazendo com que o retorno do portfólio fosse de -0.72%, com variação de 0.04%. O MDD foi de -27.94%, demonstrando uma variação de -3.48%.

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean                -0.000164
std                  0.008828
annualized_return   -0.039147
information_ratio   -0.287436
max_drawdown        -0.139818
```

Figura 4.22: LightGBM com treino e validação da B3 até 2019

No Experimento 1, a maioria das simulações teve um resultado melhor quando simulado com os dados da B3, o que também se repetiu nesse primeiro modelo, com desempenho de -0.039, totalizando um retorno de -0.71%, com variação de 0.03%. O MDD teve uma leve melhora, porém ficando bem próximo do valor anterior com -27.86%.

4.4.2 CatBoost

Foram avaliados os resultados com o modelo CatBoost.

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean                -0.000070
std                  0.007057
annualized_return   -0.016552
information_ratio   -0.152034
max_drawdown        -0.161617
```

Figura 4.23: CatBoost com treino e validação do IBOV até 2019

O CatBoost com os dados do IBOV ficou próximo do benchmark, com um retorno anualizado de -0.69%, uma diferença de somente -0.1%. O MDD foi de -0.1616, totalizando -28.40%, variando -3.95%, próximo ao LightGBM.

Confere-se os resultados com a B3.

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean                 0.000162
std                  0.015497
annualized_return    0.038587
information_ratio    0.161406
max_drawdown        -0.428351
```

Figura 4.24: CatBoost com treino e validação da B3 até 2019

Com os dados da B3 como um todo, o CatBoost teve um bom desempenho, ficando com um retorno bem próximo dos últimos resultados, com +0.0385, o que representa um retorno de -0.654%, um ganho de 0.026% comparado com o IBOV. O MDD foi maior com -0.4283, o que totaliza -34.92%, uma variação de consideráveis -10.47%.

O CatBoost, mais uma vez, mostra uma consistência muito interessante e apresenta bons desempenhos para as duas situações.

4.4.3 XGBoost

Expõe-se os resultado do XGBoost, que no experimento 1 teve os melhores desempenhos.

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean                -0.000104
std                  0.007210
annualized_return   -0.024663
information_ratio   -0.221746
max_drawdown        -0.164730
```

Figura 4.25: XGBoost com treino e validação do IBOV até 2019

Na primeira simulação do experimento 2, o XGBoost apresentou um retorno de -0.0246, com retorno de -0.7%, uma variação de -0.016%. Já o MDD teve valor de -28.47, bem próximo ao MDD do CatBoost, na mesma situação.

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean                -0.000210
std                  0.012643
annualized_return   -0.049995
information_ratio   -0.256318
max_drawdown        -0.215641
```

Figura 4.26: XGBoost com treino e validação da B3 até 2019

Com os dados da B3, o XGBoost fugiu um pouco do que era observado em outras simulações, o resultado piorou, com retorno em -0.71%, uma variação de 0.033%. O MDD foi de -29.72%, o que representa uma queda de -5.27% em relação ao benchmark, no pior momento.

O XGBoost foi bem afetado pela mudança dos dados de treinamento como observado por esses resultados, tendo alguns dos piores resultados entre os 3 primeiros modelos.

4.4.4 Ensemble

A seguir, verifica-se como foi o desempenho do modelo com método de ensemble, no cenário proposto.

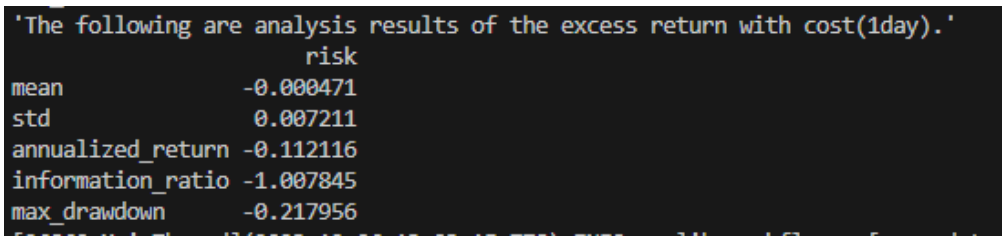

Figura 4.27: Ensemble com treino e validação do IBOV até 2019

Esse modelo teve o pior desempenho dessa primeira parte, com um retorno anualizado de -0,1121, o que resulta em um retorno de -0.75%, variando -0.076%, O MDD teve valor de -29.77%, o que representa uma diferença de -5.33%.

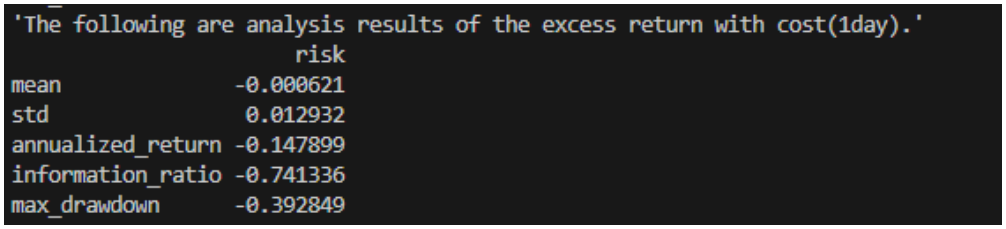

Figura 4.28: Ensemble com treino e validação da B3 até 2019

No segundo dataset, o modelo também fugiu da tendência observada de melhorar o desempenho, com um piora de -0.1%, em relação benchmark, trazendo o retorno para o pior resultado da parte I de -0.78%. O MDD foi o segundo pior desse cenário, com -0.3928, o que resultou em um total de -34.05%, variando -9.60%.

4.4.5 Compilado de Resultados Experimento 2 - Parte I

A Tabela 3 compara os resultados das simulações da primeira parte do Experimento 2.

Tabela 4.3: Compilado de Resultados para dados de Treino e Validação até 2019

Modelo	Fonte de Dados	Retorno Total (%)	Δ de Retorno (%)	MDD (%)	Δ de MDD (%)
LightGBM	IBOV	-0,72	-0,04	-27,94	-3,49
	B3	-0,71	-0,03	-27,86	-3,41
CatBoost	IBOV	-0,69	-0,01	-28,40	-3,95
	B3	-0,65	0,03	-34,92	-10,47
XGBoost	IBOV	-0,70	-0,02	-28,47	-4,02
	B3	-0,71	-0,03	-29,72	-5,27
Ensemble	IBOV	-0,75	-0,07	-29,77	-5,32
	B3	-0,78	-0,10	-34,05	-9,60

Vale destacar que o retorno do benchmark foi bem pequeno, nesses casos as variações e impactos dos modelos não parecem tão relevantes, porém devem ser avaliadas proporcio-

nalmente. O CatBoost manteve a consistência observada no primeiro experimento, obtendo dessa vez os melhores resultados com o dois blocos de dados.

O Ensemble foi o que teve o pior resultado nessa situação, além de um desempenho não muito interessante do XGBoost, dois algoritmos que tinham sido beneficiados pela mudança de dataset para B3 no experimento 1 e nesse tiveram uma piora no desempenho.

4.5 Experimento 2 - Parte II

O objetivo dessa segunda parte é verificar o desempenho dos modelos ao se adicionar o período da pandemia nos dados de validação, sendo assim, possível compará-los com os resultados da parte I.

O resultado do benchmark para essa segunda parte foi o mesmo da parte I, devido a manutenção do período testado.

```
'The following are analysis results of benchmark return(1day).'
                        risk
mean               -0.000029
std                 0.012439
annualized_return  -0.006810
information_ratio  -0.035490
max_drawdown       -0.244532
```

Figura 4.29: Benchmark - IBOV de 2021 a 2022

4.5.1 LightGBM

O LightGBM teve os seguintes resultados, considerando o ano de 2020 nos dados de treino e validação.

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean               -0.000300
std                 0.007182
annualized_return  -0.071289
information_ratio  -0.643400
max_drawdown       -0.143369
```

Figura 4.30: LightGBM com treino e validação do IBOV até 2020

Já na primeira simulação dessa parte, foram obtidos resultados menores dos que observados nos outros testes. O retorno do modelo foi de -0.73%, com variação de -0.04%. O MDD foi de -0.1433, totalizando -27.95%, com um queda comparativa de 3.5%.

Com os dados do IBOV, não se obteve um bom desempenho e a seguir pode-se conferir o desempenho com a B3 completa.

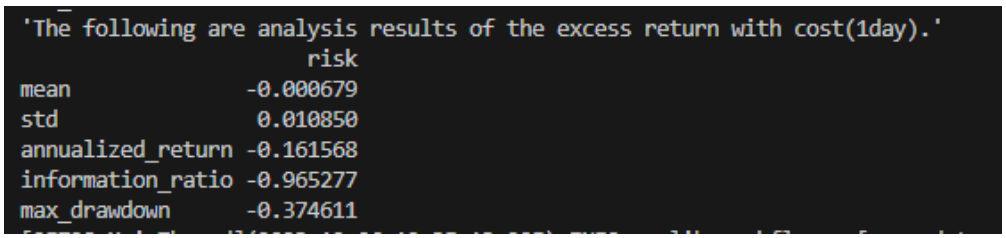

Figura 4.31: LightGBM com treino e validação da B3 até 2020

O desempenho com os dados da B3 foram ainda pior, gerando um retorno de -0.79% e uma variação de 0.11%, valores menores que os observados em qualquer simulação da 1ª parte. O MDD foi de -33.60%, caindo -9.15% do benchmark.

4.5.2 CatBoost

O primeiro modelo não obteve bons resultados, assim avaliou-se se o CatBoost conseguiu manter a consistência observada no experimento 1 e na parte anterior.

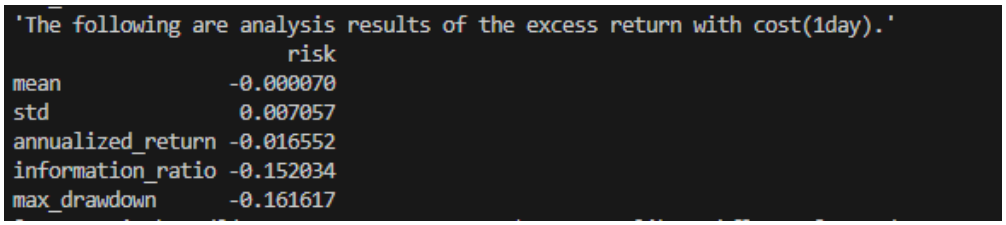

Figura 4.32: CatBoost com treino e validação do IBOV até 2020

Na simulação com o IBOV, tivemos um retorno anualizado de -0.69%, uma diferença de -0.1%. O MDD foi de -0.1616, totalizando -28.40%, variando -3.95%.

É possível notar que esses resultados são os mesmos da simulação com treino e validação até 2019, mostrando que 2020 não teve impacto no treinamento desse modelo com o IBOV.

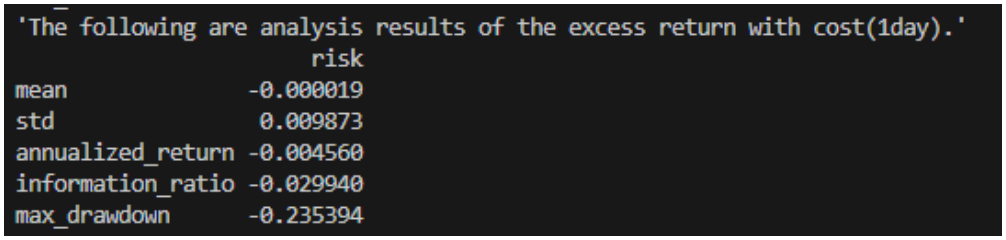

Figura 4.33: CatBoost com treino e validação da B3 até 2020

O retorno com os dados mais abrangentes foi de -0.683%, uma piora de 0.003%, muito próximo ao benchmark. Já o MDD teve piora de -5.75%, chegando a -30.20%.

4.5.3 XGBoost

Demonstra-se os resultados do modelo XGBoost.

```
'The following are analysis results of the excess return with cost(1day).'
                      risk
mean              -0.000193
std                0.007076
annualized_return -0.046035
information_ratio -0.421729
max_drawdown      -0.189348
```

Figura 4.34: XGBoost com treino e validação do IBOV até 2020

Com os dados do IBOV, o modelo XGBoost apresentou resultado de 0.71%, uma variação de -0.03%. O MDD teve queda comparativa de 0.1893, o que representa -29.08%, uma variação de -4.63%.

A seguir, são exibidos os resultados com todos os dados da B3.

```
'The following are analysis results of the excess return with cost(1day).'
                      risk
mean              -0.000601
std                0.010233
annualized_return -0.143107
information_ratio -0.906533
max_drawdown      -0.345528
```

Figura 4.35: XGBoost com treino e validação da B3 até 2020

O XGBoost, teve o pior desempenho, quando comparado as outras simulações do modelo nesses experimentos, com -0.1431, que representa um desempenho de -0.78%, caindo -0.097%. O MDD também teve um resultado ruim, com -0.3455, ou seja, - 32.89%.

4.5.4 Ensemble

O modelo usando o método de ensemble teve o pior resultado na parte 1, verificou-se o resultado para o novo cenário.

```
'The following are analysis results of the excess return with cost(1day).'
                      risk
mean              -0.000096
std                0.007057
annualized_return -0.022956
information_ratio -0.210845
max_drawdown      -0.137859
```

Figura 4.36: Ensemble com treino e validação do IBOV até 2020

Com o IBOV, o ensemble apresentou uma melhora, em relação aos resultados da parte I, com retorno de -0.69% e variação de -0.015%. O MDD teve um discreta queda de -0.1378,

que representa -27.81% de queda no pior momento do período testado, com diferencial de -3.36% do benchmark.

```
'The following are analysis results of the excess return with cost(1day).'
                        risk
mean               -0.000302
std                 0.012038
annualized_return  -0.071864
information_ratio  -0.386970
max_drawdown       -0.353260
```

Figura 4.37: Ensemble com treino e validação da B3 até 2020

Com os dados da B3, o ensemble piorou um pouco o rendimento, tendência também observada na parte I. O retorno foi de -0.73%, menor -0.05.% que o benchmark. O MDD foi de -33.08%, caindo -8.63% do valor do benchmark.

4.5.5 Resultado compilados Experimento 2 - Parte II

A tabela a seguir compila os resultados da 2ª parte do experimento 2.

Tabela 4.4: Compilado de Resultados para dados de Treino e Validação até 2020

Modelo	Fonte de Dados	Retorno Total (%)	Δ de Retorno (%)	MDD (%)	Δ de MDD (%)
LightGBM	IBOV	-0,73	-0,05	-27,95	-3,50
	B3	-0,79	-0,11	-33,60	-9,15
CatBoost	IBOV	-0,69	-0,01	-28,40	-3,95
	B3	-0,68	0,00	-30,20	-5,75
XGBoost	IBOV	-0,71	-0,03	-29,08	-4,63
	B3	-0,78	-0,10	-32,89	-8,44
Ensemble	IBOV	-0,69	-0,01	-27,81	-3,36
	B3	-0,73	-0,05	-33,08	-8,63

Pode-se observar que o resultado dos modelos piorou de forma geral. O CatBoost teve o melhor desempenho dos modelos novamente, mantendo um boa consistência. O XGBoost não teve um bom desempenho, assim como o LightGBM. Um ponto a se destacar é em relação ao método de ensemble, que teve uma melhora com os dados da pandemia inclusos no treino e validação.

Uma tendência observada na parte II, foi que com os dados da pandemia, os resultados foram piores no dataset da B3, indo na contramão do observado em outras simulações. Uma hipótese para esse comportamento, pode ser que com dados mais corrompidos, os modelos pioraram a avaliação com empresas com menos liquidez e de tamanho diversos, não só as maiores que o IBOV representa.

4.5.6 Conclusões do Experimento 2

No experimento II foi demonstrado o impacto dos dados de 2020, quando incluídos no dataset de treino e validação. Notou-se que ao não incluir os dados, na parte I, os desempenhos não foram tão interessantes como no experimento I e isso se deve ao período de teste. Apesar de se excluir o ano com maiores impactos para os dados, o sistema é composto de dados dependentes, logo, o outlier de 2020 impacta o desempenho dos papéis em períodos subsequentes e não podem ser tratados como eventos independentes. Além disso, um ponto relevante é o próprio momento do mercado avaliado, que constava com muita incerteza e desempenhos ruins, vide resultado negativo do benchmark no período.

Avaliado esse ponto, foi observado o desempenho da parte II do experimento, que mostrou uma piora nos desempenhos, de modo geral. De fato, acrescentar a pandemia nos datasets não ajudou os modelos, mostrando que o impacto de dados ruins nos datatsets de treino e validação podem impactar substancialmente até bom modelos, como o CatBoost.

4.5.7 Conclusões Gerais

Realizados os experimentos, foi visto que os dados utilizados em um modelo de machine learning são extremamente importantes para um bom desempenho, desde de datasets com boa qualidade, até o processamento desses datasets, como os que foram realizados com os dados da bolsa brasileira para melhorar os desempenhos.

Foi notado que os dois experimentos apresentaram uma queda considerável de rendimento com a inclusão do ano de 2020 nos datasets, seja como teste ou treino e validação. Com isso, mostrou-se interessante comparar os desempenhos dos modelos entre as partes de cada experimento, a fim de avaliar qual modo teve o maior impacto nos resultados, adicionar os dados no dataset de teste ou adicionar os dados no dataset de treino e validação. Para isso, foi calculado a diferença entre os desempenhos da primeira parte e da segunda dos dois experimentos, utilizando a seguinte fórmula.

$$Var = 100 * (Parte_2 - Parte_1)/abs(Parte_1) \tag{4.7}$$

Essa equação resulta na porcentagem de piora ou melhora entre parte 1 e 2, fazendo a divisão da diferença pelo módulo da referência, que no caso foram as partes I.

Avalia-se o quanto variou esse desempenho entre as partes dos experimentos, analisando a tabela abaixo.

Tabela 4.5: Análise da piora no desempenho das partes

Modelo	Fonte de Dados	Experimento 1	Experimento 2	Dataset com Maior Piora
LightGBM	IBOV	-55,55	-28,09	**Teste**
	B3	-51,79	-315,38	**Treino**
CatBoost	IBOV	-7,69	-6,25	**Teste**
	B3	-2,70	-113,15	**Treino**
XGBoost	IBOV	-95,31	-91,66	**Teste**
	B3	-29,60	-186,00	**Treino**
Ensemble	IBOV	1500,00	79,46	**Treino**
	B3	-82,45	51,35	**Teste**

Observou-se que houve um comportamento similar entre os 3 primeiros algoritmos simulados, o LightGBM, CatBoost e o XGBoost, apresentando-se uma mudança da tendência no Ensemble, que foi muito melhor na parte 2 do 1º experimento com o IBOV e, também, foi melhor com os dois dados no 2º experimento.

Pode-se analisar com os resultados, que a depender do dataset utilizado, sendo com ações do IBOV ou com ações da B3 inteira, o impacto foi maior em um dos testes. Verifica-se que com os dados do IBOV, o impacto no resultado, foi maior no experimento que se alterava o dataset de teste, já com os dados da B3, o impacto foi maior com as alterações nos datasets de treino e validação. A exceção a essa tendência observada, foi novamente no modelo que utilizou o método de ensemble. Uma possível explicação para essa tendência pode ser a diversidade dos dados da B3, que quando os modelos são treinados e validados com eles, o impacto é muito relevante, já que tem mais papéis, que representam empresas pequenas/médias. Assim, com um dataset de empresas mais consolidadas ou maiores, como o IBOV, o impacto não é tão grande, mostrando que nesses casos, um dataset diversificado com a presença de um outlier impacta muito mais um modelo quando o outlier está nos dados de treino e validação.

Por fim, foi possível identificar a força e grandes possibilidades de sucesso que essa modalidade de investimentos com inteligência artificial detém e como é importante se preocupar com os dados para trabalhos na área.

4.5.8 Tutorial

Uma segunda parte, trabalhada nesse projeto, foi a montagem de um pequeno tutorial com informações importantes para a utilização da biblioteca. A biblioteca conta com uma boa documentação, porém com poucos tutoriais e explicações mais diretas.

Primeiro, para o download da biblioteca, só seguir as instruções da documentação, essa parte é bem direto ao ponto. Contudo, o download pode ser feito por "pip install"com "pipi install pyqlib". Um ponto de observação nesse download é a compatibilidade dos pacotes, a

qlib foi feita para trabalhar com algumas versões do python e as mais atuais podem não estar habilitadas. O mesmo vale para outras bibliotecas como numpy, então é necessário ter muito cuidado com as versões dos arquivos.

Com a biblioteca baixada, passa-se para a estruturação dos modelos. É necessário realizar um script para baixar os dados, a documentação é bem detalhada para os dados chineses e americanos. É preciso ter muito cuidado da hora de definir a primeira parte do código, checando se os dados estão no caminho intencionado e se esse caminho está descrito em "provider_uri". Outro cuidados se mostra na definição do benchmark e market, sendo que os benchmarks devem ser incluídos com um ântes, por exemplo, "b̂vsp". Para checar os benchmarcks disponíveis nos dados, basta procurar a pasta de downloads dos dados e verificar na pasta de "features"quais estão marcados com ˆ.

A execução do modelo envolve vários processamentos, que podem ser induzidos a uma execução em paralelo, e com isso, um erro de multiprocessing pode aparecer. Para resolver esse possível problema, deve-se incluir o código dentro da condicional "if __name__ == "__main__":'.

Para a função de "data_handler", deve-se definir os períodos analisados, além das classes dos processadores, muita atenção nessa parte de processadores, porque a depender dos dataset, esse processamento pode fazer muita diferença no desempenho dos modelos.

Para a definição do modelo, nota-se a importância do cuidado com a definição dos hiperparâmetros e sua otimização, seja por meio de simulações e ajustes ou por softwares especializados. Um ponto a se destacar nessa definição é o "module_path", é preciso definir o caminho correto de acordo com a organização da biblioteca, então importante verificar a estrutura da mesma. Por exemplo, "qlib.contrib.model.xgboost", significa que o xgboost está na pasta model, dentro da pasta contrib, que por sua vez se encontra na pasta qlib.

Na parte de task deve-se atentar-se ao "module_path"novamente e principalmente a divisão do dataset. O backtest tem pontos de atenção parecidos, com a os "module_path"do executor e da estratégia e a definição do período de teste. Além disso, é nessa seção que são definidos os custos das operações e a frequência de avaliação.

A parte final, consiste na execução do modelo e na geração de métricas de avaliação. A documentação da biblioteca é bem rica na parte de avaliação do portfólio.

4.5.9 Recomendações para Pesquisas Futuras

Para futuros trabalhos, recomenda-se a utilização de outros tipos de avaliações complementares. Como explicado na Introdução, essa modelagem contém limitações e com isso, podem ser tomados caminhos para a melhora do desempenho e da aplicação no mundo real, como a utilização de indicadores econômicos que influenciam movimentos no mercado, indicadores técnicos das empresas, além de análises de notícias para computar como a componente externa de avaliação. Por fim, entende-se como um excelente caminho essa avaliação

de notícias aliada a uma análise intrínseca dos papéis, sendo possível gerar um modelo preditivo bem robusto.

Capítulo 5

Conclusão

Com esse trabalho foi possível avaliar a importância dos dados na modelagem de soluções com inteligência artificial, tanto para treino, quanto para testes. Tendo como claras as limitações dos modelos, viu-se que o desempenho da adição de outliers nos datasets foram bem impactantes pros desempenhos, porém mais ainda dependendo das características do dataset utilizado. Quando o dataset foi mais diverso, com variações maiores, a adição do outlier nos dados de treino e validação foi pior, mas com dados um pouco mais padronizados, o impacto na simulação com o dataset de teste foi pior. Isso foi uma tendência observada nos treinamentos, com algumas exceções, principalmente com o modelo de ensemble. Contudo, uma constatação é que para esse tipo de problema, regressão voltada para o mercado de renda variável, os métodos de boosting com árvores de decisão tem, em geral, um bom desempenho.

Por fim, espera-se que esse seja um campo com cada vez mais pesquisas, visto que o potencial de benefícios é muito grande.

Capítulo 6

Referências Bibliográficas

1. LAURETTO, Marcelo. Árvores de Decisão.https://edisciplinas.usp.br/pluginfile.php /4469825/mod_resource/content/1/ArvoresDecisao_normalsize.pdf. Novembro, 2010.
2. SPOLADOR, Rodolfo. Aplicação do método de Gradient Boosting. https://app.uff.br/riuff/bitstream/handle/1/25319/tcc_20202_RodolofoHauret_117054008 .pdf?sequence=1isAllowed=y. Maio, 2021.
3. GAMA, João. Árvores de Decisão. https://www.dcc.fc.up.pt/ ines/aulas/0910/MIM/aulas/arvores_de_decisao.pdf. 2002.
4. SCHAPIRE, Robert e FREUND, Yoav. Boosting Foundations and Algorithms. https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Boosting_% 20Foundations%20and%20Algorithms%20%5BSchapire%20%26%20Freund%202012-05-18%5D.pdf.
5. Qlib Documentation. https://qlib.readthedocs.io/en/latest/index.html
6. CatBoost Documentation. https://catboost.ai/en/docs/features/training
7. XGBoost Documentation. https://xgboost.readthedocs.io/en/stable/tutorials/index.html
8. Apostila Certificação AAI - Topinvest. Revisada Julho/2023.
9. https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2023-generative-ais-breakout-year/pt-BR
10. https://forbes.com.br/forbes-tech/2023/02/chatgpt-tem-recorde-de-crescimento-da-base-de-usuarios/
11. https://www.b3.com.br/pt_br/noticias/numero-de-investidores-na-b3-cresce-34-em-renda-fixa-e-23-em-renda-variavel-em-12-meses.html
12. https://www.ibm.com/br-pt/topics/machine-learning

13. https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-machine-learning/

14. https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/

15. https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/

16. https://sites.icmc.usp.br/andre/research/neural/

17. https://brilliant.org/wiki/backpropagation/text=Backpropagation

18. https://www.btgpactualdigital.com/como-investir/artigos/investimentos/tudo-sobre-bolsa-de-valores

19. https://www.b3.com.br/noticias/numero-de-investidores-na-b3-cresce-mesmo-em-cenario-de-alta-volatilidade.htm

20. https://professor.pucgoias.edu.br/SiteDocente/admin/arquivosUpload/3843/material/RISCO20E20INCERTEZA2052017gabAULAS.pdf

21. https://www.suno.com.br/artigos/benchmark/

22. https://www.seudinheiro.com/2020/bolsa-dolar/fechamento-ibovespa-dolar12-03/