

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Inteligência Artificial e Aprendizado de Máquina

Wagner A. Monzyne

APRENDIZAGEM DE MÁQUINA PARA
PREVISÃO DE PREÇO MÁXIMO E MÍNIMO DE UMA AÇÃO
DURANTE PREGÃO DA BOLSA DE VALORES

Irecê-BA
Setembro de 2022

Wagner A. Monzyne

**APRENDIZAGEM DE MÁQUINA PARA
PREVISÃO DE PREÇO MÁXIMO E MÍNIMO DE UMA AÇÃO
DURANTE PREGÃO DA BOLSA DE VALORES**

Trabalho de Conclusão de Curso apresentado
ao Curso de Especialização em Inteligência
Artificial e Aprendizado de Máquina, como
requisito parcial à obtenção do título de
Especialista.

Irecê-BA
Setembro de 2022

SUMÁRIO

1. Introdução	4
2. Descrição do Problema e da Solução Proposta	4
3. Canvas Analítico.....	6
4. Coleta de Dados	7
4.1. Histórico de Cotações	7
4.1. Eventos de Desdobramento de Ações	8
5. Processamento/Tratamento de Dados	9
5.1. Importação dos dados da B3.....	9
5.2. Seleção e Ajuste das Séries Históricas	10
5.3. Resultado do Processamento	11
6. Análise e Exploração dos Dados	12
6.1. Análise Univariada	13
6.2. Análise Bivariada/Multivariada	18
7. Preparação dos Dados para os Modelos de Aprendizado de Máquina	23
8. Aplicação de Modelos de Aprendizado de Máquina	26
8.1. Erro Percentual Absoluto Médio (MAPE)	26
8.2. Modelo de Regressão Linear	27
8.3. Modelo Floresta Randômica	27
8.4. Modelo Rede Neural Artificial	28
9. Discussão dos Resultados	30
9.1. Pipeline completo até a Faixa de Negociação.....	31
10. Conclusão	33
11. Links.....	34

1. Introdução

A definição do preço de uma ação de empresa listada em bolsa de valores sofre influências que vão além do desempenho financeiro e operacional dela própria. Fatores econômicos, políticos, climáticos e até emocionais, entre outros, podem levar a oscilações no julgamento do valor do ativo pelos investidores. Por conta da grande quantidade de variáveis influenciadoras, os preços, especialmente no curto prazo, apresentam um comportamento aleatório.

Neste cenário complexo, encontrar um padrão no comportamento dos preços que impliquem em movimentos futuros determinados poderia proporcionar uma vantagem para o operador reduzindo os riscos de seus investimentos. A Análise Técnica de Ações busca isto, ao considerar que os preços formados refletem todas as informações disponíveis aos investidores, e que estudando e interpretando o histórico dos preços passados é possível antecipar movimentos futuros.

No campo da Inteligência Artificial, a busca por padrões e regras a partir de grandes quantidades de informações e dados é objetivo da Aprendizagem de Máquina. Neste trabalho, serão utilizados modelos de aprendizagem para processar dados históricos de preços de ações para avaliar sua eficácia na predição de movimentos futuros, sem desconsiderar a natureza complexa e aleatória da formação dos preços, mas mantendo a premissa de que o estudo do passado pode gerar regras para comportamentos futuros que podem vir a ser úteis para redução de risco de investimentos no mercado de ações.

2. Descrição do Problema e da Solução Proposta

Durante um dia típico de pregão na bolsa de valores, os preços de um ativo podem oscilar bastante. Para investidores que negociam papéis dentro do mesmo dia (daytrade), ou seja, iniciam posições compradas ou vendidas e as encerra antes do fechamento do dia, conhecer até que ponto o preço da ação ainda tem espaço para aumentar ou para diminuir pode ser útil para tomada de decisão sobre a viabilidade da operação, reduzindo riscos e aumentando a probabilidade de obter lucro.

O problema a ser solucionado é a previsão dessa faixa de preços onde é esperado que se contenha a oscilação. Para isto, serão analisados históricos de cotações de

uma das principais empresas listadas na Bolsa brasileira, a Petróleo Brasileiro S.A. (PETR4 – Petrobrás), aplicando Modelos de Aprendizagem de Máquina para tarefa de Regressão, a fim de prever o **Preço Máximo** e o **Preço Mínimo** que devem ser alcançados durante o dia de pregão já iniciado.








Foram escolhidos três algoritmos que suportam esta tarefa: Regressão Linear (RL), Floresta Randômica (FR) e a Rede Neural Artificial (RNA), todos com implementações disponíveis na biblioteca Scikit-Learn, para linguagem Python. O indicador de desempenho dos modelos será o Erro Médio Percentual Absoluto (MAPE – Mean Absolute Percentage Error) por conta do tipo de tarefa e porque irá trabalhar com séries temporais com grandes diferenças no nível de preços ao longo dos muitos anos.

O resultado esperado é ter um modelo treinado capaz de prever o preço máximo e o mínimo, dentro de uma margem de erro percentual aceitável, para que um operador disponha, logo no início do pregão, de uma faixa de preços para utilizar como referência em suas decisões de investimentos.

3. Canvas Analítico

Software Analytics Canvas

Project: Definição de Faixa de Negociação por Inteligência Artificial

<p> 1. Question</p> <p><i>What is it that we want to know about the software / processes / usage / organization / etc.?</i></p> <p>Uma Inteligência Artificial pode prever uma faixa de preço máximo e mínimo que em que uma ação irá oscilar durante um pregão da Bolsa de Valores?</p>	<p> 2. Data Sources</p> <p><i>Which data can possibly answer our question? What information do we need?</i></p> <p>Será utilizado o histórico diário de cotações disponibilizado ao público pela principal Bolsa brasileira, a B3, em seu site. Esta base será processada, convertida e ajustada.</p>	<p> 3. Heuristics</p> <p><i>Which assumptions do we want to make to simplify the answer to our question?</i></p> <p>É assumido que, embora a definição do preço de um ativo sofra interferências de vários fatores não expressos na base de dados como desempenho da empresa, fatores políticos, econômicos e até emocionais, seja possível prever, com alguma precisão, uma faixa em que os preços poderão oscilar no dia do pregão com base nos movimentos passados utilizando modelos de Aprendizagem de Máquina para regressão e projeção de um preço máximo e um preço mínimo para o dia.</p>	<p> 4. Validation</p> <p><i>What results do we expect from our analysis, how are they reviewed and presented in an understandable way?</i></p> <p>O modelo treinado deve atingir um Erro Percentual Absoluto Médio (MAPE) compatível com a volatilidade do ativo de modo que possa ser útil para reduzir o risco de operações daytrade.</p>
<p> 5. Implementation</p> <p><i>How can we implement the analysis step by step and in a comprehensible way?</i></p> <ul style="list-style-type: none"> •Obter os arquivos com históricos de Cotações da B3; •Processar os arquivos da B3, consolidando e extraíndo os campos relevantes em forma de tabela; •Aplicar transformações específicas para Séries Temporais; •Construir Modelos e Treinar, aplicando validação específica para séries temporais; •Visualizar a taxa de erro de das predições; 		<p> 6. Results</p> <p><i>What are the main insights from our analysis?</i></p> <p>Os modelos de Aprendizado de Máquina são úteis no processamento de grandes quantidades de informações sobre os preços, muitas vezes com regras em quantidades que seriam desafiadoras para a interpretação humana. Por outro lado, o desempenho cai durante períodos de pouca racionalidade, como crises políticas, catástrofes e outros eventos mais emocionais. Ainda assim, demonstram ser mais eficazes do que métodos aleatórios, podendo ser úteis para auxiliar na tomada de decisão de investimentos ao serem utilizados em conjunto com outras metodologias de análise.</p>	<p> 7. Next Steps</p> <p><i>What follow-up actions can we derive from the findings? Who or what do we need to address next?</i></p> <ul style="list-style-type: none"> •Utilizar dados de ativos correlacionados, bem como outros indicadores econômicos ou políticos com potencial de influenciar no desempenho dos ativos; •Utilizar análise de sentimento de mercado a partir de publicações em redes sociais ou sites especializados; •Implementar um indicador para MetaTrader (linguagem mql5) que desenhe no gráfico de preços, em tempo real, os limites de preços previstos pelo modelo treinado;

4. Coleta de Dados

4.1. Histórico de Cotações

Para treinamento e validação dos modelos, são utilizadas séries históricas das cotações diárias de ativos negociados na principal bolsa de valores brasileira, a B3. As séries são disponibilizadas publicamente no site da instituição no endereço <http://www.b3.com.br>, em arquivos formato texto com layout específico, agrupados por ano, contendo o histórico de preços dos títulos negociados nesta Bolsa desde 1986.

Os arquivos estavam disponíveis em 3 de junho de 2022, através da URL https://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/historico/mercado-a-vista/series-historicas/, assim como a descrição do layout para interpretação destes dados através do link https://www.b3.com.br/data/files/33/67/B9/50/D84057102C784E47AC094EA8/SeriesHistoricas_Layout.pdf.

Estão disponíveis todas as negociações de ações realizadas no mercado nacional nas últimas décadas nesta Bolsa brasileira, mas é considerado na análise apenas o período compreendido entre 2001 a 2021, totalizando 21 arquivos: um para cada ano. Além disso, nem todos os atributos são utilizados, sendo extraídos apenas os listados nas definições abaixo:

Nome do dataset: COTAHIST.AAAA.TXT Descrição: Arquivo de histórico de cotações referente ao ano AAAA. Link: https://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/historico/mercado-a-vista/series-historicas/			
Nome do Atributo	Descrição	Tipo	Posição
DATA DO PREGÃO	Data em que ocorreu o pregão, em formato AAAAMMDD	Data	03-10
CODNEG	Código de negociação do ativo	Texto	13-24
NOMRES	Nome resumido da empresa emissora do papel	Texto	28-39

PREABE	Preço de abertura do papel no pregão	Numérico (centavos)	57-69
PREMAX	Preço máximo do papel no pregão	Numérico (centavos)	70-82
PREMIN	Preço mínimo do papel no pregão	Numérico (centavos)	83-95
PREULT	Preço do último negócio do papel no pregão	Numérico (centavos)	109-121
QUATOT	Quantidade total de títulos negociados neste papel no pregão	Numérico	153-170

4.1. Eventos de Desdobramento de Ações

A base de dados fornecida pela B3 não contempla informação sobre desdobramentos de ações, o que pode representar variação abrupta nos valores da série histórica para um ativo. Estes eventos ocorrem quando a empresa representada pelo papel decide reduzir o preço individual da ação ao multiplicar a quantidade de títulos disponíveis. Dessa forma, o preço é recalculado diminuindo na mesma proporção do aumento na sua quantidade, não gerando alteração de valor.

Estes eventos são divulgados em sites especializados, e para este trabalho foram coletados manualmente (apenas para os ativos analisados) no site Investing.com, acessível em 3 de junho de 2022 através da URL <https://br.investing.com/stock-split-calendar/>. Os dados foram agrupados em uma planilha, conforme definição a seguir:

Nome do dataset: DESDOBRAMENTOS.XLS

Descrição: Histórico de eventos de desdobramentos de Ações.

Link:

<https://docs.google.com/spreadsheets/d/1N9qivSrDTdjapXOOXs2GYe6Zi-LvIUQB/edit?usp=sharing&ouid=107214047519977145247&rtpof=true&sd=true>

Nome do Atributo	Descrição	Tipo
DATA DO PREGÃO	Data em que ocorreu o desdobramento, no formato AAAAMMDD	Data
CODNEG	Código de negociação do ativo	Texto
FATOR	Fator de ajuste a ser aplicado nas séries. Este fator divide os campos preços, e multiplica a quantidade de títulos negociados, para ajuste da série temporal.	Numérico

5. Processamento/Tratamento de Dados

Para processamento e análise dos dados será utilizada linguagem de programação Python em conjunto com bibliotecas especializadas e amplamente difundidas na área da Ciência de Dados, como a Pandas para manipulação de dados, a Matplotlib para visualização de gráficos e o Scikit-Learn para Aprendizagem de Máquina, que simplificam o trabalho com grandes volumes de dados.

5.1. Importação dos dados da B3

As análises serão realizadas a partir dos dados disponibilizados pela Bolsa B3, em arquivos de texto, com informações de cotações diárias, compactados, agrupados por ano. Os arquivos referentes aos anos de 2001 a 2021 devem ser previamente baixados para a pasta “cotacoes_b3”, de onde serão processados (obedecendo o layout fornecido) para extração dos atributos 'DATA DO PREGÃO', 'CODNEG', 'NOMRES', 'PREABE', 'PREMAX', 'PREMIN', 'PREULT' e 'QUATOT', especificados na

sessão 4.1, gerando uma tabela alocada em memória consolidada com o histórico diário das cotações dos diversos ativos em todo o período.

Após a importação, é feita a conversão dos tipos de dados referentes aos preços para ponto flutuante com duas casas decimais, e o campo referente à data do pregão é convertido para o tipo DateTime e definido como índice da tabela.

A código da função a seguir realiza esse processamento dos arquivos retornando os dados como um objeto do tipo DataFrame da biblioteca Pandas:

```

1 def importar_dados_b3():
2     '''Processa os arquivos da b3 importando dados para um DataFrame'''
3     df_historico = pd.DataFrame()
4     # Layout do arquivo fornecido pela b3
5     LAYOUT_B3 = dict(
6         columns = ['TIPREG', 'DATA DO PREGÃO', 'CODBDI', 'CODNEG', 'TPMERC', 'NOMRES',
7                   'ESPECI', 'PRAZOT', 'MODREF', 'PREABE', 'PREMAX', 'PREMIN', 'PREMED',
8                   'PREULT', 'PREOFC', 'PREOFV', 'TOTNEG', 'QUATOT', 'VOLTOT', 'PREEXE',
9                   'INDOPC', 'DATVEN', 'FATCOT', 'PTOEXE', 'CODISI', 'DISMES'],
10        widths = [2, 8, 2, 12, 3, 12, 10, 3, 4, 13, 13, 13,
11                 13, 13, 13, 13, 5, 18, 18, 13, 1, 8, 7, 7, 12, 9]
12    )
13    # itera entre os arquivos
14    for b3_zip in os.listdir(DADOS_BOLSA_DIR):
15        print('Processando arquivo: %s' % b3_zip)
16        # descompacta arquivo de dados...
17        zip_file = zipfile.ZipFile(os.path.join(DADOS_BOLSA_DIR, b3_zip), 'r')
18        b3_txt = zip_file.namelist()[0]
19        zip_file.extractall()
20        zip_file.close()
21        # converte o texto em um DataFrame intermediário
22        df_hist_ano = pd.read_fwf(b3_txt, widths=LAYOUT_B3['widths'], names=LAYOUT_B3['columns'],
23                                skipfooter=1, skiprows=1)
24        # mantém apenas as colunas de interesse
25        df_hist_ano = df_hist_ano[['DATA DO PREGÃO', 'CODNEG', 'NOMRES', 'PREABE', 'PREMAX',
26                                'PREMIN', 'PREULT', 'QUATOT']]
27        # concatena ano ao histórico consolidado
28        df_historico = df_historico.append(df_hist_ano)
29        # descarta arquivo texto descompactado
30        os.remove(b3_txt)
31        # converte tipos de dados
32        df_historico['DATA DO PREGÃO'] = pd.to_datetime(df_historico['DATA DO PREGÃO'], format='%Y%m%d')
33        for col in ['PREABE', 'PREMAX', 'PREMIN', 'PREULT']:
34            df_historico[col] = df_historico[col] / 100.0
35        # define a data como índice da tabela
36        df_historico.set_index('DATA DO PREGÃO', inplace=True)
37        df_historico.sort_index(inplace=True)
38        # - - -
39    return df_historico

```

O processo pode demorar vários minutos, e ao final terão sido importados mais de 9 milhões de registros. Os arquivos contemplam informações dos diversos tipos de mercados administrados pela B3, e de todos os ativos que os compõem.

5.2. Seleção e Ajuste das Séries Históricas

Serão analisados apenas os dados referentes às ações da Petrobrás (PETR4) negociadas no Mercado À Vista. Estas séries ainda precisam ser ajustadas aos

eventos de desdobramentos manualmente coletados e tabulados na planilha “Desdobramentos.xls”, conforme especificado na sessão 4.2.

O ajuste nas cotações devido aos eventos desdobramentos é necessário para preservar a relação da variação da série ao longo do tempo, do contrário, grandes variações provocariam descontinuidades gerando falsos sinais de mudanças na formação do preço e sentimento de mercado, quando na verdade tratam apenas de decisão administrativa que não altera o seu valor. É preciso ter em mente que o valor real praticado à época não é preservado, uma vez que o ajuste consiste em recalcular todas as cotações anteriores ao evento, e análises baseadas no valor nominal podem levar a falsas conclusões.

A função a seguir aplica um filtro sobre o histórico de cotações, importado anteriormente, selecionando apenas dados de um único ativo e em seguida faz os ajustes na série histórica com base nos eventos de desdobramentos, retornando um objeto do tipo DataFrame com dados da empresa:

```

1 def get_historico_empresa(codneg, df_historico, df_desdobramentos):
2     '''Filtra histórico de uma empresa e aplica ajustes por desdobramentos.'''
3     # seleciona sub-tabela
4     df_ativo = df_historico[df_historico['CODNEG'] == codneg].copy()
5     # - - -
6     # ajusta série histórica com os eventos de desdobramento
7     eventos_desdobramentos = df_desdobramentos[df_desdobramentos['CODNEG'] == codneg]
8     colunas_inversa = ['PREABE', 'PREMAX', 'PREMIN', 'PREULT']
9     colunas_direta = ['QUATOT']
10    for data_evento, fator in eventos_desdobramentos[['DATA DO PREGÃO', 'FATOR']].values:
11        idx_corrigir = df_historico_ativo.index < data_evento
12        df_ativo.loc[idx_corrigir, colunas_inversa] = df_ativo[idx_corrigir][colunas_inversa] / fator
13        df_ativo.loc[idx_corrigir, colunas_direta] = (df_ativo[idx_corrigir][colunas_direta] * fator).astype(int)
14    # - - -
15    return df_ativo

```

5.3. Resultado do Processamento

Ao final desta etapa de processamento, temos um dataset contendo séries históricas da empresa a ser analisada: o histórico da Petrobrás contém 5.197 registros, referentes às cotações diárias entre 2 de janeiro de 2001 à 30 de dezembro de 2021.

	CODNEG	NOMRES	PREABE	PREMAX	PREMIN	PREULT	QUATOT
DATA DO PREGÃO							
2001-01-02	PETR4	PETROBRAS	5.72500	5.97125	5.72500	5.88625	8130400
2001-01-03	PETR4	PETROBRAS	5.88750	6.21250	5.82500	6.20625	15184000
2001-01-04	PETR4	PETROBRAS	6.21625	6.26875	6.19625	6.25000	11534400
2001-01-05	PETR4	PETROBRAS	6.25000	6.38750	6.16250	6.19125	10341600
2001-01-08	PETR4	PETROBRAS	6.25000	6.28750	6.13750	6.17375	8992000
...
2021-12-23	PETR4	PETROBRAS	28.33000	28.42000	28.15000	28.33000	28713600
2021-12-27	PETR4	PETROBRAS	28.32000	28.78000	28.15000	28.75000	44227900
2021-12-28	PETR4	PETROBRAS	29.01000	29.28000	28.70000	28.78000	30688100
2021-12-29	PETR4	PETROBRAS	28.70000	28.90000	28.42000	28.54000	35508400
2021-12-30	PETR4	PETROBRAS	28.55000	28.70000	28.39000	28.45000	43229100

5197 rows x 7 columns

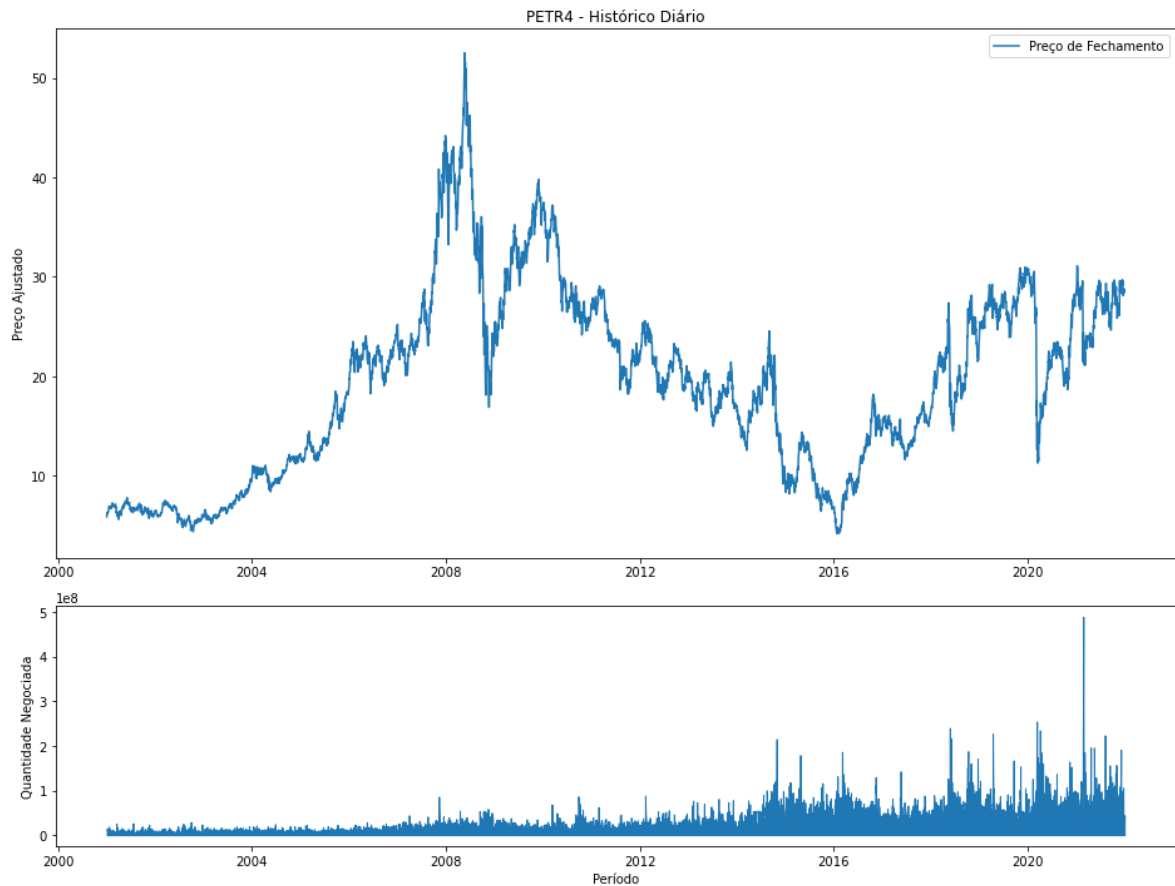
6. Análise e Exploração dos Dados

Os dados coletados são referentes basicamente a Preço de negociação e ao Volume negociado diariamente, ajustados em relação aos eventos de Desdobramentos de Ações ocorridos durante o período.

Em relação ao Preço, as variáveis PREABE (preço de abertura) e PREULT (preço de fechamento), são coletadas em momentos de grande participação de investidores, que são os leilões de Abertura (Call de Abertura) que definem os preços dos primeiros negócios do dia, e os leilões de Encerramento, definindo o preço dos últimos negócios do dia e que é considerado pelo Mercado como preço de referência. Já as variáveis PREMIN (preço mínimo) e PREMAX (preço máximo), são registrados em momentos indefinidos, e representam os maiores e menores preços que os investidores estavam dispostos a fechar negócios durante o pregão.

Em relação ao Volume, temos a variável QUATOT que expressa o interesse em negociar as ações da empresa através da quantidade de títulos transferidos entre os operadores durante o dia.

No gráfico a seguir, é representado o conjunto de dados históricos das cotações da empresa Petrobrás (PETR4):



Apesar da natureza aleatória do comportamento do mercado no curto prazo, verificada pelos serrilhados ao longo na linha do preço no gráfico, é possível verificar pelo menos três longas tendências no período analisado: altista entre 2003 à meados de 2008, baixista de 2010 a 2015 e novo comportamento altista de 2016 a 2021, que desenharam duas grandes inversões no sentimento e percepção dos investidores sobre o valor das ações da empresa, expressas pelas mudanças na direção da formação dos preços ocorridas volta de 2009 e outra no ano de 2015.

Outro ponto representado no gráfico é o crescimento da quantidade de títulos negociados ao longo do período, que demonstra que houve um aumento constante do interesse na participação no Mercado de ações pelos investidores para este ativo.

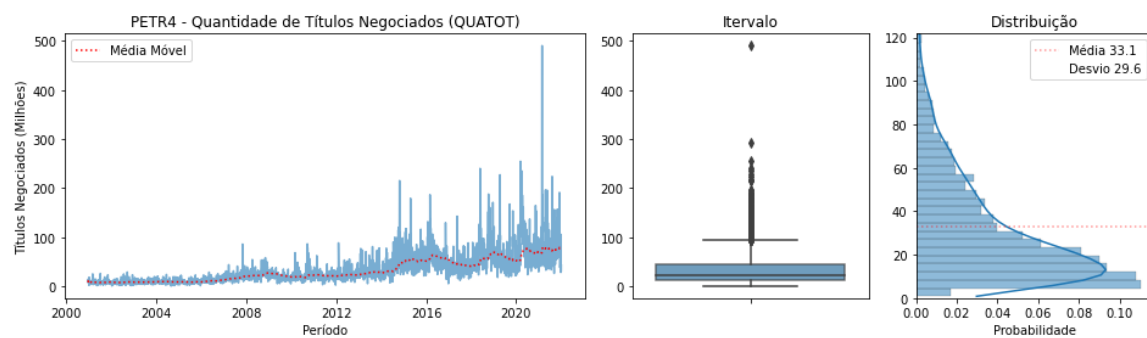
6.1. Análise Univariada

À seguir, são apresentadas algumas medidas de posição e dispersão dos dados para as variáveis disponíveis, bem como decomposições e transformações aplicáveis em

análise de séries temporais que ajudam a evidenciar padrões nas variações ao longo do tempo.

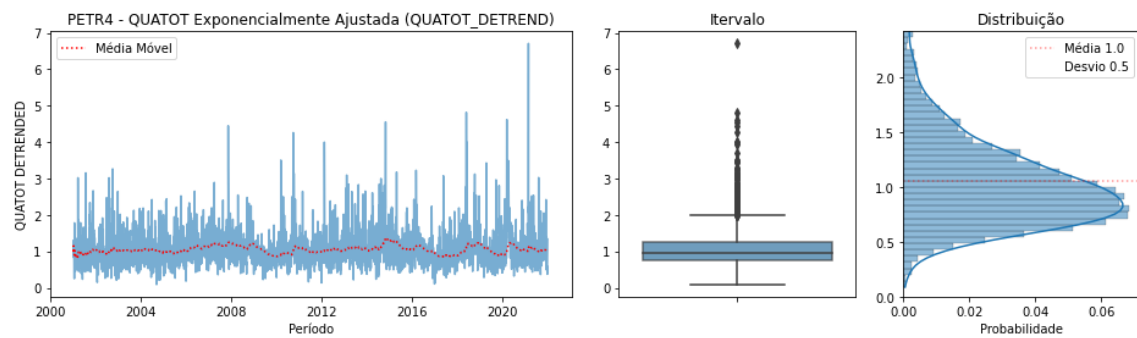
	PREABE	PREMAX	PREMIN	PREULT	QUATOT
Média	19,20	19,49	18,89	19,18	33,1 mi
Desvio Padrão	9,22	9,34	9,08	9,20	29,6 mi
Mediana	19,46	19,74	19,10	19,40	23,4 mi
Mínimo	4,20	4,27	4,12	4,20	0,8 mi
Máximo	52,58	53,68	51,95	52,51	490,23

6.1.1 Volume / Quantidade de Títulos Negociados (QUATOT)



- A distribuição dos valores se concentra em torno dos 20 milhões de títulos negociados
- O volume cresce durante todo o período, mas a partir de 2015 o crescimento é mais acentuado

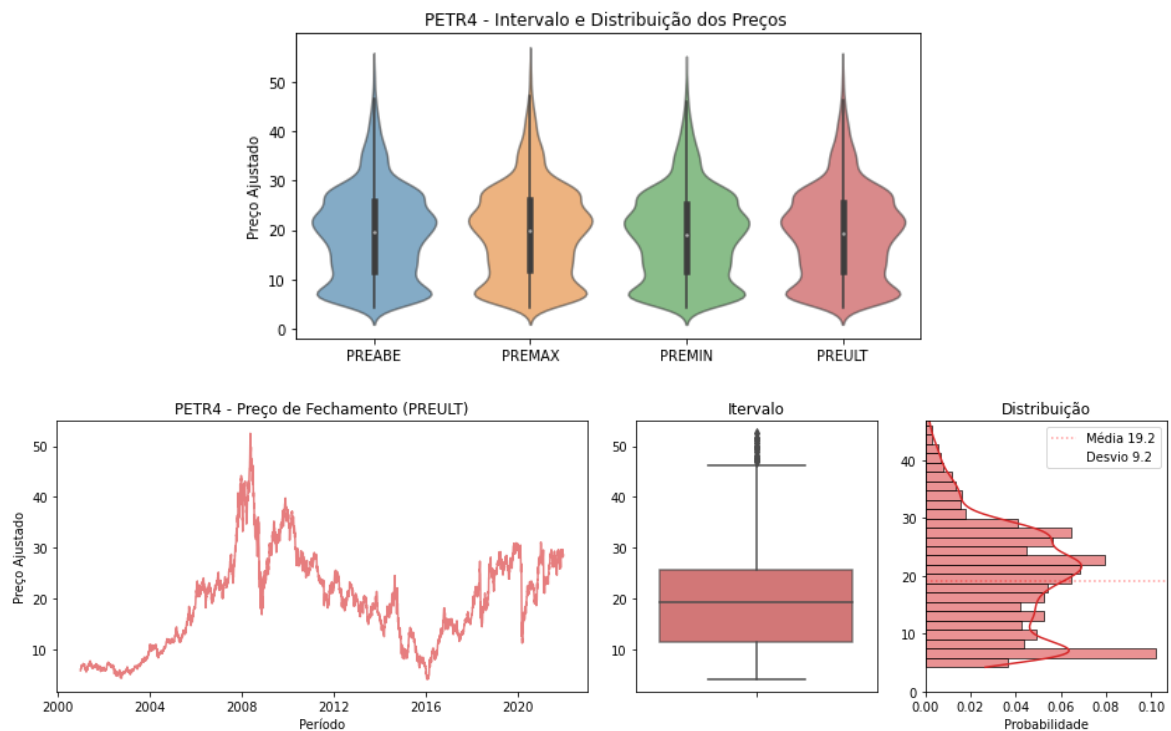
Para descontar os efeitos acumulados pela tendência e evidenciar as oscilações de curto prazo, é aplicado a decomposição da série ajustando a uma média exponencial. No modelo multiplicativo utilizado, é gerada uma nova série que oscila em torno de 1, com característica mais estacionária:



- Variação do volume homogênea durante todo o período
- Distribuição dos valores se assemelha à curva normal

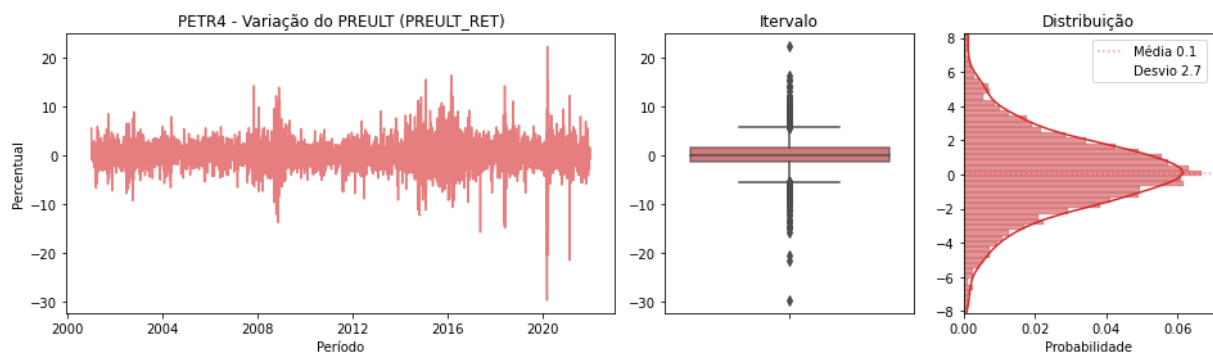
6.1.2 Preços (PREABE, PREMAX, PREMIN, PREULT)

As quatro variáveis relativas ao preço possuem estimadores muito semelhantes, apresentando coeficiente de variação em torno de 48%, desvio padrão em torno de R\$ 9,20, e média em torno de R\$ 19,20. Esta semelhança pode ser explicada pelo fato de serem coletas da mesma variável Preço no mesmo dia, apenas em momentos distintos: Preço de Abertura (PREABE), Preço Máximo (PREMAX), Preço Mínimo (PREMIN) e Preço de Fechamento (PREULT).

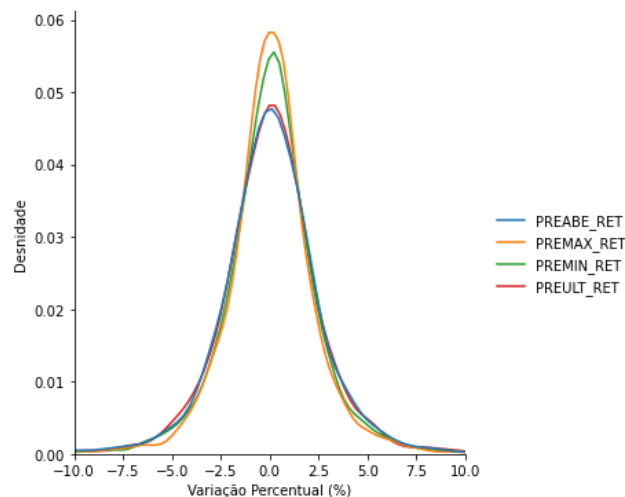


- Maior frequência da distribuição abaixo dos R\$ 30
- Pico isolado de frequência em torno de R\$ 6, verificado quase exclusivamente nos primeiros 4 anos da série

Para recuperar o comportamento da variação diária, são extraídas Séries de Retornos (estacionárias), descartando a tendência e oscilações de curto prazo. A série é obtida calculando o percentual de variação em relação ao registrado no dia anterior.



- A distribuição dos valores se concentra em torno de 0% de forma simétrica
- Mais de 99% das amostras oscilam entre +8% e -8%, aproximadamente
- Valores extremos atingem até 30% de variação



Graficamente, as quatro séries geradas (PREABE_RET, PREMAX_RET, PREMIN_RET e PREULT_RET) são muito semelhantes entre si. Principalmente o preço de abertura em relação ao preço de fechamento, e o preço máximo em relação ao preço mínimo. A tabela abaixo mostra algumas medidas estatísticas para elas.

	PREABE_RET	PREMAX_RET	PREMIN_RET	PREULT_RET
Média	0,07%	0,06%	0,06%	0,07%
Desvio Padrão	2,82pp	2,42pp	2,58pp	2,72pp
Mediana	0,06%	0,06%	0,12%	0,09%
Mínimo	-26,10%	-24,45%	-31,66%	-29,70%
Máximo	35,04%	20,35%	20,28%	22,22%
Intervalo a 3 desvios padrão da média	[-8,39%, 8,53%]	[-7,19%, 7,31%]	[-7,68%, 7,81%]	[-8,11%, 8,24%]

6.1.3 Variação Intradiária dos Preços

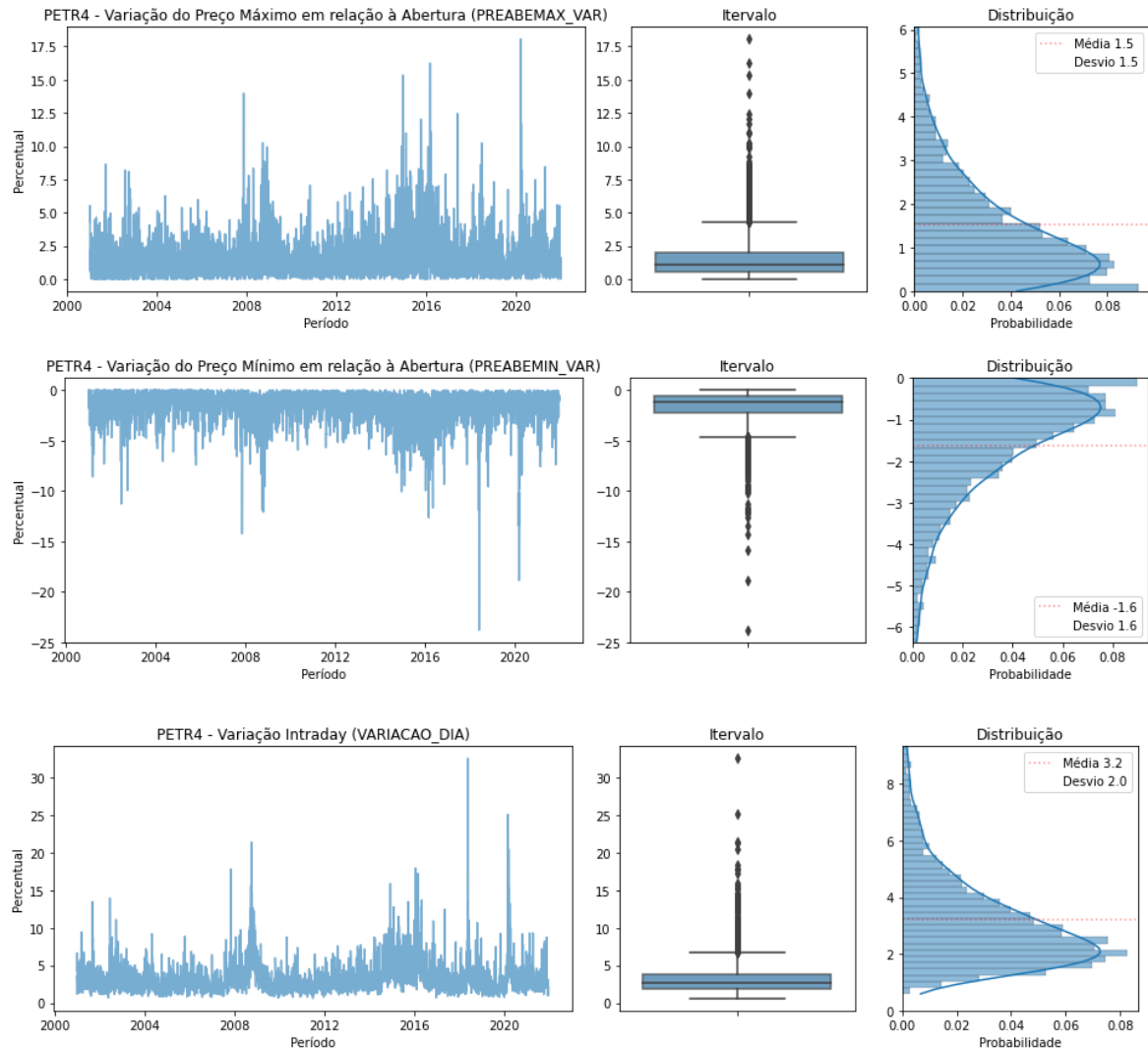
Foram geradas três novas séries a partir da relação entre os preços que representam comportamentos ocorridos durante o pregão:

- **PREABEMAX_VAR**: Representa a variação percentual do Preço Máximo em relação ao Preço de Abertura do dia
- **PREABEMIN_VAR**: Representa a variação percentual do Preço Mínimo em relação ao Preço de Abertura do dia
- **VARIACAO_DIA**: Representa a variação percentual do Preço Máximo em relação ao Preço Mínimo do dia

O comportamento destas variáveis revela um padrão para a Faixa de Negociação para o dia: Podemos esperar que os preços variem entre -6,1% e +6,4% em relação ao preço de abertura, com uma amplitude de até 9,3% entre a máxima e mínima do dia.

	PREABEMAX_VAR	PREABEMIN_VAR	VARIACAO_DIA
Média	1,52%	-1,63%	3,22%
Desvio Padrão	1,51pp	1,58pp	2,03pp
Mediana	1,09%	-1,21%	2,69%
Mínimo	0,00%	-23,82%	0,59%
Máximo	18,06%	0,00%	32,55%

Intervalo a 3 desvios padrão da média	[0%, 6,06%]	[-6,36%, 0%]	[0,0%, 9,33%]
--	-------------	--------------	---------------



6.2. Análise Bivariada/Multivariada

Nesta sessão, serão verificadas possíveis influências das variáveis disponíveis entre si e em relação às variáveis que definirão a Faixa de Negociação do dia.

A Faixa de Negociação será definida pelas previsões para as variáveis `PREMIN_RET` e `PREMAX_RET` para o dia que se inicia. A previsão deverá ocorrer após conhecer o preço de abertura. Segue definições das novas variáveis:

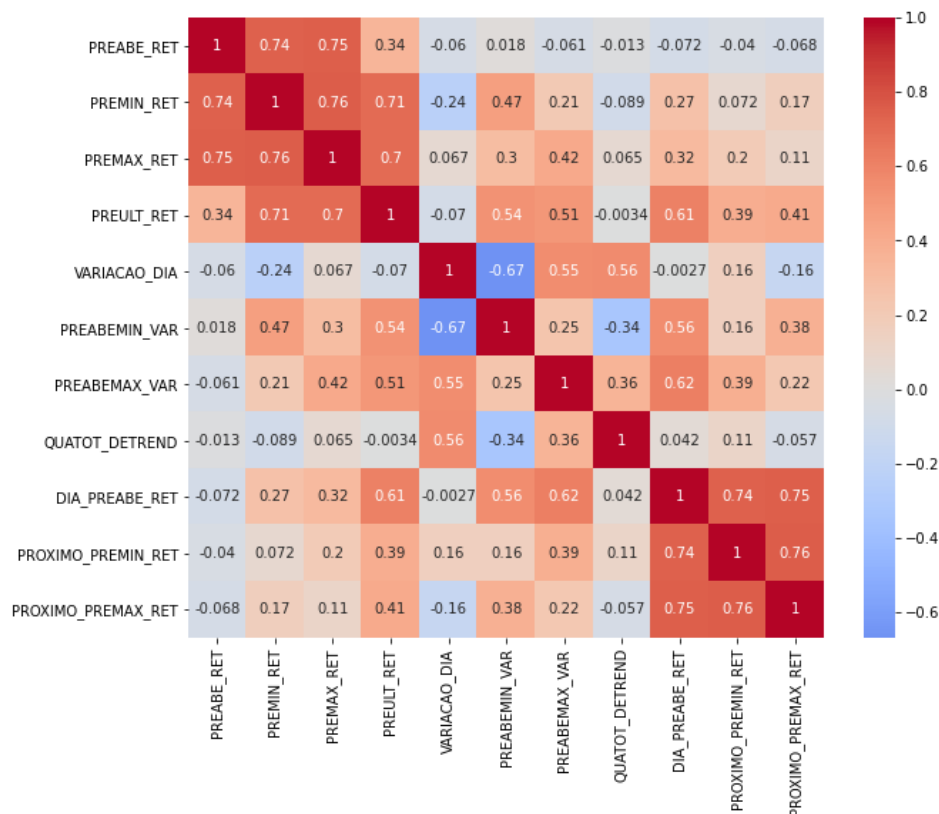
- **DIA_PREABE_RET:** Continuação da série `PREABE_RET` atualizada no dia do pregão

- **PROXIMO_PREMIN_RET**: Variável a ser predita, sendo o próximo valor da série PREMIN_RET no dia do pregão
- **PROXIMO_PREMAX_RET**: Variável a ser predita, sendo o próximo valor da série PREMAX_RET no dia do pregão

6.2.1 Mapa de Correlação

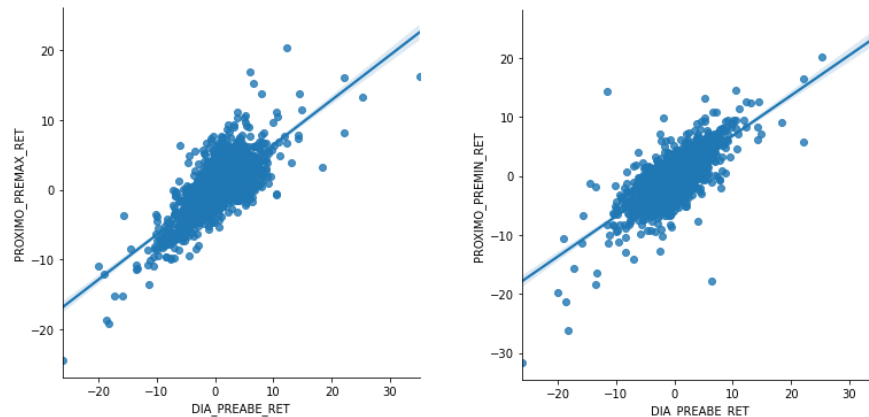
O mapa de calor a seguir mostra o coeficiente de correlação de Pearson entre as variáveis independentes e também entre as que queremos prever. Alguns valores se destacam:

- A variável DIA_PREABE_RET é a que apresenta maior correlação com as variáveis a serem previstas
- As variáveis PREMIN_RET e PREABEMIN_VAR tem maior correlação com a variável a ser predita PROXIMO_PREMAX_RET do que a PROXIMO_PREMIN_RET
- As variáveis PREMAX_RET e PREABEMAX_VAR tem maior correlação com a variável a ser predita PROXIMO_PREMIN_RET do que a PROXIMO_PREMAX_RET
- A variável VARIACAO_DIA apresenta correlações com sentidos opostos em relação às variáveis a serem previstas



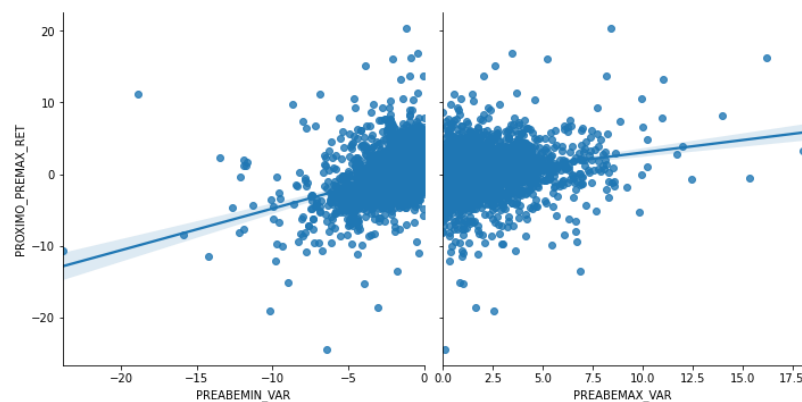
6.2.2 Preço de abertura do dia (DIA_PREABE_RET)

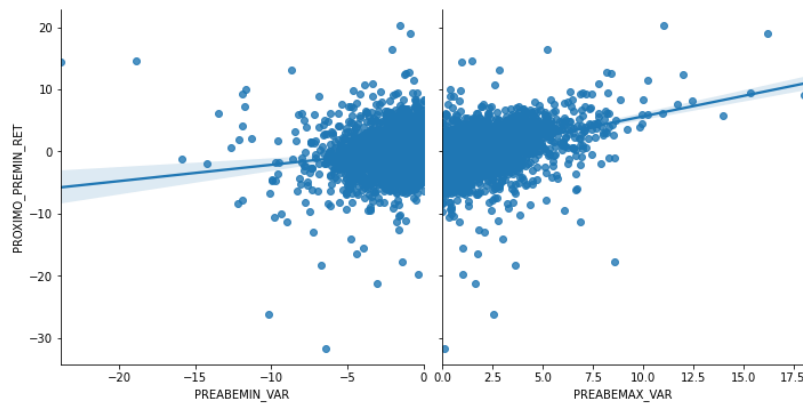
O comportamento da variação do preço de abertura do pregão e das variáveis a serem preditas é bastante semelhante. Ela carrega a variação ocorrida desde a abertura do dia anterior, além da consolidação de informações pelos investidores enquanto o mercado esteve fechado.



6.2.3 Preço mínimo e máximo em relação à abertura (PREABEMIN_VAR, PREABEMAX_VAR)

Nos gráficos de dispersão a seguir, é possível perceber que a variável PROXIMO_PREMAX_RET é mais sensível à variação do PREABEMIN_VAR, enquanto a variável PROXIMO_PREABEMIN_RET possui uma reta de regressão mais inclinada com a variável PREABEMAX_VAR.

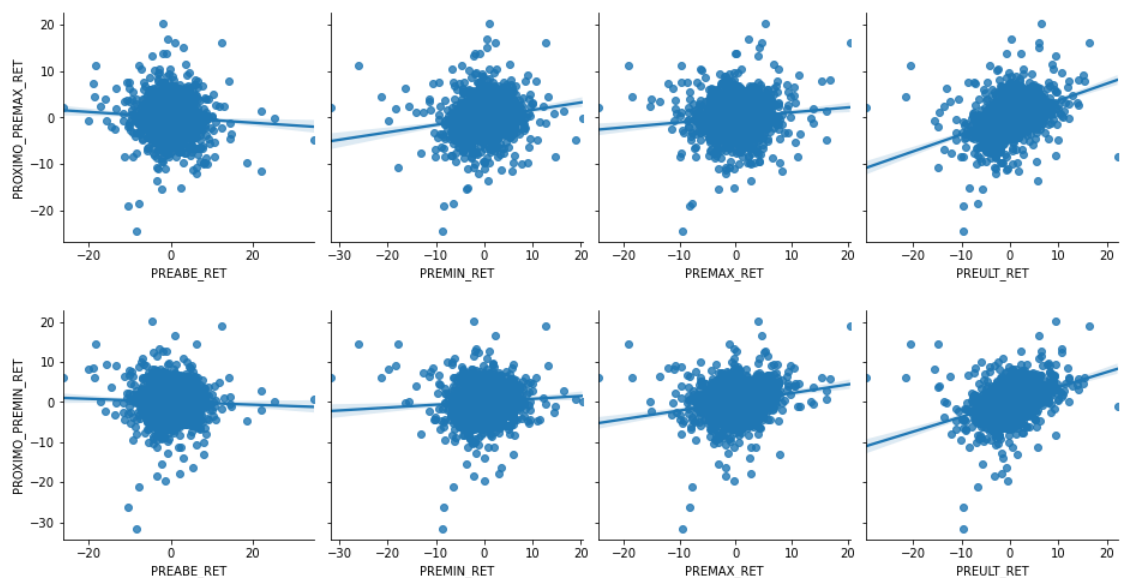




6.2.4 Preços de Abertura, Mínimo, Máximo e Fechamento (PREABE_RET, PREMIN_RET, PREMAX_RET, PREULT_RET)

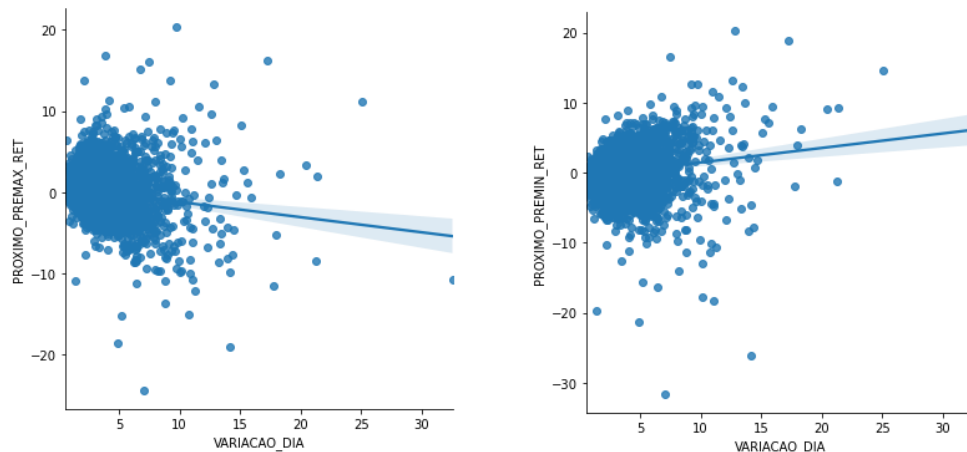
Em relação às variações de preço até o dia anterior, a maior influência é verificada com a variável PREULT_RET. A reta de regressão em relação à variação do preço de abertura (este do dia anterior ao pregão da previsão) esboça pouca inclinação e negativa.

Já em relação às máximas e mínimas do dia anterior, novamente o padrão se repete com a influência da variação do preço mínimo esboçando mais influência sobre o preço máximo do dia seguinte do que a própria autocorrelação com a variável preço máximo. Mesmo padrão ocorre com o preço mínimo do dia seguinte.



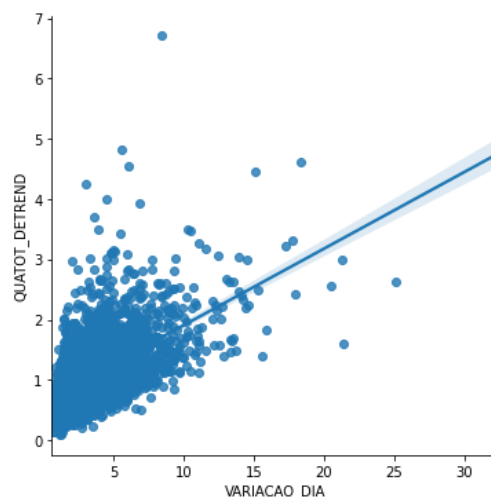
6.2.5 Variação Intradiária (VARIACAO_DIA)

O efeito da variação intradiária é inverso em relação à variação dos preços máximos e mínimos do dia seguinte: Quanto maior o valor da variação intradiária, maior tende ser a variação negativa do preço máximo do dia seguinte. Já o preço mínimo do dia seguinte tende a ter uma variação positiva. Isto implica uma faixa de negociação com amplitude menor quanto maior for a do dia anterior.



6.2.6 Relação da Variação Intradiária com o Volume (VARIACAO_DIA, QUATOT_DETREND)

O volume de títulos negociados guarda forte correlação positiva com a amplitude da variação intradiária: Quanto maior a variação do preço máximo em relação ao mínimo no dia, maior tende a ser a quantidade de títulos negociados.



7. Preparação dos Dados para os Modelos de Aprendizado de Máquina

Como já comentado anteriormente, não serão utilizados diretamente os preços e volumes por não representarem os valores praticados à época devido os ajustes por conta dos desdobramentos. Ao invés disso, serão utilizadas séries derivadas com as relações de variações nas séries temporais, tanto diárias quanto intradiárias.

Outro motivo para não utilizar os preços diretamente é a dificuldade de extrapolação dos algoritmos Floresta Randômica (FR) e Rede Neural Artificial (RNA). Esta característica traz certa limitação para os modelos preverem valores que não tiveram contato durante o treinamento, e por isso teriam dificuldade em prever um preço nunca antes atingido. Já o algoritmo Regressão Linear (RL) não sofre com esta limitação.

Com base nas possíveis relações de dependência apontados nas análises anteriores, foram selecionadas as seguintes variáveis que servirão como entrada para os modelos de aprendizagem:

Variáveis de Entrada (FEATURES)	Descrição
PREMIN_RET	Variação do preço mínimo em D-1 em relação ao do dia anterior (D-2).
PREMAX_RET	Variação do preço máximo em D-1 em relação ao dia anterior (D-2)
PREULT_RET	Variação do preço máximo em D-1 em relação ao dia anterior (D-2)
PREABEMIN_VAR	Variação do preço mínimo em relação ao preço de abertura em D-1
PREABEMAX_VAR	Variação do preço máximo em relação ao preço de abertura em D-1
VARIACAO_DIA	Variação entre o preço máximo e o mínimo em D-1
QUATOT_DETREND	Total de títulos negociados (Volume) corrigido pela média em D-1
PREABE_RET	Variação do preço de abertura no dia da previsão (D-0) em relação ao do dia anterior (D-1)
DIA	Dia da semana no dia da previsão (D-0)

Para definição da Faixa de Negociação para o pregão, tentaremos prever a variação do preço máximo e a variação do preço mínimo em relação aos atingidos no dia

anterior. A partir destes percentuais, serão calculados o Preço Mínimo e o Preço Máximo nos quais serão aplicadas as métricas de desempenho. Assim, as variáveis de saída dos modelos serão duas:

Variáveis de Saída (TARGETS)	Descrição
PROXIMO_PREMIN_RET	Variação entre o preço mínimo e o preço de abertura no dia do pregão (D-0)
PROXIMO_PREMAX_RET	Variação entre o preço máximo e o preço de abertura para o pregão (D-0)

O código a seguir transforma e extrai as variáveis que iremos utilizar:

```

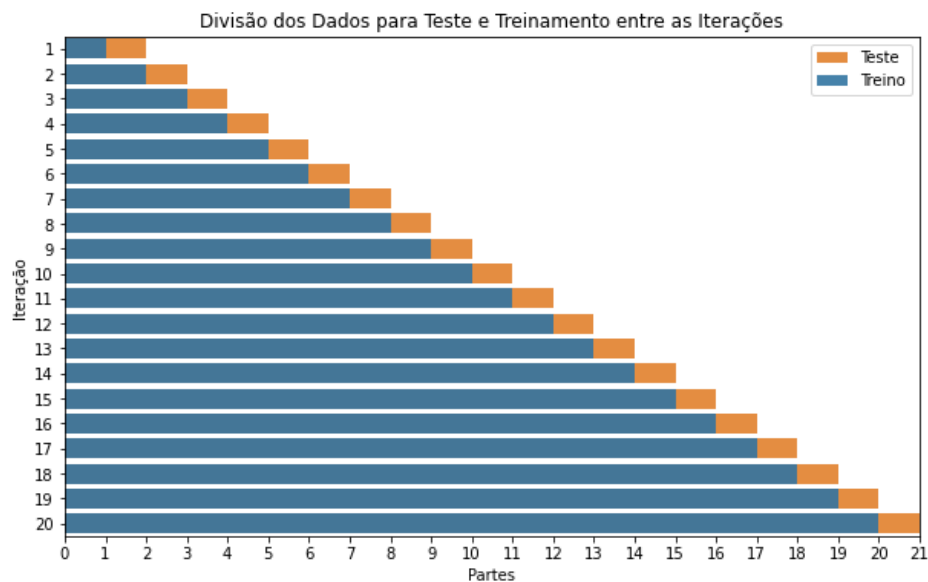
1 def extract_features_and_targets(df):
2     '''Extração e transformação de variáveis'''
3     df_dataset = pd.DataFrame(index=df.index)
4
5     # Volume ajustado
6     df_dataset['QUATOT_DETREND'] = df['QUATOT'].shift(+1) / df['QUATOT'].shift(+1).ewm(span=245).mean()
7
8     # Retornos diários
9     df_dataset['PREABE_RET'] = (df['PREABE'].shift(+1) / df['PREABE'].shift(+2) - 1.0)
10    df_dataset['PREMIN_RET'] = (df['PREMIN'].shift(+1) / df['PREMIN'].shift(+2) - 1.0)
11    df_dataset['PREMAX_RET'] = (df['PREMAX'].shift(+1) / df['PREMAX'].shift(+2) - 1.0)
12    df_dataset['PREULT_RET'] = (df['PREULT'].shift(+1) / df['PREULT'].shift(+2) - 1.0)
13
14    # Variações Intradiaárias
15    df_dataset['VARIACAO_DIA'] = (df['PREMAX'].shift(+1) / df['PREMIN'].shift(+1) - 1.0)
16    df_dataset['PREABEMIN_VAR'] = (df['PREMIN'].shift(+1) / df['PREABE'].shift(+1) - 1.0)
17    df_dataset['PREABEMAX_VAR'] = (df['PREMAX'].shift(+1) / df['PREABE'].shift(+1) - 1.0)
18
19    # Entradas no dia do Pregão
20    df_dataset['DIA_PREABE_RET'] = (df['PREABE'].shift(0) / df['PREABE'].shift(+1) - 1.0)
21    df_dataset['DIA'] = df.index.dayofweek
22
23    # - - -
24    # Saídas em D-0
25    df_dataset['PROXIMO_PREMIN_RET'] = (df['PREMIN'].shift(0) / df['PREMIN'].shift(+1) - 1.0)
26    df_dataset['PROXIMO_PREMAX_RET'] = (df['PREMAX'].shift(0) / df['PREMAX'].shift(+1) - 1.0)
27    # - - -
28    # descarta registros com valores inválidos
29    df_dataset.dropna(inplace=True)
30    # separa features e targets
31    targets = ['PROXIMO_PREMIN_RET', 'PROXIMO_PREMAX_RET']
32    X, y = (df_dataset.drop(columns=targets),
33            df_dataset[targets])
34    # - - -
35    return X, y
36    # - - -

```

7.1 Dados para Treino e Teste

Será utilizada uma variação do método de validação cruzada para trabalho com séries temporais, respeitando a cronologia das coletas de forma parecida com o que acontece na prática: para predizer o futuro temos acesso apenas a dados passados.

Os 21 anos de registros disponíveis serão divididos em 21 partes, sendo que os testes serão realizados sempre em dados referente a uma parte, equivalente a um ano, com modelos treinados nas partes que a antecedem. A cada iteração, a parte de teste será deslocada adiante, fazendo com que o intervalo de dados de treinamento seja expandido. A figura a seguir ilustra este procedimento:



Dessa forma, teremos como avaliar o comportamento dos modelos durante vários momentos do Mercado ao longo de 20 anos, e o resultado final da avaliação será a média dos valores dos indicadores de desempenho obtidos.

O código a seguir realiza esta divisão dos dados utilizando a classe `TimeSeriesSplit`, da biblioteca Scikit-Learn, e itera entre os treinamentos e testes:

```
1 from sklearn.model_selection import TimeSeriesSplit
2
3 # extrai features e targets
4 X, y = extract_features_and_targets(df_historico_ativo)
5
6 # dados de treino de 1 a 20 anos, com 1 ano de teste
7 train_test_split = TimeSeriesSplit(n_splits=20)
8
9 # itera entre os testes
10 all_scores = []
11 for train_index, test_index in train_test_split.split(X):
12     X_train, y_train = X.iloc[train_index], y.iloc[train_index]
13     X_test, y_test = X.iloc[test_index], y.iloc[test_index]
14     test_period = X.index[[test_index[0], test_index[-1]]]
15
16 # treina e avalia
17 model.fit(X_train, y_train)
18 y_pred = model.predict(X_test)
19 all_scores.append(dict(período=test_period, scores=calc_metrics(y_test.values, y_pred)))
20 # ...
```

8. Aplicação de Modelos de Aprendizado de Máquina

Nesta seção, serão implementados e avaliados os três modelos de aprendizagem propostos, todos disponíveis na biblioteca scikit-learn. Serão treinados e validados a partir do mesmo conjunto de dados, por validação cruzada, conforme já descrito anteriormente.

O código a seguir descreve a obtenção dos dados que serão utilizados para treinamento e validação dos modelos:

```
1 # seleciona dados da empresa
2 df_historico_ativo = get_historico_empresa('PETR4', df_historico, df_desdobramentos)
3
4 # extrai features e targets para modelos de aprendizagem
5 X, y = extract_features_and_targets(df_historico_ativo)
6
7 # dados de treino de 1 a 20 anos, com 1 ano de teste
8 from sklearn.model_selection import TimeSeriesSplit
9 train_test_split = TimeSeriesSplit(n_splits=20)
```

Os dados serão divididos em vários segmentos para treino e validação, conforme descrito no código da função abaixo que irá retornar os valores de Erro Percentual Absoluto Médio (MAPE) das previsões em cada iteração:

```
1 def calc_mape(y_true, y_pred, target_names):
2     targets_scores = {}
3     for t in range(y_true.shape[1]):
4         trues = y_true[:, t] + 1.
5         preds = y_pred[:, t] + 1.
6         mape = 100. * np.mean(np.abs(trues - preds) / np.abs(trues))
7         targets_scores[target_names[t]] = mape
8     return targets_scores
9 # - - -
10
11 # itera entre os testes
12 def avaliar_modelo(model, X, y, train_test_split):
13     '''Avalia o modelo por validação cruzada para série temporal'''
14     all_scores = []
15     for train_index, test_index in train_test_split.split(X):
16         X_train, y_train = X.iloc[train_index], y.iloc[train_index]
17         X_test, y_test = X.iloc[test_index], y.iloc[test_index]
18         test_period = X.index[[test_index[0], test_index[-1]]]
19         # treina e avalia
20         model.fit(X_train, y_train)
21         y_pred = model.predict(X_test)
22         scores = calc_mape(y_test.values, y_pred, target_names=y.columns)
23         #
24         scores['periodo'] = test_period
25         all_scores.append(scores)
26         # - - -
27     return all_scores
```

8.1. Erro Percentual Absoluto Médio (MAPE)

A métrica de desempenho MAPE foi escolhida por apresentar um valor que não é impactado pelo nível de preços ao longo da série temporal, além de ser expressa em uma grandeza cotidianamente utilizada pelos operadores no mercado de ações: variação percentual. Ela é calculada fazendo a média simples das diferenças, em

percentual, entre o valor previsto e o valor real, considerando o valor absoluto. Como resultado, temos um valor percentual médio para o erro das previsões.

8.2. Modelo de Regressão Linear

Os modelos de Regressão Linear, utilizados para tarefas de regressão em aprendizagem supervisionada, buscam aproximar o valor de uma variável de saída (dependente) por meio da equação de uma reta, atribuindo constantes e pesos (coeficientes) para as variáveis de entrada (independentes) de forma a minimizar a diferença entre os valores previstos e os esperados.

O código a seguir implementa o modelo a partir da classe `LinearRegression` disponível na biblioteca `scikit-learn`, e em seguida submete para avaliação:

```
1 from sklearn.linear_model import LinearRegression
2
3 lr_model = LinearRegression()
4
5 lr_scores = avaliar_modelo(lr_model, X, y, train_test_split)
```

A tabela abaixo mostra o resumo dos resultados dos testes:

Erro Percentual Absoluto Médio - MAPE		
	PROXIMO_PREMIN_RET	PROXIMO_PREMAX_RET
Total de Iterações	20	20
Média	1,05%	0,96%
Mediana	0,92%	0,84%
Mínimo	0,75%	0,69%
Máximo	1,56%	1,50%

8.3. Modelo Floresta Randômica

Os modelos de aprendizagem Árvore de Decisão, aplicáveis a tarefas supervisionadas também de regressão, buscam a partir do Nó Raiz que engloba todo o conjunto de dados, encontrar regra baseada em alguma variável de entrada

(independente) que melhor o subdivide em novos nós. Para cada novo nó, o processo é repetido até que não seja mais possível subdivisão dos dados, ou alcance alguma outra regra de término, sendo estes últimos chamados de Nós Folhas. Os conjuntos de regras geradas desde o Nó Raiz até cada Folha são as regras de classificação ou regressão, as quais são utilizadas para prever as variáveis de saída (dependentes).

Já os modelos de aprendizagem Floresta Randômica se baseiam na combinação de várias Árvores de Decisão, treinadas a partir de subconjuntos aleatórios (randômicos) dos dados de entrada. O resultado final da previsão será o padrão predominante entre os resultados de todas as árvores componentes.

O código a seguir implementa o modelo a partir da classe RandomForestRegressor disponível na biblioteca scikit-learn, configurado para usar 100 árvores de decisão internamente, com no máximo 10 níveis de altura. Em seguida o modelo é submetido para avaliação:

```
1 from sklearn.ensemble import RandomForestRegressor
2
3 rf_model = RandomForestRegressor(n_estimators=100, max_depth=10, random_state=42)
4
5 rf_scores = avaliar_modelo(rf_model, X, y, train_test_split)
```

A tabela abaixo mostra o resumo dos resultados dos testes:

Erro Percentual Absoluto Médio - MAPE		
	PROXIMO_PREMIN_RET	PROXIMO_PREMAX_RET
Total de Iterações	20	20
Média	1,11%	1,02%
Mediana	0,98%	0,88%
Mínimo	0,78%	0,71%
Máximo	1,73%	1,61%

8.4. Modelo Rede Neural Artificial

O modelo de aprendizagem Rede Neural Artificial tem inspiração no que se conhece do funcionamento do cérebro biológico. Nesta sessão, utilizaremos uma arquitetura

Multicamadas (*MultiLayer Perceptrons - MLP*) que são encadeamentos tanto em série quanto em paralelo das unidades básicas de decisão, chamadas de Neurônios Artificiais (*perceptrons*). Cada uma dessas unidades consiste em uma equação linear recebendo valores aos quais são aplicados pesos, somados, e o resultado é transformado por uma função não linear (função de ativação) gerando seu valor de saída. Cada aglomerado em paralelo de Neurônios é chamado de Camada, onde todos compartilham os mesmos valores de entrada, e suas saídas podem ser encadeadas em série alimentando uma próxima camada de neurônios. A primeira camada (Camada de Entrada) são as próprias variáveis independentes, e a saída da última camada (Camada de Saída) entrega o valor da variável que se quer prever (dependente), sendo que entre estas duas ficam as Camadas Escondidas.

A aprendizagem supervisionada desse modelo resulta do ajuste dos pesos aplicados às entradas dos diversos neurônios da rede de forma a aproximar o valor de saída com o valor real esperado para cada conjunto de variáveis de entrada.

O código a seguir implementa o modelo a partir da classe `MLPRegressor` disponível na biblioteca `scikit-learn`, com uma camada escondida contendo 30 neurônios com função de ativação Tangente Hiperbólica, e a camada de saída com dois neurônios de saída. Em seguida o modelo é submetido para avaliação:

```
1 from sklearn.neural_network import MLPRegressor
2
3 nn_model = MLPRegressor(
4     hidden_layer_sizes=(30,),
5     activation='tanh',
6     batch_size=32,
7     solver='adam', learning_rate_init=1e-4,
8     tol=1e-7, n_iter_no_change=10, max_iter=2000,
9     random_state=42,
10 )
11
12 nn_scores = avaliar_modelo(nn_model, X, y, train_test_split)
```

A tabela a seguir mostra o resumo dos resultados dos testes:

Erro Percentual Absoluto Médio - MAPE		
	PROXIMO_PREMIN_RET	PROXIMO_PREMAX_RET
Total de Iterações	20	20
Média	1,06%	1,02%
Mediana	0,96%	0,91%
Mínimo	0,76%	0,68%
Máximo	1,62%	1,55%

9. Discussão dos Resultados

Os três modelos apresentaram desempenho muito semelhantes, sendo que o modelo de Regressão Linear obteve menores valores nos testes, com média dos erros nos ficando em 1,05% e 0,96% para as variáveis previstas PROXIMO_PREMIN_RET e PROXIMO_PREMAX_RET, respectivamente.

O gráfico abaixo compara o desempenho dos modelos durante as várias iterações sobre os segmentos de dados ao longo dos anos. É possível observar alguns anos com aumento expressivo nos erros, como 2008, 2014, 2015 e 2016, que coincidem com momentos de grande incerteza dos investidores registrados no histórico de preços destes anos, marcados inclusive com reversões de longas tendências.



9.1. Pipeline completo até a Faixa de Negociação

A seguir, será resumido todo o processo de obtenção e tratamento dos dados, construção do modelo, previsão e aplicação da Faixa de Negociação em um pequeno intervalo de dados separados para validação.

Considerando que os arquivos de dados já foram baixados para a devida pasta, e que a planilha de desdobramentos está preenchida com o histórico do ativo a ser analisado, é realizada a importação, ajuste e seleção dos dados referente ao ativo a ser analisado, conforme descrito no código abaixo:

```

1 # define o código do ativo base
2 codigo_ativo = 'PETR4'
3
4 # Importa dados da B3
5 df_historico = importar_dados_b3()
6
7 # Ajusta Desdobramentos de Ações
8 df_desdobramentos = importar_desdobramentos(DESDOBRAMENTOS_XLS)
9
10 # Seleciona apenas dados do ativo estudado
11 df_historico_ativo = get_historico_empresa(codigo_ativo, df_historico, df_desdobramentos)

```

Em seguida, a partir das informações do histórico de preços e volumes de negócios das ações da Petrobrás (PETR4), são aplicadas transformações e extrações das variáveis treinamento do modelo de aprendizagem. Para fins de demonstração, foram reservados dados dos últimos 4 meses do ano de 2021 para ser usado na predição. O código a seguir exibe os passos da extração de *features*, treinamento até a previsão:

```

1 # extrai variáveis
2 X, y = extract_features_and_targets(df_historico_ativo)
3
4 # separa dados para treino e para validação
5 X_train, y_train = X.loc[:'2020-08-31'], y.loc[:'2020-08-31']
6 X_valid, y_valid = X.loc['2021-09-01:'], y.loc['2021-09-01:']
7
8 # cria e treina modelo
9 from sklearn.linear_model import LinearRegression
10 model = LinearRegression()
11 model.fit(X_train, y_train)
12
13 # realiza previsão
14 y_pred = model.predict(X_valid)

```

A partir das variações em percentuais previstas pelo modelo treinado para o Preço Máximo e o Preço Mínimo do dia, é calculada a Faixa de Negociação para o dia que se inicia. Em seguida, é aplicada a variação percentual prevista sobre os valores dos preços máximos e mínimos do dia imediatamente anterior:

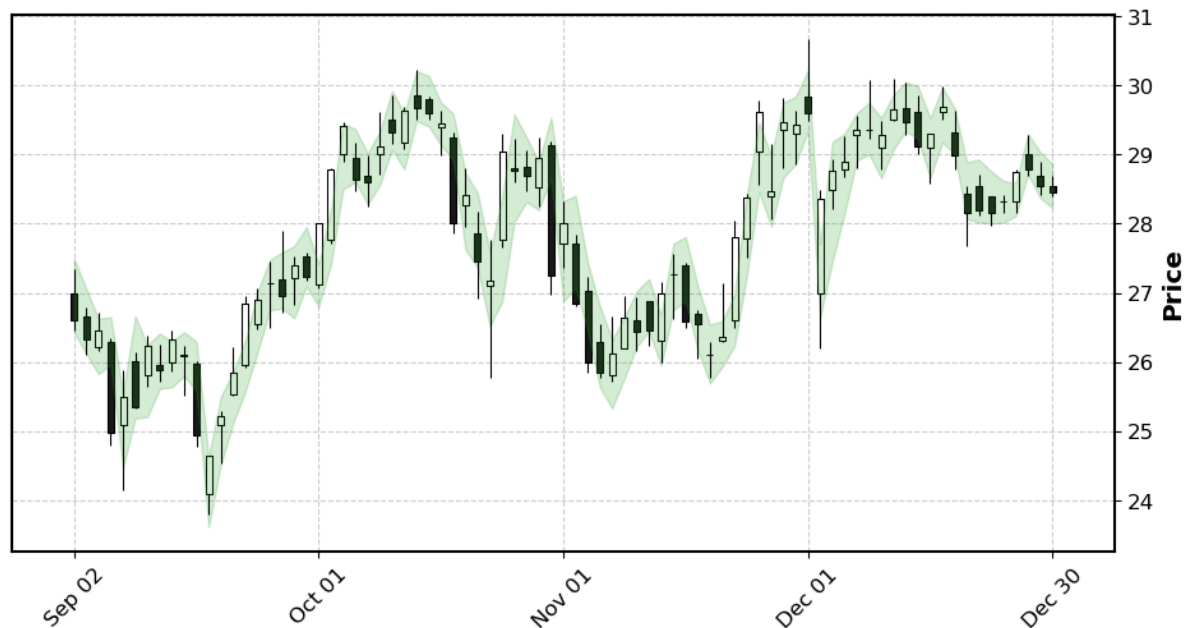
```

1 def aplicar_previsao(y_pred, indices, df_historico):
2     '''Calcula PRED_PREMAX e PRED_PREMIN (Faixa de Negociação) a partir das previsões'''
3     y_pred = pd.DataFrame(data=y_pred, columns=['PROXIMO_PREMIN_RET', 'PROXIMO_PREMAX_RET'], index=indices)
4     # calcula faixa a partir dos percentuais previstos
5     df_hist_extend = df_historico.loc[indices].join(y_pred)
6     df_hist_extend['PRED_PREMAX'] = (1+df_hist_extend['PROXIMO_PREMAX_RET']) * df_hist_extend['PREMAX'].shift(+1)
7     df_hist_extend['PRED_PREMIN'] = (1+df_hist_extend['PROXIMO_PREMIN_RET']) * df_hist_extend['PREMIN'].shift(+1)
8     return df_hist_extend.dropna()
9     -----
10
11 df_validacao = aplicar_previsao(y_pred, X_valid.index, df_historico_ativo)

```

Dessa forma, agora temos as variáveis PRED_PREMAX e PRED_PREMIN, com valores previstos para o preço máximo e o preço mínimo, respectivamente, para o dia que se inicia. O Gráfico de *candles* abaixo exibe os últimos quatro meses de variações diárias em 2021 para a ação da Petrobrás, e o preenchimento em verde representa a **Faixa de Negociação** gerada a partir das previsões do modelo, onde é esperado que se contenha o movimento do preço durante o dia:

PETR4 - Histórico de Preços com Faixa de Negociação



Para gerar o gráfico acima foi utilizada a biblioteca MPLFinance, que suporta gráficos voltados para análises financeiras em Python, conforme o código abaixo:

```
1 # exibe gráfico de preços de ações com Faixa de Negociação prevista
2 import mplfinance as mpf
3 df_plot = df_validacao.rename(columns={'PREABE': 'Open', 'PREULT': 'Close', 'PREMAX': 'High', 'PREMIN': 'Low'}).dropna()
4 mpf.plot(df_plot,
5         type='candle',
6         title='PETR4 - Histórico de Preços com Faixa de Negociação',
7         style='classic',
8         fill_between=dict(y1=df_plot['PRED_PREMAX'].values, y2=df_plot['PRED_PREMIN'].values, color='tab:green', alpha=0.2),
9         figsize=(12,6),
10 )
```

10. Conclusão

Dada a complexidade da formação dos preços dos ativos financeiros, a previsão do seu comportamento futuro é um desafio, e identificar algum padrão que possa reduzir a incerteza gera vantagem competitiva para quem opera neste mercado. Diante da problemática, este trabalho propôs e apresentou o uso de modelos de aprendizagem de máquina para prever uma Faixa de Negociação na qual o preço de uma ação tende a se conter durante o pregão na bolsa de valores, de modo que possa auxiliar a tomada de decisão do investidor *daytrade*.

Para análise e treinamento dos modelos, foram utilizados o histórico de cotações diárias de preços registrados durante 21 (vinte e um) anos – de 2001 à 2021 – da empresa Petrobrás (PETR4). Estes dados foram pré-processados e ajustados, antes

de seguir para etapas de exploração, transformação e seleção dos atributos relevantes para os modelos de aprendizagem.

Registraram resultados muito semelhantes os três modelos testados: Rede Neural Artificial, Floresta Randômica e Regressão Linear, sendo que este último foi o que apresentou o melhor desempenho, tanto no tempo de processamento no treinamento, quanto no Erro Percentual Absoluto Médio (MAPE) de 1,05% na previsão do Preço Mínimo e 0,96% na do Preço Máximo.

Na rotina de testes, os dados de validação foram separados por anos civis, sendo que os piores resultados coincidiram com anos marcados com fortes crises políticas e econômicas como a crise da bolha imobiliária americana em 2008; a crise político-econômica com recessão técnica no Brasil em 2014, seguida pelo agravamento em 2015 e impeachment da então presidente Dilma Rousseff em 2016. Nesses anos, o erro MAPE atingiu valores acima de 1,50%.

Por fim, considerando a dimensão do problema em um ambiente influenciado por tantos fatores, a previsão utilizando tão somente o histórico diário de preços talvez pudesse ser melhorada com informações em periodicidades mais curtas, como por hora, ou por minuto, ou até mesmo negócio a negócio, que descreveriam com mais detalhes o que ocorreu durante o pregão. Além disso, o modelo não leva em consideração ativos financeiros que concorrem em atenção no mesmo mercado, ou mesmo em mercados adjacentes, tampouco leva em consideração informações sobre sentimentos de mercado vindos de outros meios, como o noticiário. Entretanto, mesmo diante dessas limitações e desafios, o método aqui proposto pode ajudar a reduzir a incerteza do investidor ao ser utilizado para gerar um indicador de Faixa de Negociação.

11. Links

O código fonte da solução discutida neste trabalho pode ser encontrado no GitHub neste link <<https://github.com/wamonzzyne/ML---Faixa-de-Negocia-o.git>>