Universidade Federal de Minas Gerais
Departamento de Ciência da Computação

DCC057 Mineração de Dados

Trabalho Prático 4

Alexander Thomas Mol Holmquist

04 de abril de 2021

1 Introdução

A sistematização do processo de investimento em uma bolsa de valor é um processo antigo. Há tempo que existem teorias como regressão à média ou fator momento, que auxiliam na tomada de decisão. A automatização quase completa deste tipo de tomada de decisão é, por outro lado, um fenômeno recente, mas que já se concretizou, nas últimas duas décadas, com a criação de empresas totalmente dedicadas a este propósito, como [1, 2].

Com o objetivo de fazer o mesmo que estas empresas, este projeto toma como base um conjunto de dados [3] que inclui registros feitos pela Bolsa de Nova York (NYSE), a maior do mundo, de centenas de suas empresas afiliadas. Mais especificamente, o objetivo se tornou criar um modelo preditor, que, dado um conjunto de dados até o dia anterior, tem a capacidade de prever o preço médio de uma ação no próximo dia.

Apesar do número atraente obtido no final (pontuação R2 de quase 100%), este projeto falhou, devido à falta de tempo para consertar um erro cometido. Foi criado um regressor, e não um preditor. Assim, o modelo desenvolvido pelo projeto só é capaz de inferir o preço (um atributo da tabela de dados principal), dado os outros parâmetros para aquele mesmo dia. Este modelo claramente não tem utilidade neste contexto.

Este projeto se baseou no CRISP-DM, principalmente na confecção do plano inicial, contido na proposta. Houve algumas modificações consideráveis, quando comparado com o terceiro projeto desta disciplina. Em retrospectiva, este plano inicial acabou se mostrando pior que o anterior, ao ponto de em certas situações trazer mais confusão que direcionamento.

O repositório remoto do projeto encontra-se na ferramenta GitHub [4].

2 Fonte de Dados

A única fonte de dados utilizada no projeto pode ser encontrada em [3]. Já foi explicado, na proposta, o seu conteúdo, mas aqui será repetido para que o presente relatório não fique faltante. Não foi possível carregá-los para o repositório remoto, devido ao tamanho. Assim, o diretório "data" no repositório remoto contém somente um link para a fonte dos dados. A tabela principal contém 851264 registros e 7 colunas, e se encontra no arquivo "prices.csv" (daqui em diante chamada *pricesDf*). Também foi utilizado o arquivo "fundamentals.csv", que contém uma diversidade impressionante de informações sobre o histórico das empresas.

O significado de cada coluna de pricesDf é explicado abaixo:

- **date:** data a que a entrada se refere.
- **symbol:** o símbolo (abreviatura) correspondente ao nome da empresa referenciada pelo registro.
- **open:** preço de uma ação ao abrir a bolsa no dia.
- **close:** preço de uma ação ao fechar a bolsa no dia.
- **low:** preço mínimo de uma ação da empresa no dia.
- **high:** preço máximo de uma ação da empresa no dia.
- **volume:** quantidade de ações negociadas no dia.

É importante notar que o arquivo “prices.csv” dá uma tabela indexada por tempo. A frequência dos registros é diária. O primeiro dia é 04 de janeiro de 2010, e o último dia é 30 de dezembro de 2016. Foi necessário transformar a coluna “date” que contém as informações temporais para que contenha valores inteiros, a partir de 0.

Nem todos os dias tem um registro correspondente. Somente são registrados dias úteis, e a quantidade de registros não é uniforme para cada dia da semana. Isto é, alguns dias da semana tem mais registros que outros. Apesar disso, a disparidade provavelmente não é causada pela diferença entre os dias da semana em si.

Todos os arquivos do banco de dados estão no formato CSV. Os campos estão separados por vírgula, e registros distintos são separados pelo caractere de nova linha. A primeira linha dos arquivos define os nomes das colunas.

3 Execução do Plano

O plano inicial, baseado no CRISP-DM, é dividido por fases, cada fase é dividida em passos. A execução deste plano é delineada abaixo, com os resultados e considerações resultantes.

3.1 Fase 1 — Carregamento e Descrição dos Dados

Esta fase cobre o carregamento, a realização de checagens básicas de qualidade, assim como descrições que ajudem a se familiarizar com os dados. O Relatório de Carregamento e Descrição dos Dados foi produzido como saída.

3.1.1 Passo 1 — Embarcar Tabela

Foi feito o carregamento dos dados para a ferramenta *Jupyter Notebook*. O interessante aqui é que não só os dados da tabela principal, provinda de “prices.csv”, foram considerados. As funções Python produzidas no código fonte final englobam a possibilidade de carregar qualquer um dos arquivos .csv. Essa foi uma decisão muito feliz, pois na fase 2 foi necessário carregar “fundamentals.csv”, o que pôde ser realizado sem modificar o código. O código fonte pode ser encontrado no arquivo “src/loading/step1.py”.

3.1.2 Passo 2 — Verificar a Qualidade dos Dados

Decidiu-se formatar a coluna de datas “date” de pricesDf da maneira mencionada na Seção 2 já aqui, para facilitar a checagem por valores faltantes, e para assegurar que essa formatação não introduziu mais nenhum. Foi realizada a verificação, e concluído que existiam zero valores faltantes. O código fonte pode ser encontrado no arquivo “src/loading/step2.py”.

3.1.3 Passo 3 — Listagem de Estatísticas Básicas

Foram anotadas as estatísticas: variância, curtose (Fisher) e assimetria. Os resultados são mostrados na Figura 1. Note como a coluna “volume” apresenta valores mais altos que as outras. Este fato foi inicialmente neglicenciado, mas se tornou de suma importância posteriormente, como será mencionado na Seção 3.2.1.

	open	close	low	high	volume
variância	7.00	7.00	6.87	7.13	1.56
curtose	66.33	66.28	66.42	66.17	326.07
assimetria	6.66	6.65	6.66	6.65	13.13

Figura 1: Estatísticas fundamentais de pricesDf, por coluna.

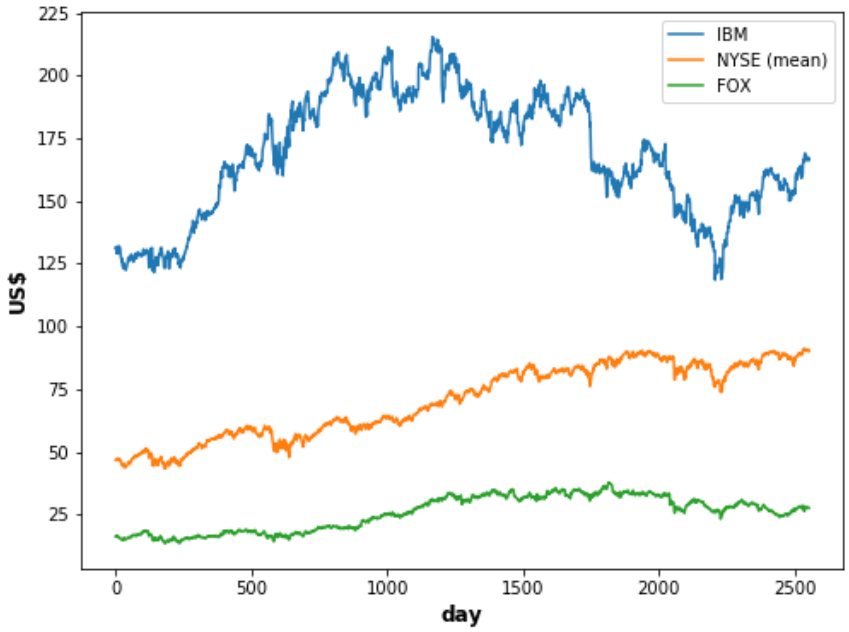

Figura 2: Preço das ações no momento de abertura de mercado, para as empresas conhecidas IBM e Fox. Para a bolsa de Nova York como um todo, a média entre todas as empresas é considerada.

3.1.4 Passo 4 — Produzir Visualizações

Alguns gráficos foram produzidos que podem ajudar muito o leitor a compreender a estrutura dos dados. Decidiu-se utilizar as empresas IBM e Fox como exemplo, para ilustrar as diferenças entre empresas, nos dados.

A Figura 2 mostra uma comparação entre essas duas empresas e a média da Bolsa de Valores, para o preço de abertura de mercado (coluna "open" da tabela). Fica evidente que as ações da IBM desde o início são mais caras que a da Fox. Mais que isso, é possível observar que a variação do preço das ações para a Fox é altamente correlacionada com a média geral para a Bolsa, mas isso não ocorre para a IBM. Talvez a razão principal dessa não-correlação é que a empresa é do ramo tecnológico, o que leva os preços a ser mais flutuantes.

As Figuras 3 e 4 buscam mostrar a *diferente* liquidez de ações das empresas selecionadas. Ao analisar as figuras, porém, o contrário do fica claro: elas são muito parecidas neste aspecto.

3.1.5 Passo 5 — Reduzir Dimensionalidade

Esse passo foi introduzido porque muitas vezes a visualização dos dados com dimensão reduzida por alguma das técnicas conhecidas trás informações interessantes. Contudo, foi decidido

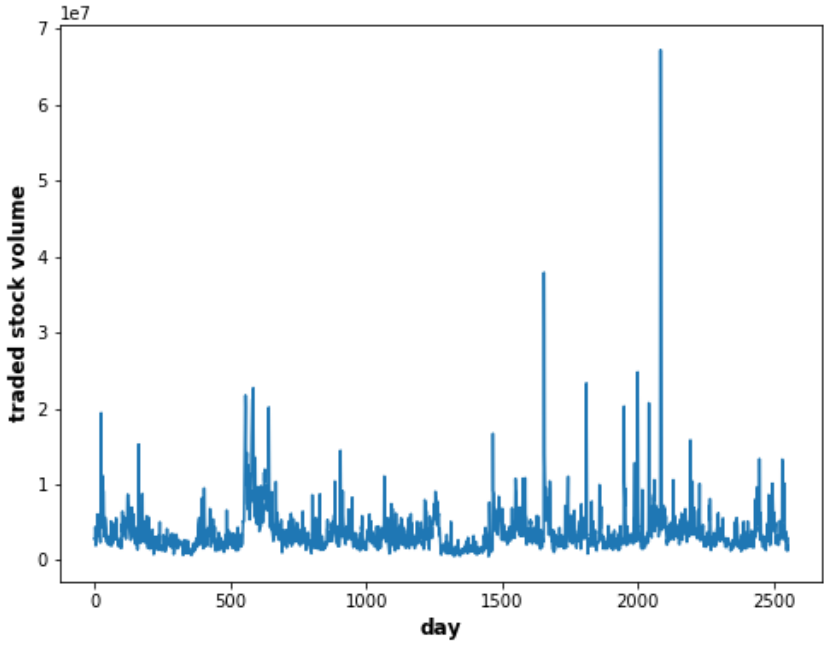

Figura 3: Volume de transações diárias para a empresa Fox

não prosseguir com essa redução para esse conjunto de dados, pois não se mostrou atraente. No processo de tomada desta decisão, algumas considerações cruciais foram feitas, e anotadas para serem resolvidas na fase 2 (ver Seção 3.2.1).

3.1.6 Passo 6 — Explorar os Dados

Este passo é muito abrangente, e tem como objetivo possibilitar cursos de exploração não cobertos pelas sugestões dos passos anteriores. Neste projeto, ao contrário de alguns anteriores da disciplina, não foi feito nada aqui.

3.2 Fase 2 — Preparação dos Dados

3.2.1 Passo 1 — Solucionar Problemas de Qualidade

A primeira transformação realizada, seguindo as anotações constantes no Relatório de Carregamento e Descrição dos Dados, foi substituir as colunas “open” e “close” por novas colunas, com os valores computados da forma:

1. Média — (open + close) / 2
2. Diferença — close - close

Em seguida, após uma discussão de como lidar com a coluna “symbol”, cujos valores são categóricos, foi decidido utilizar os dados fornecidos pelo arquivo “fundamentals.csv” para substituir o símbolo representando a empresa (valor categórico, cadeia de caracteres), pela média dos valores referentes à coluna “Total Assets”, quando presente para a empresa. O

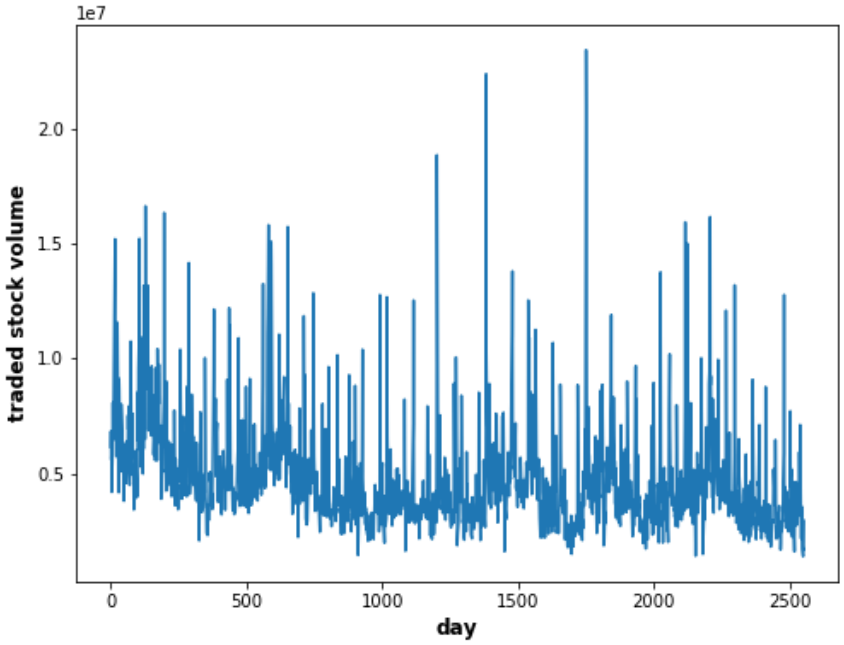

Figura 4: Volume de transações diárias para a empresa IBM

novo identificador é numérico e comparável, características estas mandatórias para que entre no modelo de regressão. A nova coluna foi chamada de "totAssets".

Finalmente, foi percebido que a coluna "volume" e a artificial "totAssets" apresentavam valores de curtose a assimetria muito altos; respectivamente: curtose 86.07 e assimetria 8.69, curtose 327.78 e assimetria 13.32. Notavelmente, a coluna totAssets, apesar de ter valores menores, contém pontos excepcionais que ameaçavam diminuir a qualidade do modelo de regressão. Para transformar essas duas distribuições à semelhança de uma curva gaussiana, foi aplicada uma transformação logarítmica.

3.2.2 Passo 2 — Compilar Lista de Modelos

Este passo se restringe a discutir diferentes modelos candidatos para o projeto (nenhuma ação é necessária). Foram considerados:

1. Regressão linear (elegido);
2. Regressão logística;
3. Árvore de decisão;
4. Rede neuronal.

Os itens de 2 a 4 foram rejeitados já nesse passo; a razão é dada a seguir. Regressão logística foi rejeitado por estar mais ligado com classificação que com regressão. Rede neuronal foi rejeitado por conta da complexidade de implementação, e ainda da possível complexidade computacional do modelo. Árvore de decisão foi rejeitado quando comparado com regressão linear, pois ameaçava ser computacionalmente mais lento, sem, por outro lado, apresentar vantagens — os resultados excepcionalmente interpretáveis produzidos por árvores de decisão não são atraentes para o projeto.

3.2.3 Passo 3 — Construir Ambiente de Testes

Como última saída desta fase, propõe-se a construção de um ambiente de testes, isto é, prontificar código Python para testagem do modelo de regressão linear. Esse código foi produzido, e pode ser encontrado no arquivo "src/preprocessing/step1.py".

Além das transformações feitas no passo 1, o ambiente de testes acrescenta a z-normalização, e um procedimento para dividir os dados em conjuntos de treinamento e teste com 10 dobras.

3.3 Fase 3 — Aplicação dos Modelos e Retrospectiva

3.3.1 Passo 1 — Aplicação dos Modelos

Foi utilizado a função $linear\_model.RidgeCV$, da biblioteca *sklearn*. Esta função realiza regularização pela norma L2 do vetor de pesos. A taxa de regularização é calibrada automaticamente, por validação cruzada. Um vetor de taxas a serem testadas é passado como parâmetro, e a função então testa o modelo com cada uma, escolhendo aquela para qual obter melhor pontuação R2.

3.3.2 Passo 2 — Calibração dos Parâmetros

Com a utilização da biblioteca $sklearn$, como mencionado acima, não foi necessário calibrar os meta-parâmetros do modelo de qualquer forma. A taxa ideal encontrado pelo procedimento foi aproximadamente 0.278.

3.3.3 Passo 3 — Retrospectiva

Neste passo, é proposto que se faça uma retrospectiva do processo de mineração de dados, como garantia de que o modelo em mãos é o melhor possível. Contudo, existe um problema, que foi mencionado na Introdução. O modelo produzido é um "regressor", e não um "preditor", como havia sido planejado.

Mesmo assim, a escolha de regressão linear, em específico, parece ter sido correta. Além disso, o que impediu a correção deste erro foi o prazo; não houve tempo suficiente para voltar a trás. Portanto, pode-se dizer que o modelo em mãos é o "melhor possível", como havia sido colocado para esse passo na proposta.

3.4 Fase 4 — Avaliação

3.4.1 Passo 1 — Considerar os Resultados Obtidos

Foi obtido uma pontuação R2 de 99.9976%, obviamente um absurdo (muito alto). Isso provavelmente ocorreu, em primeiro lugar, pela falha mencionada na seção anterior. Em segundo lugar, há uma altíssima correlação entre o atributo independente "ocMean", e os atributos dependentes, notavelmente "low" e "high".

Para concluir, é preciso deixar claro que, apesar da alta pontuação, este projeto falhou em cumprir seu objetivo de negócio inicial, que era "produzir um preditor robusto para o preço de ações na bolsa de Nova York". Vale a pena mencionar ainda que o objetivo principal de mineração de dados correspondente ao objetivo principal de negócio não foi colocado de maneira correta na proposta. Era para constar "preditor", onde se encontra a palavra "regressor".

Referências

[1] https://www.betterment.com/. Accessado em 19 de março de 2021.

[2] https://www.wealthfront.com/. Acessado em 19 de março de 2021.

[3] https://www.kaggle.com/dgawlik/nyse. Acessado em 19 de março de 2021.

[4] https://github.com/Yowgf/MD-TP4. Acessado em 04 de abril de 2021.