Projeto 3 -Projeto Final

Felipe Buniac e Rafael Molines

24 de Novembro de 2017

1 Introdução

O mercado de ações é bastante complexo, pois existem diversos fatores que influenciam no preço e no comportamento dessas ações. Fatores estes, que são difíceis de prever, até para os melhores profissionais da área. Esse assunto interessa muita gente e existem vários estudos que buscam criar estratégias vencedoras na bolsa.

Tendo em vista que a previsão de comportamento de ações é um assunto debatido e que desafia economistas e analistas há décadas, escolhemos estudar mais a fundo esse tema. Para compreender a dificuldade e ainda tentar prever com certa precisão comportamentos futuros da bolsa, o presente relatório consiste da descrição de um modelo preditivo que utiliza regressão logística para decidir se o valor de fechamento de certa ação da bolsa irá subir ou descer no dia seguinte, utilizando como base para essa previsão a variação relativa do preço de fechamento da mesma e também de outras ações para uma certa janela de tempo passada.

2 Descrição da solução

Buscando verificar a possibilidade de prever o comportamento de ações da bolsa, contamos com mais de 20 anos de dados sobre o preço do fechamento diário de inúmeras ações de diversas empresas, atualizados diariamente. Criamos então, um modelo que não se mostrou muito efetivo neste propósito e provou a dificuldade de tal predição.

Regressão Logística

O modelo preditivo construído utiliza a regressão logística para fazer previsões. A regressão logística tem como objetivo produzir, a partir de um conjunto de observações, um modelo que permita a predição de valores tomados por uma variável categórica, frequentemente binária, a partir de uma série de variáveis explicativas contínuas e/ou binárias.[1]

Na regressão logística, a probabilidade de ocorrência de um evento pode ser estimada diretamente. No caso da variável dependente Y assumir apenas dois possíveis estados (1 ou 0) e haver um conjunto de p variáveis independentes $X_1, X_2, ..., X_p$, o modelo de regressão logística pode ser escrito da seguinte forma:

$$P(Y = 1) = \frac{1}{1 + e^{-g(x)}}$$

onde,

$$g(x) = B_0 + B_1X_1 + \cdots + B_pX_p$$

Os coeficientes $B_0, B_1, \cdots, B_p$ são estimados a partir do conjunto dados, pelo método da máxima verossimilhança, em que encontra uma combinação de coeficientes que maximiza a probabilidade da amostra ter sido observada.Considerando uma certa combinação de coeficientes $B_0, B_1, \cdots, B_p$ e variando os valores de X. Observa-se que a curva logística tem um comportamento probabilístico no formato de uma sigmoide[2], o que é uma característica da regressão logística[3].

Para a curva abaixo sabemos que:

$$g(x) \to +\infty, P(Y = 1) \to 1$$

$$g(x) \to -\infty, P(Y = 1) \to 0$$

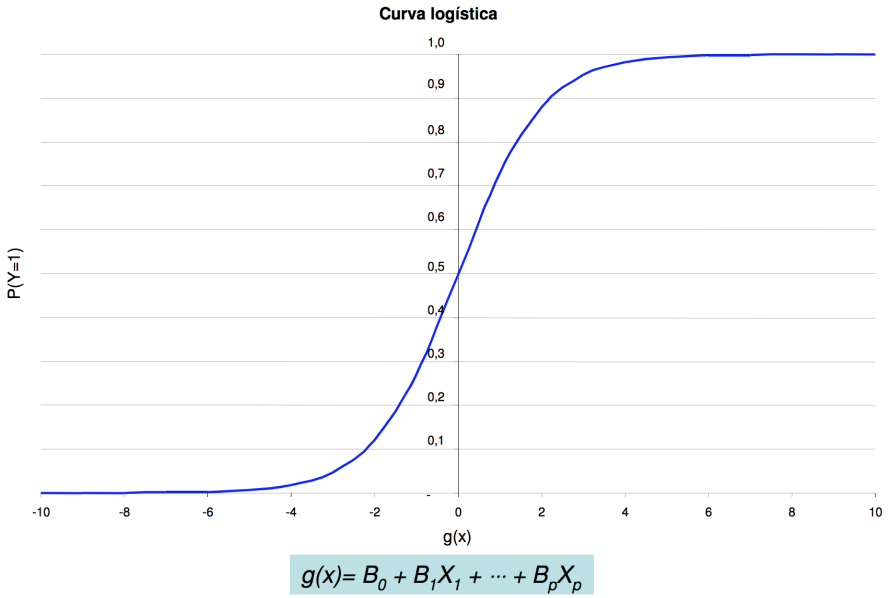

Figure 1: Curva de Regressão Logística em formato de sigmoide

Algoritmo

Tendo em vista o funcionamento da regressão logística passamos para geração de nosso algoritmo. Passo a Passo:

- Obtivemos/geramos nosso dataset através de uma API que fornece uma grande coleta de dados da bolsa[4].
- Transformamos o conjunto de dados para que o nossas "features" fossem as variações relativas do preço de fechamento de todas as ações que o usuário pedir, dentro de uma janela de tempo também definida pelo usuário.
- Dividimos os dados em dois conjuntos, um conjunto de teste e um conjunto de treinamento.
- Criamos um modelo treinando o classificador com o conjunto de treino.
- Testamos o classificador com o conjunto de teste.
- Repetimos o processo para diferentes janelas de tempo, calculando a cada uma delas a área embaixo da curva ROC.
- Construimos um gráfico com todas as áreas para permitir ao usuário tomar a melhor decisão possível.

Inicialmente esse classificador foi feito em um Ipython Notebook utilizando a biblioteca Pandas do Python e depois passou para o Apache Zeppelin pois como esta é uma ferramenta de big data seria possível analisar um conjunto de dados maiores e obter resultados mais precisos, através do que chamamos de RDD (*"Resilient Distributed Datasets*), os quais nos permitem dividir a carga entre diferentes máquinas rodando em paralelo. Isso foi feito através de um cluster criado na AWS Amazon.

Para analisar a qualidade do nosso modelo utilizamos a porcentagem de erro no conjunto de treino. Em seguida partimos para uma análise mais aprofundada, em que analisamos a área abaixo da curva ROC, um importante gráfico para nossa regressão. Porém, para melhor entender como foi está análise é necessário entender o conceito de matriz de confusão primeiro.

A curva ROC e Matriz de Confusão

Uma matriz de confusão é um resumo dos resultados de previsão em um problema de classificação. A matriz de confusão passa uma visão não apenas dos erros feitos pelo seu classificador, mas dos tipos de erros que estão sendo feitos[5].

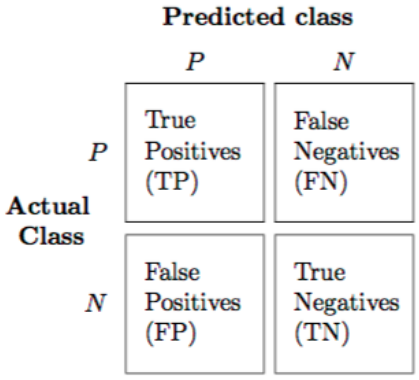

Figure 2: Matriz de Confusão

Os termos da matriz de confusão são:

- **True positives/Positivos verdadeiros(TP):** Esses são os casos em que a previsão é sim e o resultado real é sim.
- **True negatives/ Negativos verdadeiros(TN):** Esses são os casos em que a previsão é não e o resultado real é não.
- **False positives/ Falsos positivos (FP):** Esses são os casos em que a previsão é sim e o resultado real é não.
- **False negatives/Falsos negativos (FN):**Esses são os casos em que a previsão é não e o resultado real é sim.

Tendo em vista o funcionamento da matriz de confusão passamos para construir a curva ROC [6].

A curva ROC baseia-se na taxa de positivos verdadeiros (TPR) e na taxa de falsos positivos (FPR). Constrói-se um gráfico de coordenadas no qual a taxa TPR é associada ao eixo Y e FPR ao eixo X. Esse gráfico por sua vez, nos diz quão preciso é nosso modelo quando utilizamos um sistema de classificação binária. Esse sistema ocorre quando a saída do modelo é apenas sim ou não. Para isso, precisamos de um "*threshold*" que irá falar para o modelo a partir de que probabilidade ele deverá considerar sim.

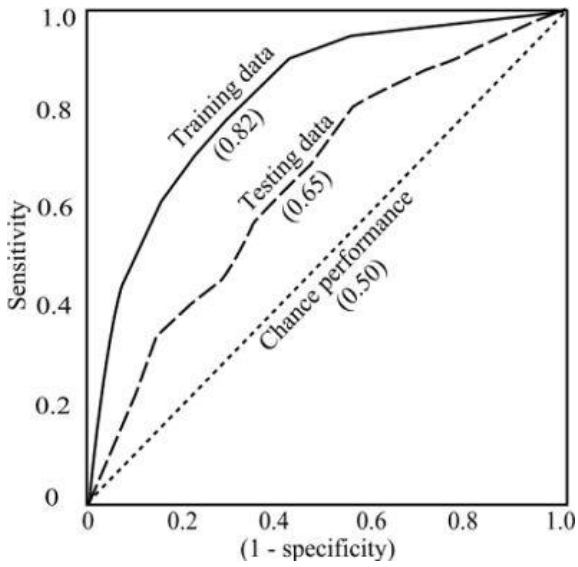

Figure 3: Exemplo de curva ROC. Eixo Y é de acordo com TPR e o eixo X é de acordo com FPR

No gráfico acima deve se considerar que:

- **Sensitivity:** a probabilidade de que um resultado de teste seja positivo quando o resultado real é positivo (taxa positiva verdadeira, expressa em porcentagem).

$$Sensitivity = \frac{TP}{(TP + FN)}$$

- **Specificity:** probabilidade de que um resultado de teste seja negativo quando o resultado for negativo (taxa negativa verdadeira, expressa em porcentagem).

$$Specificity = \frac{TN}{(FP + TN)}$$

Para pontos específicos do gráfico:

- O ponto (0,0) nunca classifica um exemplo como positivo.
- O ponto (1,1) ou (100%,100%) sempre classifica um novo exemplo como positivo.
- O ponto (0,1) ou (0%, 100%) representa o modelo perfeito, isto é, no qual não ocorrem erros na classificação.
- O ponto (1,0) ou (100%,0%) representa o modelo que sempre erra na classificação.
- Um modelo de classificação é representado por um ponto no gráfico ROC, calculado através da taxa de verdadeiros e falsos positivos (TPR e FPR) desse modelo a partir da sua matriz de confusão.

3 Resultados

Entendendo melhor o que significa a análise feita no modelo gerado passamos para os resultados de nosso modelo.

A primeira análise feita foi baseada apenas nos valores brutos de fechamento das ações. Com isso, obtivemos uma probabilidade de acerto em torno de 50%, tão boa quanto "jogar uma moeda". Apesar disso, conseguimos uma melhora significativa ao utilizar as variações relativas dos preços de fechamento. Com isso, conseguimos subir em média de cinco a seis por cento nosso acerto. Ainda assim, consideramos esse um valor muito baixo, ainda que muito promissor. Esta "precisão" do método comprova a teoria chamada de *Hipótese do mercado eficiente*[7], que diz que um analista não consegue prever o comportamento de ações pois esses comportamentos são randômicos, e como a motivação (movimentação alta de dinheiro) para buscar um meio de previsão é tão alta, se fosse possível, provavelmente já teria sido feito.

Uma análise mais aprofundada foi feita para diferentes quantidade de dias de intervalo, calculando para cada um deles a área abaixo da curva ROC para ver se com mais dias nos dados, a precisão do método seria melhor. Isso não se tornou verdade, visto que o comportamento desse gráfico parece aleatório.

Geramos um gráfico da área abaixo da curva ROC pelo tamanho em dias do intervalo utilizado no nosso modelo.

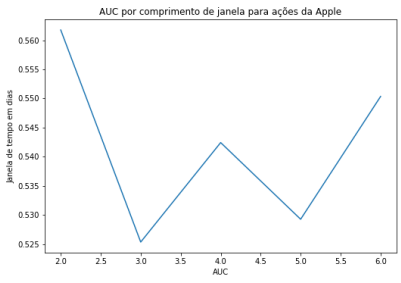

Figure 4: Gráfico para ação da Apple

O exemplo acima mostra este gráfico para a ação da empresa Apple.

4 Problemas Enfrentados

No decorrer do projeto foram enfrentados problemas que são importantes relatar.

4.1 Erro nas Datas

O primeiro problema enfrentado, que foi fundamental de ser corrigido, foi o fato da bolsa não abrir aos Sábados e Domingos. Sem esta percepção o código retornava datas repetidas e sem muito nexo. Com a correção deste erro a previsão passou a fazer sentido.

4.2 Clusterização AWS

Quando a quantidade de dados tornou-se grande demais para apenas uma máquina dar conta de todo processamento, tivemos que subir um cluster na AWS[8], para poder rodar mais instâncias que poderiam processar os dados em paralelo, fazendo-o de forma mais rápida. Entretanto ocorreram inúmero erros no cluster com o mesmo código que rodava em um computador local sem problemas algum. Este erro foi percebido que era causado pelo fato do cluster utilizar a versão do Python 2 ao invés do Python 3. Arrumando este erro foi possível rodar no Cluster.

5 Evolução para passos futuros

Existem diversos caminhos a serem tomados com o objetivo de melhorar o nosso modelo. O primeiro e mais claro seria de fato mudar como nós criamos de fato esse modelo. Regressão logística é apenas o começo de um mundo muito maior, com inúmeras técnicas de Machine Learning e redes neurais, passíveis de nos retornar resultados ainda mais promissores. Temos também uma das possibilidade de incorporar no nosso modelo, estudos sobre sentimento relativos a uma marca, o que acreditamos que pode trazer informações valiosas para nossa previsão.

Um passo que seria interessante para buscar melhores resultados seria implementar o classificador chamado de Random forests.

Random Forests

Uma random forest começa com uma técnica padrão de machine learning chamada de "árvore de decisão". Em uma árvore de decisão, uma entrada é inserida na parte superior e, ao atravessar a árvore, os dados são distribuídos em conjuntos menores e menores[9].

A random forest é um próximo nível que combina árvores com a noção de um conjunto. Assim as árvores são definidas como "weak lerarners"e as florestas como um "strong learners".

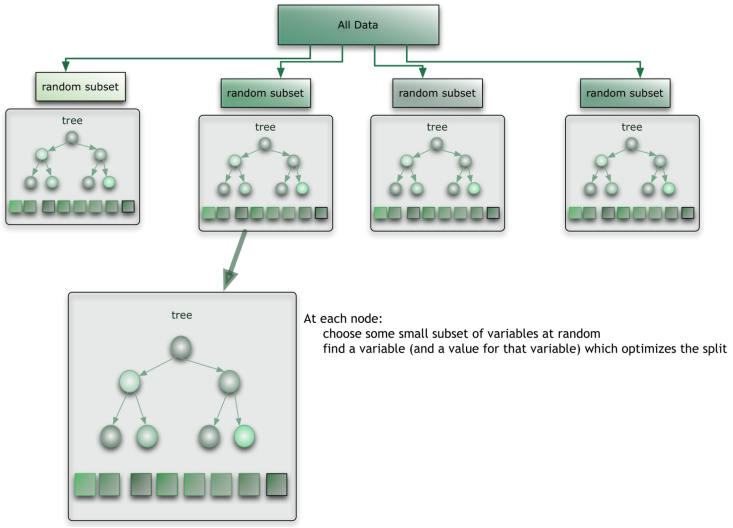

Figure 5: Funcionamento do Random Forest

Veja como esse sistema é treinado; para algumas árvores T:

- Uma amostra de N casos é aleatoriamente selecionada com substituição para criar um subconjunto dos dados (veja a camada superior da figura acima). O subconjunto deve ser cerca de 66% do conjunto total.
- Em cada nó:
- Para algum número m, m variáveis preditoras são selecionadas aleatoriamente entre o conjunto de todas as variáveis preditoras.
- A variável preditora que fornecer a melhor divisão, de acordo com alguma função objetiva, é usada para fazer uma divisão binária nesse nó.
- No próximo nó, escolha outras m variáveis aleatoriamente de todas as variáveis preditoras e faça o mesmo.
- Quando uma nova entrada é inserida no sistema, todas as árvores são executadas. O resultado pode ser uma média ou média ponderada de todos os nós terminais atingidos ou, no caso de variáveis categóricas, uma maioria de votos.

Observe que, com um grande número de preditores, o conjunto de preditores elegíveis será bastante diferente de nó para o nó, quanto maior a correlação inter-árvore, maior a taxa de erro aleatório da floresta, de modo que seria interessante ter as árvores não correlacionadas. À medida que m desce, a correlação entre árvores e a força das árvores individuais diminuem. Portanto, é necessário descobrir algum valor bom de m.

6 Conclusão

Construímos um modelo preditivo que utiliza a regressão logística para fazer previsões com uma grande quantidade de dados em um Cluster que roda em diferentes instâncias paralelizando os dados acelerando o processamento destes, entregando resultados mais rápidos. Os resultados provaram que não é possível fazer uma previsão de comportamento de ações da bolsa com uma alta taxa de acerto utilizando regressão logistica.

7 Github

https://github.com/rmolines/StocksGenie

References

[1] https://pt.wikipedia.org/wiki/Regress%C3%A3o_log%C3%ADstica, último acesso: 20/11/2017

[2] https://en.wikipedia.org/wiki/Sigmoid_function, último acesso: 20/11/2017

[3] HOSMER, David W.; LEMESHOW, Stanley. Second Edition. Ohio: Applied Logistic Regression, 1989.

[4] https://www.alphavantage.co/, último acesso: 21/11/2017

[5] https://machinelearningmastery.com/confusion-matrix-machine-learning/, último acesso: 21/11/2017

[6] REVISTABW. Curva ROC.Revista Brasileira de Web: Tecnologia. Disponível em http://www.revistabw.com.br/revistabw/curva-roc/. Criado em: 16/02/2015. Última atualização: 24/07/2015. Visitado em: 19/11/2017. (http://www.revistabw.com.br/revistabw/curva-roc/)

[7] http://www.nasdaq.com/article/investing-basics-what-is-the-efficient-market-hypothesis-and-what-are-its-shortcomings-cm530860, último acesso: 21/11/2017

[8] https://aws.amazon.com/pt/, último acesso: 21/11/2017

[9] http://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics, último acesso: 23/11/2017