

**Projeto 3**

**Ciência dos Dados**

Felipe Kocinas

Gustavo Berger

Rafael dos Santos

2 ENG A

**Sumário**

1. Pergunta  **3**
2. Introdução **3**
   1. O porquê dos Features **3**
   2. Organização do documento **4**
      1. *“Passos iniciais”* **4**
      2. *“Linear regression”*  **5**
      3. *“Random Forest”* **5**
3. Desenvolvimento **5**
   1. Processo **5**
   2. Metodologia **6**
4. Resultado **7**
5. Conclusão **8**
6. Referências **9**

**Pergunta:**

Como é afetado o rating de um produto vendido por e-commerce?

**Introdução**

O setor de e-commerce no Brasil vem sendo cada vez mais explorado, tem apresentado um enorme crescimento e uma representatividade cada vez maior. Ainda assim muitos anunciantes sentem dificuldades para elaborar seus anúncios de forma que suas vendas sejam maximizadas. Tendo isso em vista, para o projeto 3 de ciência dos dados escolhemos fazer uma regressão para prever o rating que um produto recebe com base em diversas características. Esse tipo de análise seria de suma importância para vendedores podendo assim melhorar suas vendas, dando ao projeto um tom similar à uma análise de marketing.

Para tentar responder à pergunta proposta, escolhemos analisar dentre os features disponíveis as seguintes opções:

**O porquê das features escolhidas:**

* Estado de residência consumidor ( Custumer\_State\_[UF do estado] );
  + Acreditamos que uma possível cultura de cada estado pode afetar o rigor com que a população enxerga os produtos de e-commerce.
* Valor do frete;
  + Uma cobrança excessiva de taxa de entrega pode fazer com que o cliente torne-se mais crítico ao produto
* Comprimento da descrição do produto no anúncio;
  + Um anúncio com uma descrição extensa demais pode fazer com que o cliente não preste atenção em todos os detalhes, assim, ficando insatisfeito com algum aspecto que não se atentou. O mesmo processo pode ocorrer para uma descrição curta demais, já que informações importantes sobre o produto podem ser omitidas, surpreendendo negativamente o cliente
* Quantidade de fotos no anúncio;
  + Uma baixa quantidade de fotos pode fazer com que o cliente se surpreenda negativamente com relação aos aspectos estéticos de seu produto
* Comentários sobre o produto;
  + Os comentários são o reflexo direto da nota que foi dada ao produto, já que um cliente que faz um comentário negativo tende a dar um rating baixo
* Tempo de demora entre a compra e entrega do produto.
  + Um tempo de entrega muito longo pode fazer com que o cliente perca a paciência, o deixando mais crítico ao produto

Consideramos que todos esses fatores e finalidades escolhidas para o projeto deveriam dar conta de responder nosso questionamento de forma bastante completa.

**Organização do documento:**

Para uma maior organização do documento no jupyter, julgamos adequado sua divisão em 3 arquivos diferentes, sendo eles:

**Documento “Passos iniciais”**

O arquivo “classificador de comentários”apresenta alguns pontos principais no código:

* A utilização da estratégia sklearn.pipeline para a classificação do feature “*review\_comment\_message*”. Essa metodologia será explicada mais adiante (na parte do desenvolvimento).
* Neste arquivo está presente nossa análise exploratória, que será futuramente discutida na parte de resultados.
* Também há o cálculo da variação de dias entre o pedido e a entrega do produto, o que será discutido na seção de desenvolvimento
* Por fim, nesse código também está presente uma função de dummification de colunas, que foi utilizada para separar em novas features cada estado do Brasil, os quais anteriormente estavam contidos somente em uma coluna

**Documento “Linear Regression”**

Como o próprio nome já diz, neste arquivo está localizado o programa de regressão linear. Ele foi feito com base na função OLS da biblioteca Statsmodel que, no nosso código está disposta na def regress(X,Y). Mais adiante nesse documento, temos os resultados gerados com a regressão linear, que serão discutidos na seção de comentários

**Documento “Random Forest”**

Assim como o ultimo documento, este tem seu nome bastante atrelado ao conteúdo. Dessa forma, contém o programa que faz o Random Forest Regressor, além de apresentar os resultados dessa regressão de forma gráfica e comparativa com os dados de teste.

Tendo todas essas características introdutórias explicadas, o entendimento do desenvolvimento do software é essencial para a continuidade da linha de raciocinio e compreensão de nosso processo de construção do projeto

**Desenvolvimento**

**Processo**

Primeiramente fizemos a análise exploratória, gerando matrizes de dispersão. Para realização de tal, utilizamos a biblioteca seaborn (com o commando sns.pairplot). A matriz e suas conclusões serão interpretadas na seção de resultados.

Feito isso, escrevemos um código para a classificação inicial dos comentários dos reviews, o que é essencial para a separação dos estados entre positivo, negativo e neutro dentro da base de treinamento.

Após essa classificação, começamos a fazer o cálculo do tempo (em dias) da demora entre a compra de um produto e de sua entrega na casa do cliente. Isso foi feito afim de deixar o feature **tempo de demora entre a compra e a entrega do produto** em um número inteiro, já que antes dessa manipulação tinhamos duas colunas sendo uma da data de entrega e outra da data de compra. Estas se encontravam no formato "2017-01-31 17:19:01.0000". Para fazer isso, utilizamos a função do pandas chamada pd.to\_datetime. Após a transformação das colunas foi feita uma nova cujos dados representavam a subtração entre a data da chegada e a data de entrega.

Depois dessa manipulação da data de entrega foi escrita a função para a dummification de colunas que, como dito anteriormente, foram utilizadas para a criação de novas colunas com cada estado brasileiro separado. Isto tudo foi integrado em um novo database em que retiramos todas as features extras oferecidas, isto é, todos os dados que vieram na planilha porém que não usariamos. Feito isso separamos aleatoriamente nosos dados numa divisão de 60% treinamento e 40% teste. A partir disso, usamos somente os 60% correspondentes ao treinamento para todas as mudanças feitas.

O programa de pipeline foi escrito logo em seguida. Para isso foi necessário utilizar a classificação dos comentários dos usuários em uma escala numérica. Sendo assim possível calcular as probabilidades destes serem negativos(neg\_prob), neutros(neu\_prob) ou positivos(pos\_prob). Com esse objetivo utilizamos o código sklearn.pipeline Este funciona misturando as funções *fit, transform* e *predict*. Assim, transformando os dados durante uma etapa chamada vetorização e prevendo-os quando são passados ao classificador naive-bayes, entre cada uma dessas etapas o commando fit é utilizado.

A implementação dos modelos de regressão linear e do random forest (para que fosse possível prever as notas dos clientes em relação aos produtos) ocorrem logo em seguida, assim como seus testes e resultados, cujas conclusões ocorrerão no futuro.

Por último fizemos um comparativo entre as notas que nosso método previu e as da realizade.

**Metodologia**

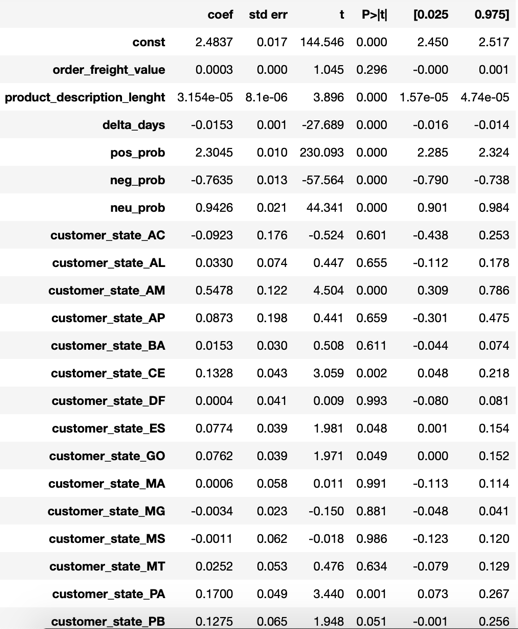
Para a regressão testamos 2 modelos diferentes, comparando os resultados e escolhendo aquele com maior acurácia. A regressão linear (primeiro método utilizado) é um mecanismo em que o programa gera uma reta que mais apróxima os dados selecionados, assim minimizando o erro dentro do possível.

O segundo método testado foi o Random Forest regressor. Primeiramente, para entender este conceito é necessário uma explicação da ideia de arvore de decisões. Este método separa os dados em pequenos intervalos, ajustando uma reta para cada delta, porém, este modelo não consegue levar em conta outliers, sendo muito alterado por cada um desses pontos extremos. Dessa forma, o Random forest regressor é utilizado como uma alternativa que consegue suprir esse erro da arvore de decisões. O recurso faz isso dividindo o data set em diversos conjuntos e aplicando a árvore de decisões em cada um desses conjuntos, após isso é realizada uma média entre todas as curvas. Isso faz com que, caso um outliner apareça, o erro que este causa seja minimizado

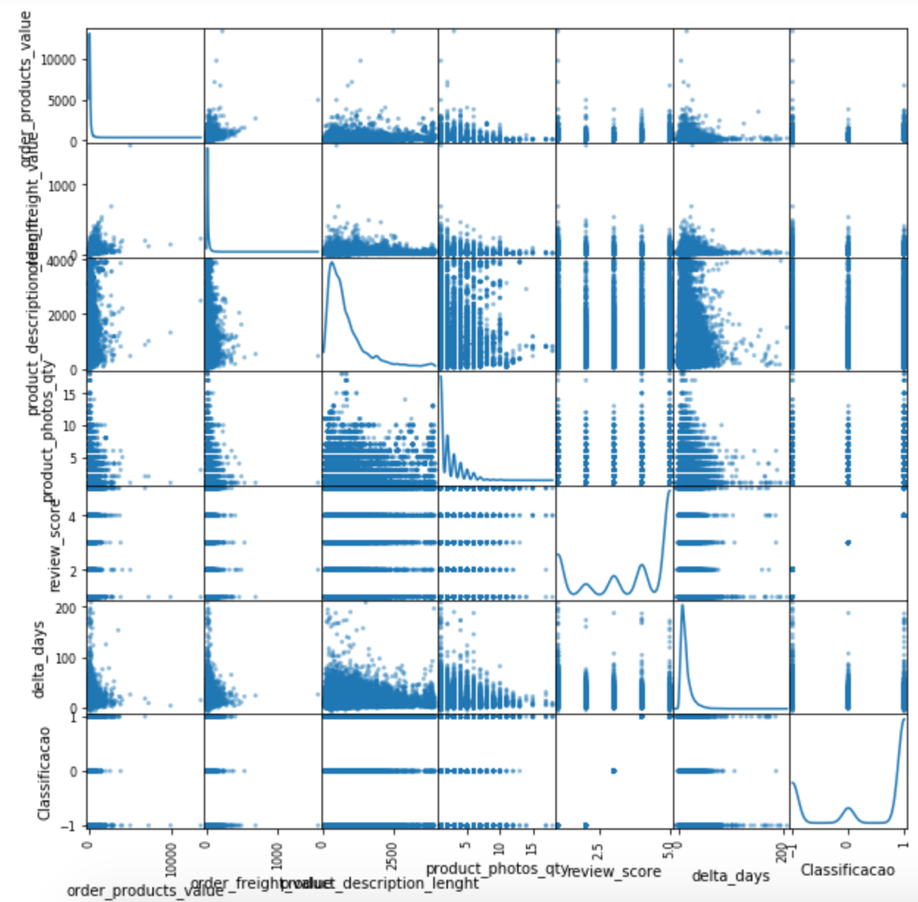
Dessa forma, como já era esperado, a precisão do Random Forest Regressor foi maior que a da Regressão Linear. Porém, surpreendentemente estas precisões foram bastante próximas.

**Resultado**

Nossa hipótese nula era de que nossos features seriam desnecessários para a tese. Porém, pelos baixos valores dos Pvalues das variáveis de valor do frete, comprimento da descrição do anúncio, o delta de dias, e os comentários das reviews, rejeitam esta hipótese. Ainda assim, os Pvalues do estado de residencia do consumidor (para a maior parte dos estados) e a quantidade de fotos em um anúncio são consideravelmente elevados, como Podemos observer abaixo:



Além disso, é de suma importância discutir sobre a similaridade entre os valores de R quadrado dos modelos aplicados (regressão linear e random forest). Isto ocorre principalmente porque alguns features escolhidos tem um alinhamento linear. Isso faz com que tanto o Random Forest quanto a regressão linear se encaixem bem aos dados, como Podemos observer na matriz de dispersão.



**Conclusão**

Tendo em vista todos os resultados obtidos, podemos concluir que a acurácia alta na resposta da pergunta é bastatante difícil, já que os features disponíveis não são tão correlacionados com o rating do produto quanto esperávamos, isso se dá por alguns motivos, sendo os principais deles:

* O Estado de residência do consumidor ( Custumer\_State\_[UF do estado] ); demonstrou-se, em muitos estados como extremamente desnecessário. Isso pode ser comprovado pela alta penetração do setor em alguns estados, já que os compradores já estão completamente acostumados com o sistema de e-commerce, dessa forma, não se incomodam com aspectos que novatos se chateiam
* O valor do frete pode acabar não influenciando tanto na opinião do cliente sobre um produto em específico, mas sim na opinião geral sobre o site de vendas
* O comprimento da descrição no anúncio acaba influenciando nos casos de um rating negativo, porém quando a nota é positiva, o fator não ajuda na sua determinação
* A quantidade de fotos, assim como o comprimento da descrição do anúncio, pode não afetar na definição de notas positivas
* Os comentários sobre o produto são bastante definitivos na hora de prever o rating de um produto por motivos já citados na introdução. Dessa forma, esse feature é de extrema importância para nossa regressão
* O tempo de demora entre a compra e a entrega do produto, assim omo o valor do frete pode não influenciar totalmente na opinião do produto mas sim do site ou da distribuidora, já que essas empresas são as principais influenciadoras na entrega do produto.

Assim, tendo em vista todos os aspectos que nossos features poderiam influenciar positivamente (descritos na introdução) e negativamente (descritos acima) na acurácia da nossa regressão, podemos concluir que o R quadrado de aproximadamente 0,7 é condizente com a realidade dos fatores escolhidos. Dessa forma podemos concluir que nosso código de regressão atingiu a maior precisão possível dentro das condições dadas e que, portanto, não é viável prever com acurácia extrema a opinião do cliente com relação a um produto com os dados oferecidos.

**Referências**

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#sklearn.ensemble.RandomForestRegressor>