NYC Airbnb Open Data: Exploração dos dados e Previsão do preço dos aluguéis

1st Caio Martim Barros
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
caiobarros@alu.ufc.br

2nd Lucas Vitoriano
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
lucasvitoriano25@gmail.com

3rd Mario Cesar F. D. Filho
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
mariocesarfreire@alu.ufc.br

***Abstract*—Airbnb é uma empresa operante no mercado online de hospdagens, de forma a permitir cadastro e busca de casas e apartamentos por meio da listagem e validação dos dados, proporcionando uma maior segurança e gerando um crescimento na sua busca dentro desse setor devido a inovação. Por conta de sua ascensão, é possível recolher dados sem muitas complicações e assim fazer estudos amplos, realizando uma análise exploratória desses dados a fim de entender mais sobre suas aplicações e criar materiais para uma abordagem mais específica utilizando esses dados disponíveis. Nesse sentido, nosso propósito para este artigo é, a partir da utilização de regressões lineares, fazer previsões utilizando as informações do *dataset* para estabelecer o valor do preço dos aluguéis dos imóveis e assim comparar com o valor real com intuito de entender se os valores cobrados são compreensíveis. Nesse âmbito, foram-se feitas previsões sobre os valores dos aluguéis com uma margem de erro média de 1.26 dólares sobre o valor real. Ademais, é realizada uma análise exploratória dos dados, efetuando gráficos e tabelas para uma abordagem aprimorada da situação do dataset.**

***Index Terms*—Airbnb, dataset, análise, previsão, regressão linear.**

I. Introdução

No contexto de grande volume de dados, a Airbnb é uma empresa em que se utiliza bastante como objeto de estudo e análise por conta da alta disponibilidade dos dados provenientes, tornando-o, assim, uma grande percussora no âmbito da Inteligência Artificial e do Processamento de Dados. Com a utilização da linguagem Python e do *dataset* "New York City Airbnb Open Data" [1] foi possível classificar o tipo de aluguel com o uso de técnicas de modelo preditivo do tipo de classificação. Além disso realizamos operações para identificação de padrões e comportamento dos dados colunares por meio da utilização de gráficos, como *scatter* e *biplot*. Dessa forma, é possível fazer análises como: quais regiões de Nova Iorque estão mais suscetíveis a usar esse tipo de tecnologia e quais são os principais tipos de aluguel por área.

Com os dados disponivéis, foi necessário realizar uma conversão entre dados categóricos e dados numéricos em três colunas do *dataset* [1], que foram *"neighbourhood_group"*, *"neighbourhood"* e *"room_type"*. Para esse propósito, foi-se utilizado bibliotecas do python.

No que se diz a respeito das regressões lineares utilizadas para a criação das previsões dos aluguéis dos imóveis de Nova Iorque, pode-se destacar a utilização da regressão linear simples, Regressão Linear Penalizada *(PLR)*, além das Regressão por Mínimos Quadrados Parciais *(PLS)* e Regressão do Componente Principal (PCR). Dentre essas destacadas, foram feitas operações de comparação para identificar qual regressão se encaixa melhor no contexto e também foi utilizado as previsões do preço do aluguel por meio do RMSE e R^2. Nesse raciocínio, foi empregada o *k-fold cross validation* nas regressões com intuito de melhoria do modelo.

II. Análise dos dados

Nesta seção vamos abordar os dados que estamos trabalhando, explicar as variáveis que vamos considerar nas nossas regressões, técnicas que utilizamos para averiguar os dados e as transformações que neles fazeremos.

A. Conhecimento necessário

Nesse trabalho focamos em analisar os dados principalmente com foco no sua média, skewness e desvio padrão.

O desvio padrão indica se os valores de um preditor estão próximos ou muito espaçados, um alto valor de desvio padrão pode tornar um modelo impraticável, pois ele tentará abranger todos os valores, podendo gerar um overfitting. Para calcular o desvio padrão utilizamos a fórmula 1:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu_x)^2}{n-1}} \tag{1}$$

Explicando os termos na fórmula 1 temos em n o número total de componentes, μ_x a média dos componentes e x_i o componente atual

Outro valor importante é o Skewness, skewness é uma forma de analisar se o preditor está centralizado, se moda e a media estão na mesma posição, caso não esteja, métodos como PLS são afetados negativamente [1] Para calcular a skewness utilizamos a fórmula 2

$$Skewness = \frac{\sum(x_i - \mu_x)^3}{(n-1)s^3} \tag{2}$$

Sendo na fórmula 2, x_1 o componentes atual, μ_x a média dos componentes e s o desvio padrão

Para descobrir qual a melhor regressão comparamos 5 tipos de modelos, regressão linear, regressão de ridge, regressão de lasso, PCR e PLS. A seguir faremos uma breve explicação sobre cada modelo.

Regressão linear procura uma forma linear de retratar os dados de forma a ter o menor erro quadrático, a expressão da regressão possui fórmula 3:

$$Regressão = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n \quad (3)$$

Onde cada b_n é o beta que define a importância daquele preditor na definção da reta e n é a quantidade de observações.

A regressão linear simples possui como medida de sucesso o menor erro quadrático,de fórmula :

$$MSE = \frac{1}{n}\sum_{n=1}^{n}(y_i - \hat{y}_i)^2 \quad (4)$$

Onde Y_1 é o termo atual e n é o número de termos totais.

Já a regressão de ridge adiciona um outro elemento a fórmula de erro quadrático citada em 4, chegando a fórmula 5:

$$SSE_{L_2} = \sum_{n=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{P}\beta_j^2 \quad (5)$$

Na regressão de ridge temos que o P é o número de colunas e o lambda é uma forma de penalizar a bias em pró de obter uma menor variância no final, assim pode-se reduzir alguns preditores a valores próximos de 0, mas nunca iguais a ele.

Já no modelo de lasso temos uma regressão similar a de ridge, penalizada, mas com uma alteração na fórmula do SSE.

$$SSE_{L_1} = \sum_{n=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{P}|\beta_j| \quad (6)$$

Com essa alteração temos um modelo muito parecido com o de ridge, mas com a diferença que alguns preditores podem ser zerados.

Outro modelo que utilizamos foi o PCR, que diferente dos modelos anteriores ele utiliza primeiro uma PCA nos dados, para reduzir suas dimensões. Com o novo dataset após a redução é então realizado uma regressão linear para encontrar o coeficientes da reta.

O último modelo de regressão que utilizaremos é o PLS, que é um método supervisionado, a variável de saída é considerada no pré processo dos dados[2].

PLS é um método de regressão que primeiramente utiliza uma redução de dimensão, assim como o PCR, ele recebe os dados originais, reduz a dimensão dos preditores, transformando-os em novas colunas X1',X2',...,X'n, e em seguida realiza a regressão usando como método os mínimos quadrados unido com uma penalidade, para achar a regressão com menor erro.

B. Matriz Airbnb

No nosso dataset trabalharemos com uma matriz 48895 linhas x 11 colunas, que, após transformarmos os dados categóricos em numéricos, atribuindo um valor para cada string, resultando nas seguintes colunas:

Tabela I: Média desvio e skewness antes da normalização

Variável	Descrição	Tipo
Neighbourhood Group	Nome da Vizinhança	Int
Neighbourhood	Bairro	Int
Latitude	Latitude	Int
Longitude	Longitude	Int
Room type	Tipo de Quarto	Int
Price	Preço	Float
Minimum nights	Noites mínimas	int
Number of Reviews	Número de Reviews	Int
Reviews Per Month	Réviews por mês	Int
Calculated Host Listing Count	Número de Imóveis por Locador	Int
Availability 365	Avaliabilidade por ano	Int

C. Padronização dos dados

Fazendo primeiramente uma análise no preditor price, que será o principal valor do nosso trabalho, vemos que ele possui um skewness muito positivo, o que pode levar a divergências no modelo de regressão linear que usaremos.

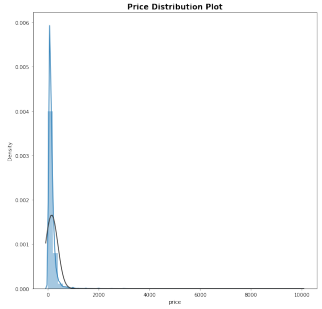

Figura 1: Preditor Price com um alto skewness

Como vemos na Figura 1, com tendência explicada pela linha, é necessário centralizar os dados, diminuir o skewness.Assim, a técnica de transformar a unidade para logarítma foi aplicada, tornando os dados mais simétricos, o que diminuirá o erro na saída do modelo preditivo.

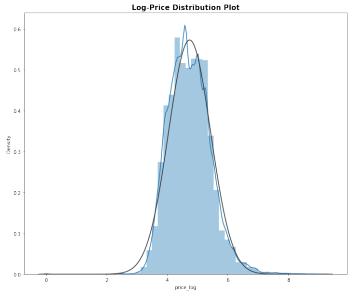

Figura 2: Preditor Price na base log e

Para as outras variáveis calculamos o desvio padrão skewness e média, para que possamos fazer uma análise mais

detalhada do dataframe que estamos trabalhando, oque gerou a tabela II.

Tabela II: Média desvio e skewness antes da normalização

variável	média	desvio padrão	skewness
latitude	40.728949	0.054530	0.237167
longitude	-73.952170	0.046157	1.284210
número_reviews	23.274466	44.550582	3.690635
reviews_por_mês	1.373221	1.497775	3.511906
grupo_bairro	1.675345	0.735816	0.373464
bairro	107.122732	68.743096	0.256005
tipo_quarto	0.504060	0.545379	0.422704
Residência por host	7.143982	32.952519	7.933174
Disponibilidade	112.781327	131.622289	0.763408
price_log	4.736885	0.695344	0.553105
noites_mínimas	7.029962	20.510550	21.827275

Vemos que em certos preditores o desvio padrão e a média estão com valores muito altos, valores esses que precisam ser padronizados para ter média igual a 0 e desvio padrão igual a 1, assim quando fizermos operações como a análise de componente principal teremos maior precisão nos resultados. Para realizarmos essa normalização, utilizamos a função *standardscaler*, com esses dados trabalhados, obtivemos os seguintes novos valores:

Tabela III: Média desvio e skewness depois da normalização

variável	média	desvio padrão	skewness
latitude	3.949141e-14	1.000010e+00	0.237167
longitude	662286e-13	1.000010e+00	1.284210
número_reviews	-2.315687e-14	1.000010e+00	3.690635
reviews_por_mês	-3.349251e-15	1.000010e+00	3.511906
grupo_bairro	-8.991399e-15	1.000010e+00	0.373464
bairro	5.305490e-16	1.000010e+00	0.256005
tipo_quarto	7.635292e-15	1.000010e+00	0.422704
Residência por host	3.545441e-14	1.000010e+00	7.933174
Disponibilidade	1.387977e-14	1.000010e+00	0.763408
price_log	2.582160e-15	1.000010e+00	0.553105
noites_mínimas	-7.665145e-16	1.000010e+00	21.827275

Como vemos na tabela III, agora nossos preditores estão normalizados. Entretanto, notamos que o skewness de longitude, numero reviews, reviews por mês, residência por host e noites mínimas são positivos, representando que a moda é menor que a média, o que é um valor aceitável, pois, com exceção de longitude, que é uma característica da região de Nova York , as variáveis estão sujeitas a ações humanas, o que pode gerar outliers, deslocando a média para a direita, mas, como esses outliers são dados importantes para considerar na nossa análise, optamos por não tirarmos.

D. Análise bi-variada

Como nós estamos tratando com mais de 10 variáveis, seria muito custoso abordamos todas as relações entre as variáveis aqui, portanto, optamos por mostrar ao leitor apenas as 4 variáveis que possuem mais influência no nosso output, o preço. Para obtermos essa ordem de influência pegaremos os maiores valores obtidos em módulo na tabela de correlação.

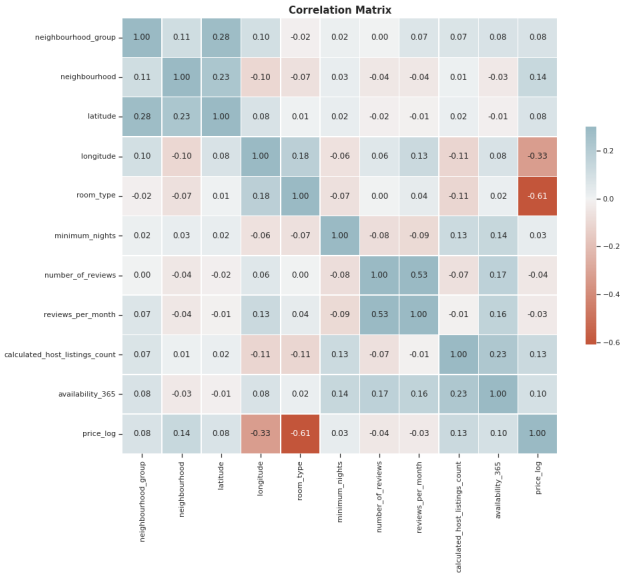

Figura 3: Matriz de correlação entre todos os preditores

Observando a tabela 3, notamos que existem duas relações moderadas, tipo_quarto e longitude, o resto das relações são fracas, sendo vizinhança e residências por host as duas maiores desse grupo. Assim, essas quatro são os que tem as maiores correlações em módulo com price_log, para mostrarmos mais detalhes essas relações traremos o scatterplot 4.

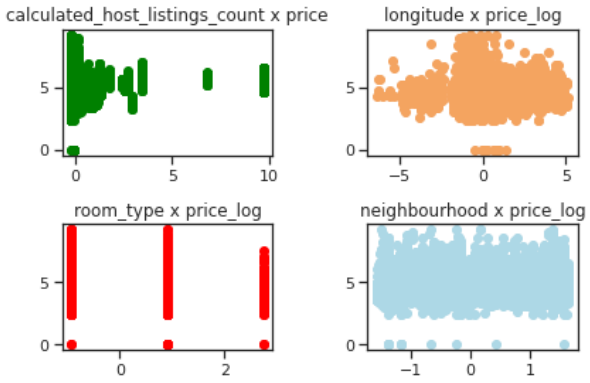

Figura 4: ScatterPlot's preço x preditores

Dentre os plots que obtivemos conseguimos retirar informações importantes sobre cada um dos 4 preditores mais correlacionados com o preço.

Os valores de longitude intermediários alcançam uma média maior comparada aos valores de longitude das extremidades, o que é explicado pela própria geografia de Nova York, onde a região central possui diversos pontos turísticos como o central park e a times square.

Para tipo de quarto (room_type) e neighbourhood podemos realizar uma análise mais detalhada se mudarmos o plot para um do tipo de barra.

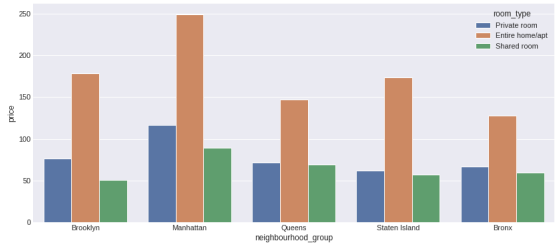

Figura 5: Preço por quarto e bairro

Com a nova figura 5, conseguimos visualizar que os alugueis do tipo entire room/apt são aqueles com maior média e aqueles do tipo shared room os com menor média. Além disso também conseguimos visualizar que os alugueis locados em Manhattam são os mais caros de todos os 5.

Com os outros preditores não conseguimos ter realizar uma boa análise somente com o plot preditor x price, o que pode resultar em um péssimo fit da regressão, perdendo precisão.

E. Análise da PCA

Como vimos na última sessão, os dados que estamos tratando possuem diversas variáveis, assim, torna-se útil realizarmos uma PCA, para extrair as informações mais importantes da tabela [5], reduzir a dimensão, usar menos variáveis e manter uma variedade de dados suficente para podermos gerar uma boa regressão.

Realizando o Principal Component Analysis com a restrição de dimensão n = 2 chegamos a dois autos valores. Para checarmos se 2 componentes principais é o suficiente para termos uma boa variância calculamos tabém a variância associada a cada componente, chegando a tabela IV:

Tabela IV: Autovalores e variância associada

autovalores	variância
1.74048852	0.16601841
1.55728554	0.14854339

Como vemos, apenas dois componentes não é o suficiente para representar bem os dados, assim, vamos ilustrar melhor a variância associada a cada componente com o uso do scree plot 6.

Figura 6: Variância por número de componentes

Exemplificado pelo gráfico 6, para alcançar uma variância de aproximadamente 60% seria necessário ao menos 5 componentes, mostrando assim que nossos dados não possuem uma boa separação.

Outra forma de vermos que nossos dados não são bem separados é com o uso do scatter plot 7.

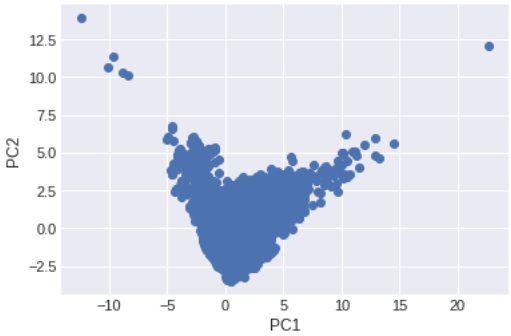

Figura 7: ScatterPlot entre PC1 e PC2

Como os dados estão muito próximos vemos que apenas duas dimensões não é o suficente para bem representa-los pois temos muita sobreprosição de dados entre PC1 e PC2, principalmente na regiao x $\approx$ 0 e y $\approx$ -1.

III. Resultados

Nesta seção serão discutidos os valores encontrados dos tipos de quarto do *dataset* por meio da utilização de modelos de classificação, os quais serão descritos ao decorrer do tópico, além de comparações pertinentes entre os mesmos e realização de gráficos e tabelas para melhor visualização das diferenças entre os classificadores utilizados.

A. Preparação dos dados para utilização dos modelos de classificação

Nesse caso, foi-se utilizada as 10 colunas de preditores mais a coluna do *log_price*, como já mencionado em seções anteriores. Nesse dataset, foi necessário fazer uma limpeza dos dados pois nosso previsor é uma coluna que contém três tipos de quartos: ”Private Room”, ”Shared Room” e ”Entire home/apt” e como queremos realizar a classificação desses tipos de quarto, queremos somente 2 tipos pois é uma necessidade do algoritmo uma operação que contenha somente tipos de dados 0 ou 1. Dessa forma, fazendo a contagem de cada um dos 3 tipos, foi-se analisado que o ”Shared Room” contém um valor relativamente menor do que os outros dois e então ele foi descartado do nosso dataset.

Dessa forma, além dessa manipulação de limpeza, foi-se utilizada *LabelEncoder*, da biblioteca *sklearn*, para transformar três colunas de dados categóricos em numéricos e, após isso, realizou-se uma normalização dos dados por meio *StandardScaler()*, restando, assim, 47735 linhas de dados em 11 colunas, as quais foram divididas entre 70% para treino e 30% para teste, utilizando o método da biblioteca *sklearn* chamada *train_test_split()*.

Tabela V: Resultados obtidos na Regressão Logística

	Resultados
Accuracy	0.8350
Precision	0.8258
Recall	0.8241

B. Estudo utilizando a Regressão Logística (LR)

Pela tabela V, percebe-se os resultados eminente da Regressão Logística, onde se obteve valores relativamentes bons, com ênfase na acurácia, em que se teve cerca de 83.5%.

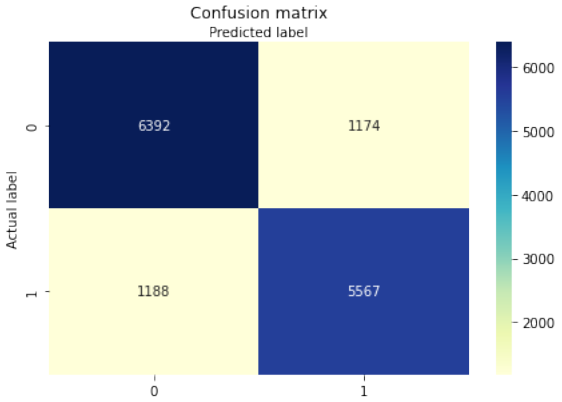

Figura 8: Matriz de Confusão para a Regressão Logística

Pela matriz de confusão 8, percebe-se que se obteve valores verdadeiros nas quantidades 6382 e 5567, o que representa que o algoritmo acertou 6382 dos 7566 totais existentes para ”Entire Room/apt” e 5567 de 6.755 para ”Private Room”. Logo, a fração condizente ao valor da precisão é dada pela quantidade de acertos sobre o total, em que temos $\frac{11.959}{14321} = 0.8350$, o que condiz com o valor encontrado na tabela V.

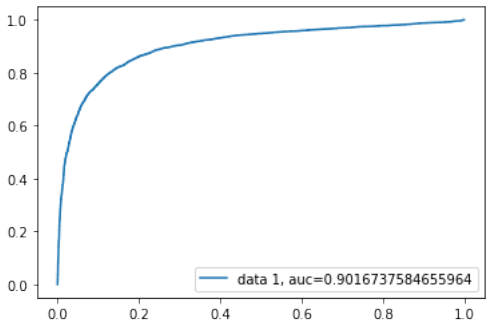

Figura 9: ROC para a Regressão Logística

A curva ROC, também conhecida como curva de característica de operação relativa, é resultado da operação de duas características: RPV (sensibilidade) e RPF (1 - especificidade), onde RPV = Positivos Verdadeiros / Positivos Totais versus RPF = Positivos Falsos / Negativos Totais, valores esses encontrados em uma matriz de confusão, por exemplo 8.

Dessa forma, a curva ROC é uma representação gráfica que ilustra o desempenho de um sistema classificador binário, como a Regressão Logística, à medida que o seu limiar de discriminação varia. Ela, portanto, auxilia na seleção de modelos possivelmente ideais, independentemente do contexto de custos ou distribuição de classe.

C. Estudo sobre a Ánalise Discriminante Quadrática (QDA)

Tabela VI: Resultados obtidos nas duas versões do QDA

	QDA (padrão)	**QDA (melhorado)**
Accuracy	0.6829	0.7172
Precision	0.6059	0.6378
Recall	0.9370	0.9267

Nesse tópico, será utilizada o classificador QDA, *Quadratic Discriminant Analysis*, em dois formatos: padrão e melhorado, as quais serão analisadas seus resultados, matrizes de confusão e ROC. Dessa maneira, é possível analisar brevemente os resultados obtidos no QDA padrão pela tabela VI, em que se obteve valores bem abaixo do que o primeiro classificador testado, que foi a Regressão Logística, em que se teve aproximadamente 68.3% para o QDA padrão e 71.7% para a versão melhorada.

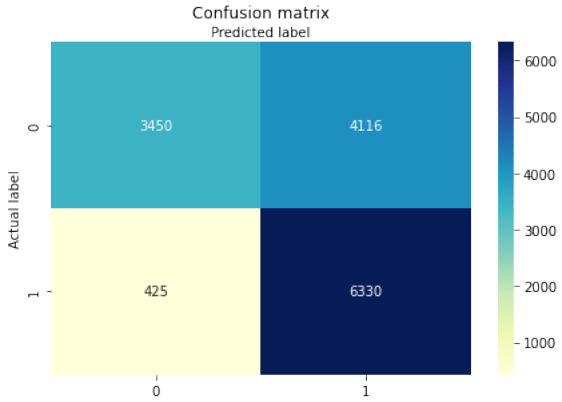

Figura 10: Matriz de Confusão para a QDA sem melhorias

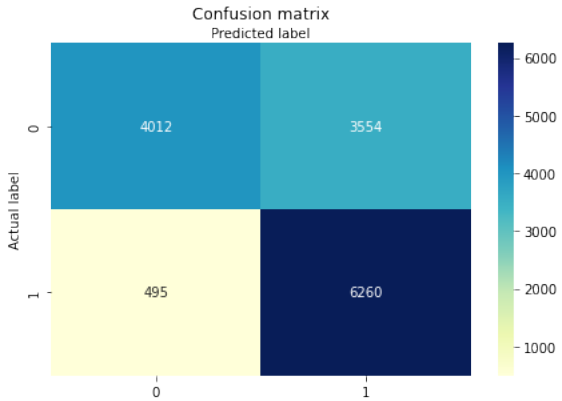

Figura 11: Matriz de Confusão para a QDA com melhorias

Como nas figuras 10 e 11 e na tabela VI, é possível destacar as diferenças de valores entre as duas versões do QDA, isso se dá pelo fato do QDA melhorado utilizar parâmetros melhores do que a versão normal e percebeu-se essa necessidade de melhoria do modelo por conta dos valores encontrados para a versão padrão desse classificador. Desse modo, para essa melhoria, foi-se utilizada a biblioteca do Python *GridSearchCV*, com um *cross-validation* de 5, em que realiza diversos "loops" no algoritmo a fim de procurar os melhores parâmetros para os quais foram colocados para teste, obtendo, assim, um valor melhor em comparação ao modelo padrão.

Nesse raciocínio, pelas matrizes de confusão de ambas versões 10 e 11, percebe-se que foram encontradas valores positivos para o binário 0 em 3450 casos para o modelo padrão e 4012 para o modelo melhorado e para o binário 1 em 6330 e 6260, respectivamente. Desas forma, em fração, percebe-se uma diferença de valores para os dois modelos em: $\frac{9780}{14321}$ $\frac{10.272}{14321}$, o que reflete em uma acurácia melhor para o modelo QDA melhorado, pois foram encontrados mais valores positivos, cerca de 492 a mais.

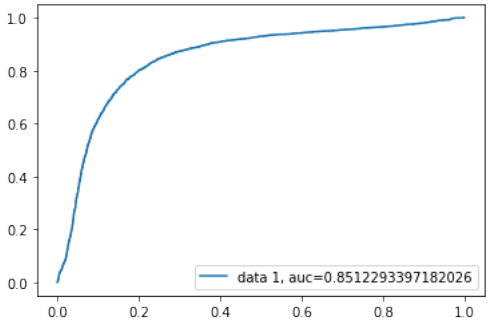

Figura 12: ROC para a QDA sem melhorias

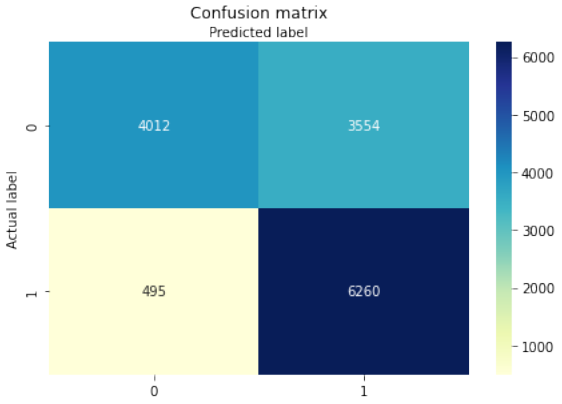

Figura 13: ROC para a QDA com melhorias

D. Estudo sobre o n-vizinho mais próximo (kNN)

Comparado aos outros dois modelos expostos anteriormente, o algoritmo kNN se aparenta promissor nesse quesito, pois pelos resultados da tabela VII e pela matriz de confusão 14, é possível analisar que se tem valor de acurácia de 84.2%,

Tabela VII: Resultados obtidos no kNN

	Resultados
Accuracy	0.8482
Precision	0.8604
Recall	0.8096

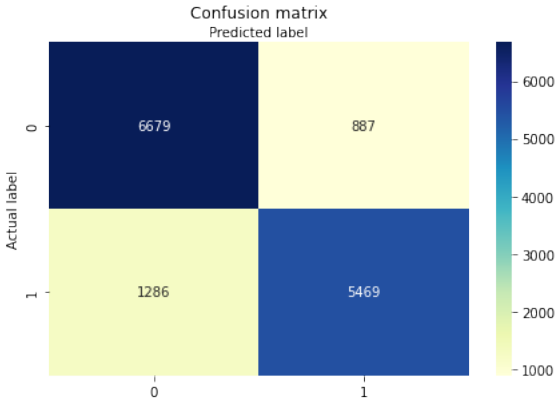

Figura 14: Matriz de Confusão para o kNN

que é o maior entre eles, e um acerto de maiores valores positivos encontrado, em que se tem $\frac{12.148}{14321}$.

Em relação a esse modelo de classificação, kNN, obteve-se valores melhores após uma série de tentativas manuais, onde o parâmetro necessário para rodar o algoritmo, que é o número de vizinhos, não é possível encontrar de maneira clara, restando colocar valores até encontrar um ótimo para o caso. Nesse sentido, após vários esforços, encontrou-se um valor de número de vizinhos igual à 20.

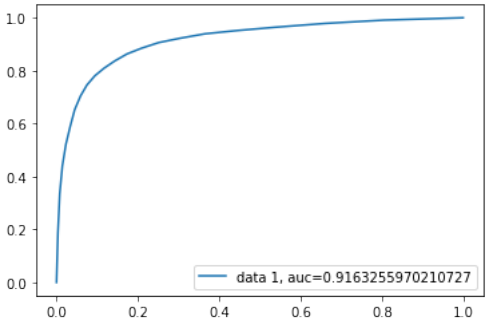

Figura 15: ROC para o kNN

E. Estudo sobre SVM

Tabela VIII: Resultados obtidos no SVM *(Support Vector Machine)*

	Resultados
Accuracy	0.8396
Precision	0.8364
Recall	0.8204

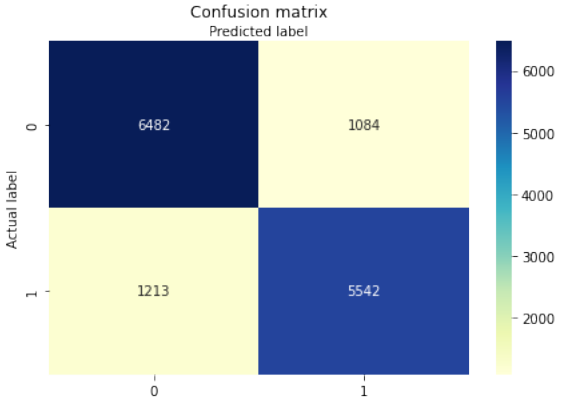

Figura 16: Matriz de Confusão para o SVM

O algoritmo SVM é um modelo de classificação fácil de ser implementado e se mostrou ser um bom classificador assim como o kNN e a Regressão Logística. Seus valores podem ser analisados pela tabela VIII e pela matriz de confusão 16, onde se tem uma acurácia de aproximadamente 84% e uma fração dos valores positivos igual a $\frac{12.024}{14321}$.

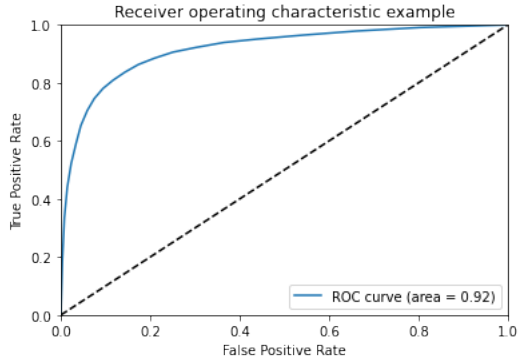

Figura 17: ROC para o SVM

IV. Conclusão dos Estudos e Comparações sobre os modelos classificadores

Tabela IX: Tabela comparadora entre os modelos de classificação

	Accuracy	**Precision**	**Recall**
LR	0.8350	0.8258	0.8241
QDA	0.7172	0.6378	0.9267
kNN	0.8482	0.8604	0.8096
SVM	0.8396	0.8364	0.8204

Referências

[1] Dgomonov, *New York City Airbnb Open Data*. (2019)
[2] Max Kuhn, Kjell Johnson, *Applied Predictive Modeling*. (2013)
[3] Vijay Swamy. *Lasso Versus Ridge Versus Elastic Net*. (2018)
[4] Alboukadel Kassambara. *Principal Component and Partial Least Squares Regression Essentials*. (2018)
[5] Herve Abdi and Lynne J. Williams. *Principal component analysis*. (2010)