NYC Airbnb Open Data: Exploração dos dados e Previsão do tipo de aluguel

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

Devido a disponibilidade de dados transparentes e em larga escala, a Airbnb é uma das empresas mais utilizadas para estudos de casos em ciencia de dados e inteligencia artificial fazendo que a empresa se torne uma das vanguardistas na utilização de processamento de dados no mercado como forma de otimizar a experiencia e a qualidade do produto. Com a utilização da linguagem Python e do *dataset* "New York City Airbnb Open Data" [1] foi possível classificar o tipo de aluguel a partir do uso de técnicas de modelo preditivo 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 analisar informações como: quais regiões de Nova Iorque estão mais suscetíveis a usar esse tipo de tecnologia (Airbnb) e quais são os principais tipos de aluguel por área.

No que se diz a respeito nos modelos preditivos utilizados para a criação das previsões do tipo do aluguel, destaca-se a Regressão Logística *(LR)*, Analise Discriminante Quadratica *QDA)* , n-vizinho mais próximo (kNN)e Support Vector Machine (SVM). Dentre essas destacadas, foram feitas operações de comparação para identificar qual predição se encaixa melhor no contexto, para isso utilizamos como métodos de avaliação a acurácia, precisão, e recall.

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 componente atual, μ_x a média dos componentes e s o desvio padrão

Para descobrir qual a melhor modelo, comparamos os seguintes modelos: regressão linear, Ánalise Discriminante Quadrática, KNN-k-nearest neighbours (k-vizinhos mais proximos) e SVM -support vector machines que serão posteriormente exploradas com mais afinco.

A 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 definição da reta e **n** é a quantidade de observações.

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

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

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

No caso da Analise Discriminante Quadrática nos utilizaremos a modelagem a distribuição de probabilidade a partir do Teorema de Bayes e selecionamos a classe que maximize a probabilidade a posteriori.

No caso do KNN, é um modelo não paramétrico que utiliza a distancia euclidiana para reunir os N vizinhos mais próximos de uma dada amostra K, e utiliza essa proximidade dos vizinhos para realizar agrupamentos aplicando a regra de Bayes de forma a permitir uma classificação, sendo um modelo simples.

Com o SVM, temos o uso de variáveis para predição de classes estritamente separadas utilizando um espaço n-dimensional de acordo com um hiperplano traçado, onde o vetor de suporte é a orientação do hiperplano que melhor separa as duas classes. Esse hiperplano é escolhido de acordo com o aprendizado supervisionado e maximização da margem (distancia entre o hiperplano e as classes), gerando um classificador.

B. Matriz Airbnb

No nosso dataset trabalharemos com uma matriz de 48895 linhas e 11 colunas, que, após transformarmos os dados categóricos em numéricos, atribuindo um valor para cada *string*, ficamos com 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. Análise mono-variada

Realizando uma análise mono-variada com base na classe conseguimos notar como cada preditor se comporta variando a relação com o aluguel do tipo apartamento inteiro ou quarto privado.

Primeiro vamos calcular o desvio padrão, *skewness* e média de cada preditor com base na sua classe, para que assim possamos fazer uma análise mais detalhada do dataframe que estamos trabalhando, gerando a Tabela I.

Tabela II: Média, desvio e skewness classe quarto privado

variável	média	desvio padrão	skewness
latitude	40.72	0.05	0.07
longitude	-73.96	0.04	1.48
número_reviews	22.84	42.40	3.31
reviews_por_mês	1.31	1.39	2.13
grupo_bairro	1.69	0.67	0.29
bairro	111.95	67.85	0.16
Residência por host	44.38	32.952519	5.84
Disponibilidade	111.92	129.80	0.73
price_log	4.76	0.68	0.67
noites_mínimas	8.50	22.94	19.78

Já para os dados relacionados ao quarto privado temos os seguintes resultados:

Tabela III: Média desvio e skewness classe apartamento inteiro

variável	média	desvio padrão	skewness
latitude	40.72	0.05	0.37
longitude	-73.94	0.04	1.12
número_reviews	24.11	47.28	3.93
reviews_por_mês	1.43	1.61	4.48
grupo_bairro	1.65	0.79	0.45
bairro	102.03	69.61	0.36
Residência por host	3.22	10.21	16.15
Disponibilidade	111.20	132.09	0.81
price_log	4.74	0.68	0.51
noites_mínimas	5.37	16.29	21.95

Como vemos comparando as Tabelas III e III, a maioria dos dados permanece muito próximos, com exceção de noites mínimas, residência por host e reviews, notamos que para os alugueis do tipo quarto privado, são exigidas em média, menos noite mínimas, os hosts possuem mais anúncios desse tipo e são feitas mais reviews.Essas diferências são importantes para aumentar a precisão do nosso modelo de classificação, afinal, quantos mais parecidos os dados mais difícil será definir a que grupo a previsão se encaixará.

Outro fator importante de comentar são os outliers,dados muito diferente dos outros, como nossas variáveis estão sujeitas a ações humanas, os outliers podem ser dados de um público que busca algum nicho , pois pode ser algum aluguel de um imóvel excepcional, optamos por não tirarmos.

D. Análise bi-variada

Como nós estamos tratando com muitas variáveis, seria muito custoso abordamos todas as relações entre as variáveis por histogramas, portanto, optamos por mostrar ao leitor a correlação entre cada preditor de cada classe.

Comparando as correlações entre as figuras 1 e 2 vemos que não existem grandes mudanças entre as relações de preditores ao mudar as classes.

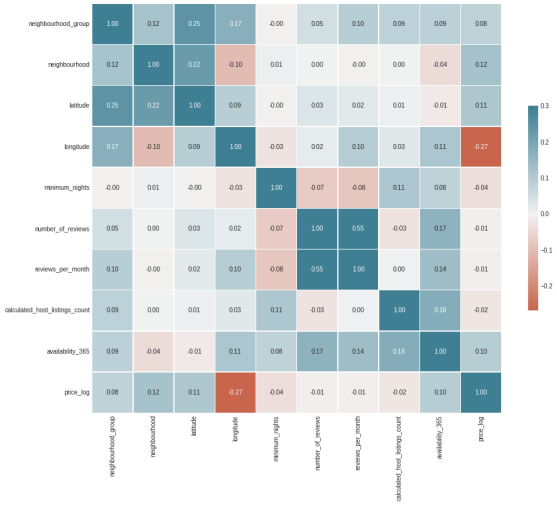

Figura 1: Matriz de correlação classe Apartamento Inteiro

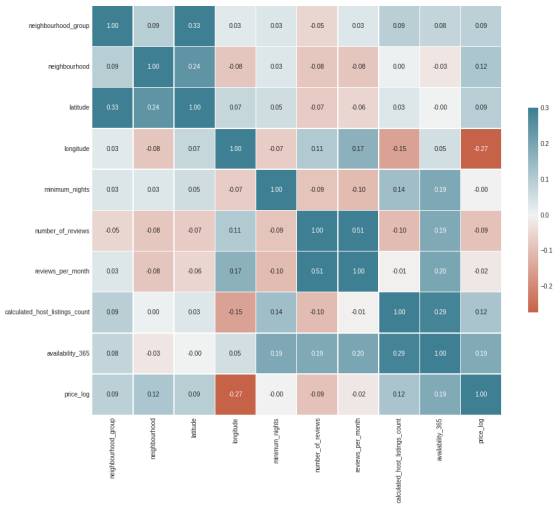

Figura 2: Matriz de correlação classe Quarto Privado

Como vemos também, nossos dados não são muito relacionados, existindo apenas uma relação forte, reviews por mês e reviews, sendo todas as outras relações fracas, por serem inferiores em módulo que 0.3 .

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
2.01376643	0.18306464
1.76327523	0.16029334

Como vemos, apenas dois componentes não é o suficiente para representar bem os dados,para ilustrar melhor a variância associada a cada componente vamos usar o scree plot em 3.

Figura 3: Variância por número de componentes

Exemplificado pelo gráfico 3, para alcançar uma variância de aproximadamente 60% seria necessário ao menos 5 componentes. Outra forma de vermos que nossos dados não são bem separados é com o uso do scatter plot 4.

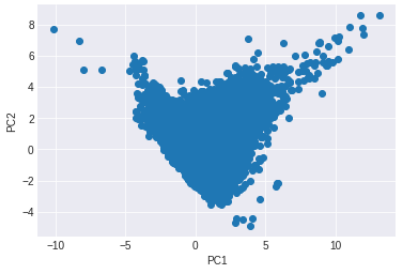

Figura 4: ScatterPlot entre PC1 e PC2

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 [1], foi necessário fazer uma limpeza dos dados pois o 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 dataset usado [1].

Dessa forma, além dessa manipulação de limpeza utilizamos *LabelEncoder*, da biblioteca *sklearn*, para transformar três colunas de dados categóricos em numéricos e, realizar a 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()*.

B. Estudo utilizando a Regressão Logística (LR)

Tabela V: Resultados obtidos na Regressão Logística

	Resultados
Accuracy	0.8350
Precision	0.8258
Recall	0.8241

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% [1]

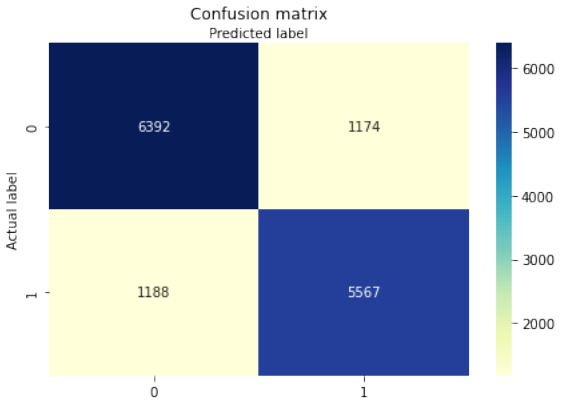

Figura 5: Matriz de Confusão para a Regressão Logística

Pela matriz de confusão da Figura 5, percebe-se que se obteve valores verdadeiros nas quantidades 6382 e 5567, o que representa que o algoritmo acertou 6382 dos 7580 totais existentes para "Entire Room/apt" e 5567 de 6.741 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.

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

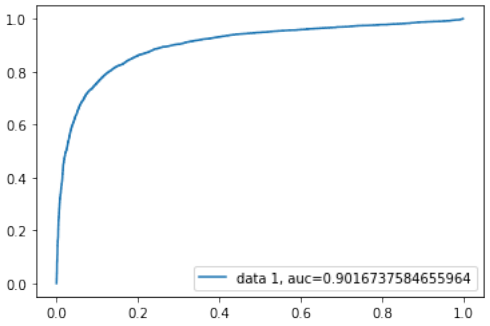

Figura 6: ROC para a Regressão Logística

RPF = Positivos Falsos / Negativos Totais, valores esses encontrados em uma matriz de confusão da Figura, por exemplo 5.

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 [6].

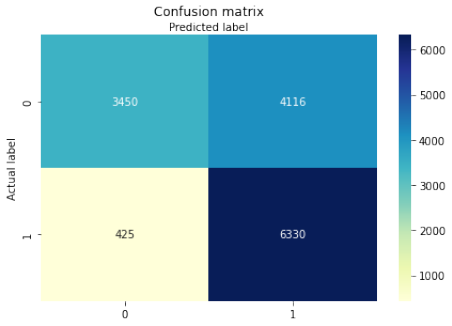

Figura 7: Matriz de Confusão para a QDA sem melhorias

Como nas figuras 7 e 8 e na Tabela VI, é possível destacar as diferenças de valores entre as duas versões do QDA, isso se dá

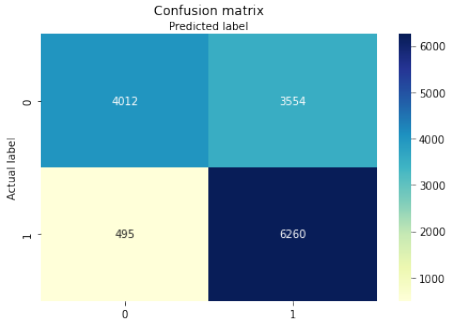

Figura 8: Matriz de Confusão para a QDA com melhorias

pelo fato do QDA melhorado utilizar parâmetros melhores do que a versão normal e percebemos a 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 realizamos diversos "loops" no algoritmo para otimização dos parâmetros para testados [6]. Obtendo assim um valor melhor em comparação ao modelo padrão.

Nesse raciocínio, pelas matrizes de confusão de ambas versões 7 e 8, 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.

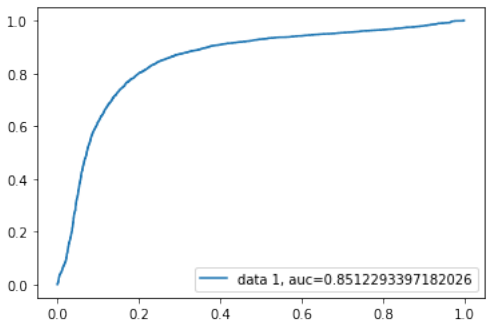

Figura 9: ROC para a QDA sem melhorias

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

Tabela VII: Resultados obtidos no kNN

	Resultados
Accuracy	0.8482
Precision	0.8604
Recall	0.8096

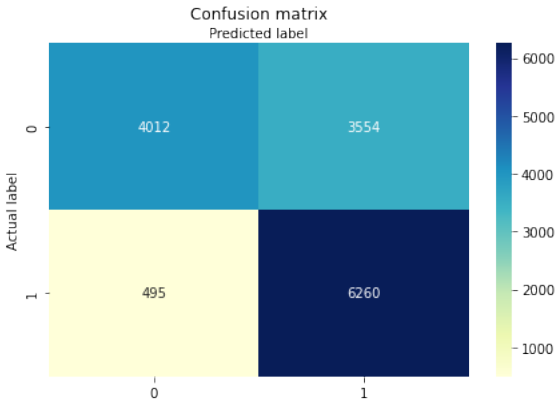

Figura 10: ROC para a QDA com melhorias

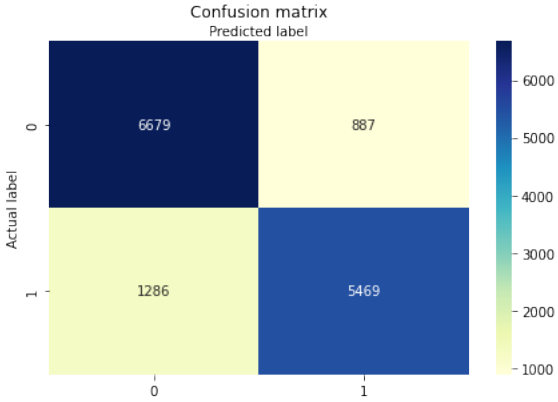

Figura 11: Matriz de Confusão para o kNN

Comparado aos outros dois modelos expostos anteriormente, o algoritmo kNN aparenta ser promissor, ja que pelos resultados da Tabela VII e pela matriz de confusão da Figura 11, é possível analisar que se tem valor de acurácia de 84.2% [5], 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.

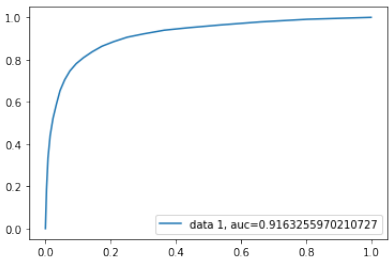

Figura 12: ROC para o kNN

E. Estudo sobre SVM

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

	Resultados
Accuracy	0.8396
Precision	0.8364
Recall	0.8204

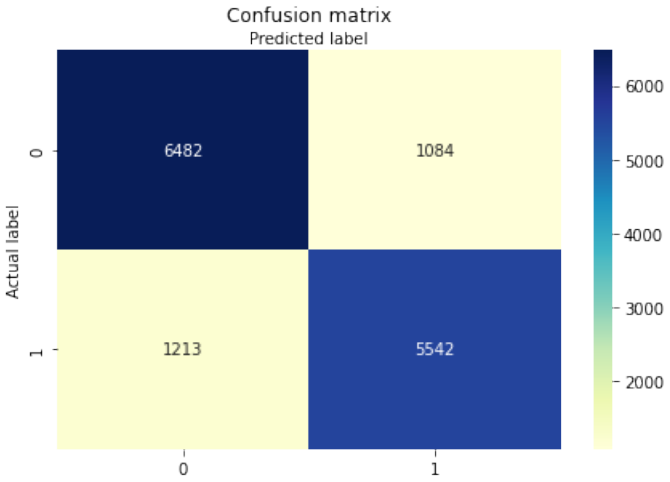

Figura 13: Matriz de Confusão para o SVM

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 da Figura 13, onde se tem uma acurácia de aproximadamente 84% e uma fração dos valores positivos igual a $\frac{12.024}{14321}$.

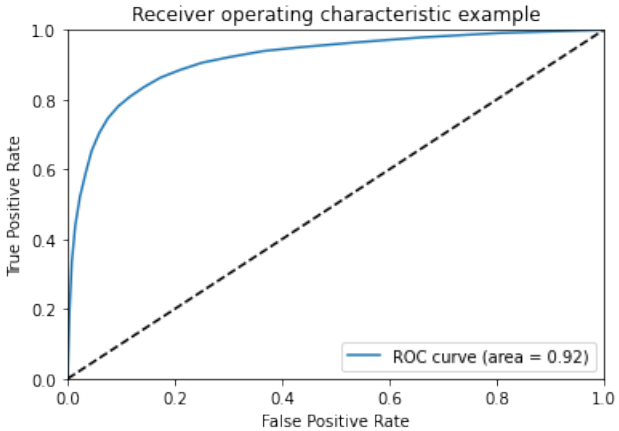

Figura 14: ROC para o SVM

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

Analisaremos a seguir o desempenho dos modelos de classificação a partir da seguinte Tabela IX, a qual tivemos maiores acurácia e precisão tanto no agrupamento por kNN como no SVM, todavia a regressão logística obteve resultado também satisfatório, isso demonstra um comportamento mais linear nos dados analisados. Isso indica que os modelos apresentam bom resultado quanto a classificações corretas e estabelecimento de positivos. Vemos, no entanto, que quem possui maior recall é a QDA, seguido da regressão linear e SVM, mostrando boas opções para um cenário onde a presença de falsos negativos sejam mais pungentes, porém, é um resultado pouco significativo para a QDA considerando o baixo desempenho nos outros criterios.

Devemos levar no entanto em consideração que o custo computacional de cada método bem como o tamanho do dataset também podem influenciar a escolha e o desempenho de cada método, sendo essa analise restrito ao desempenho absoluto dos métodos para um mesmo dataset e disponibilidade de recursos computacionais, além de tempo de desenvolvimento e parametrização desconsiderados.

Referências

[1] Dgomonov, *New York City Airbnb Open Data*. (2019)
[2] Max Kuhn, Kjell Johnson, *Applied Predictive Modeling*. (2013)
[3] Herve Abdi and Lynne J. Williams. *Principal component analysis*. (2010)
[4] Avinash Navlani. *Understanding Logistic Regression in Python*. (2019)
[5] Scott Robinson. *K-Nearest Neighbors Algorithm in Python and Scikit-Learn* (2021)
[6] Gellert Toth. *Linear and Quadratic Discriminant Analysis* (2020)