# Aplicação de Reconhecimento de Padrões em Banco de Dados de Satisfação de Passageiro em Companhias Áreas

Ricardo Piero Lippoli Batista, João Vitor Santos Barbosa, Matheus Marques Duarte

## I. INTRODUÇÃO

Objetivando assimilar o conhecimento adquirido em Ciência de Dados na disciplina de Reconhecimento de Padrões, foi proposto este trabalho, que consiste em utilizar os métodos de classificação estudados durante o período para classificar um banco de dados predeterminado.

### A. Base de Dados

Para essa análise, foi utilizado o banco de dados "Airline Passenger Satisfaction", que essencialmente consiste em uma pesquisa de satisfação com mais de 25000 passageiros de companhias aéreas. Com o objetivo de compreender os passageiros, entender as opiniões e correlacionar os dados, foram empregados os seguintes parâmetros:

- Gênero do Passageiro
  - HOMEM
  - MULHER
- Tipo de Cliente
  - LEAL
  - DESLEAL
- Idade (anos)
  - min: 7
  - max: 85
- Tipo de Viagem
  - PESSOAL
  - NEGOCIOS
- Classe
  - ECO
  - ECO PLUS
  - EXECUTIVA
- Distância de Voo (km)
  - min: 31
  - max: 4983
- Serviço Wi-Fi a bordo
- Hora da Partida/Hora de chegada conveniente
- Facilidade de reserva online
- Localização do portão
- Comida e Bebida
- Embarque Online
- Conforto do assento
- Entretenimento a bordo
- Serviço de bordo
- Serviço de quarto para pernas
- Manuseio da bagagem
- Serviço de check-in
- Serviço de bordo
- Limpeza
- Atraso de partida (min)
  - min: 0
  - max: 1128
- Atraso de chegada (min)
  - min: 0
  - max: 1115
- Satisfação
  - SATISFEITO
  - NEUTRO OU NÃO SATISFEITO

Observe que os seis primeiros dados referem-se apenas a informações informativas sobre o passageiro, e os dados de atraso também são adquiridos, totalizando 13 dados de satisfação sobre a viagem. Os níveis de satisfação variam de 0 a 5 para todos os atributos. Entre esses, destaca-se que 18 são numéricos e 5 são textuais. A fim de verificar a eficiencia dos modelos, foram calculados as metricas: Acuracia, F1 Score, Recall, Kappa e Precisão.

## II. CLASSIFICAÇÕES

Buscando trabalhar com a base de dados mencionada anteriormente, foi necessário realizar o pré-processamento dos dados para lidar com valores ausentes. Em seguida, foi gerado um histograma para cada classe, a fim de analisar e contar a quantidade de dados em cada uma. Além disso, empregou-se um código para extrair informações sobre a assimetria de cada classe. Posteriormente, foi elaborado um boxplot das variáveis com atributos numéricos. Após a conclusão dessas etapas iniciais, foi possível realizar uma busca por variáveis que apresentassem outliers no banco de dados.

### A. Classificação Linear

Uma vez que o atributo "Customer Type"foi definido como classe e "Satisfaction"como o alvo, a fim de identificar atributos redundantes para a classificação linear, utilizou-se o método corr() da biblioteca Pandas.

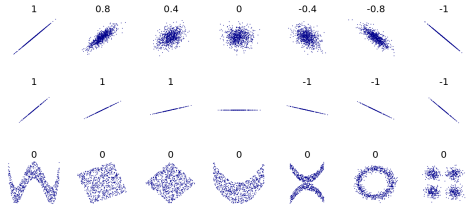


Fig. 1. Vários conjuntos de pontos (x, y), com o coeficiente de correlação de x e y para cada conjunto.

Foram utilizados três métodos diferentes para garantir a precisão da função, e após analisar o dataframe de cada método (exemplo na figura 9), foram descartados os atributos a seguir:

- Arrival Delay in Minutes
- Departure Delay in Minutes
- Cleanliness
- Inflight service
- Inflight entertainment
- Seat comfort
- Online boarding
- Ease of Online booking
- Departure/Arrival time convenient
- Inflight wifi service

Em seguida, foi realizada a Classificação Linear utilizando os atributos restantes e selecionando 500 amostras de cada classe. Dessa maneira, a classificação foi balanceada para facilitar a visualização.

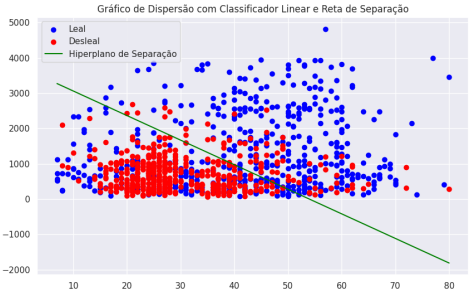


Fig. 2. Gráfico de Dispersão com Classificador Linear e Reta de Separação.

Após a classificação foi implementado um gráfico de dispersão e traçado o hiperplano de separação.

B. Naive-Bayes

A próxima Classificação realizada foi a de Naive-Bayes. Fora calculada as probabilidades de cada nível de satisfação manualmente (figura 3) para o atributo Inflight wifi service. Após isso, os atributos textuais foram convertidos em valores de -1 a 1.

```
#Frequencia:
#    Não     Sim
#0 = 2     = 811  = 813/25976  = 0.0313%
#1 = 2966  = 1522 = 4488/25976 = 0.1727%
#2 = 4923  = 1567 = 6490/25976 = 0.2498%
#3 = 4694  = 1623 = 6317/25976 = 0.2431%
#4 = 1953  = 3028 = 4981/25976 = 0.1917%
#5 = 35    = 2852 = 2887/25976 = 0.1111%
#Total = 14573/25976 = 0.561% | 11403/25976 = 0.439%
#TotalTudo = 25976

#Prob ser insatisfeito e dar 0:
#P0N = 2/14573 PN = 14573/25976 P0 = 813/25976
PN0 = (2/14573) * (0.561)/(0.0313)
PS0 = 1 - PN0
#Prob ser insatisfeito e dar 1:
#P0N = 2/14573 PN = 14573/25976 813/25976
PN1 = (2966/14573) * (0.561)/(0.1727)
PS1 = 1 - PN1
#Prob ser insatisfeito e dar 2:
#P0N = 2/14573 PN = 14573/25976 813/25976
PN2 = (4923/14573) * (0.561)/(0.2498)
PS2 = 1 - PN2
#Prob ser insatisfeito e dar 3:
#P0N = 2/14573 PN = 14573/25976 813/25976
PN3 = (4694/14573) * (0.561)/(0.2431)
PS3 = 1 - PN3
#Prob ser insatisfeito e dar 4:
#P0N = 2/14573 PN = 14573/25976 813/25976
PN4 = (1953/14573) * (0.561)/(0.1917)
PS4 = 1 - PN4
#Prob ser insatisfeito e dar 5:
#P0N = 2/14573 PN = 14573/25976 813/25976
PN5 = (35/14573) * (0.561)/(0.1111)
PS5 = 1/100 - PN5
```

Fig. 3. Calculo de Naive-Bayes de modo manual

Após determinar a matrize X com os valores do dataframe sem o atributo alvo e a matriz Y com os valores do atributo alvo, foi chamada a função de treino, com as matrizes produto da função anterior, foi feito os ajustes para o definir a classificação do modelo gausiano e a classificação.
Uma vez feita a classificação, foram calculados as métricas, para validar a eficácia da classificação, junto a isso também foi feita uma matriz de confusão (figura 4) para observar o desempenho.

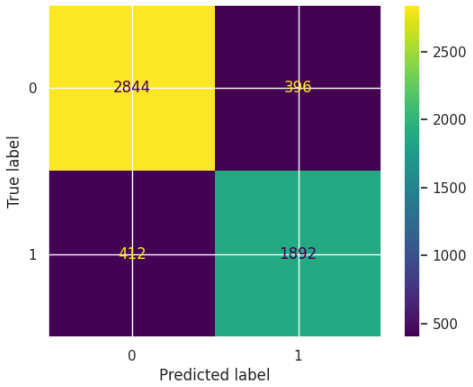


Fig. 4. Matriz de confusão referente a Naive-Bayes

C. Árvore de Decisão (simples)

Após isso, utilizando a função tree. DecisionTreeClassifier() da biblioteca sklearn, foi realizada a Classificação de Árvore de Decisão e plotado o resultado (figura 5 e em anexo). Logo em seguida também foi calculado as métricas e feita a matriz de confusão para o método de classificação (figura 6).

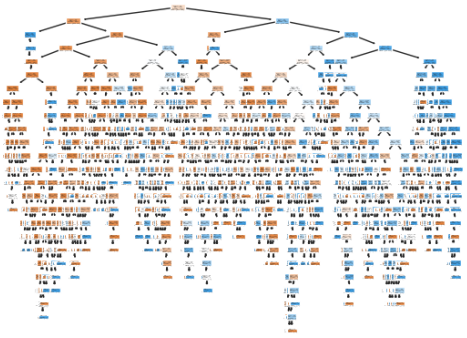

Fig. 5. Plote da classificação por árvore de decisão

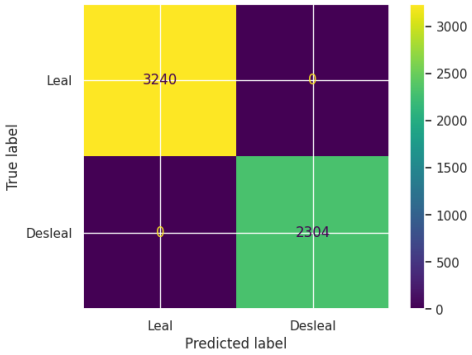


Fig. 6. Matriz de confusão referente a árvore de decisão

D. Rede Neural MLP

Utilizando a função MLPClassifier() da biblioteca sklearn, foi realizada a Classificação de rede neural MLP e após treinar o modelo, foi obtido o valor do score. Logo em seguida, foi feita a matriz de confusão para o método de classificação (figura 7).

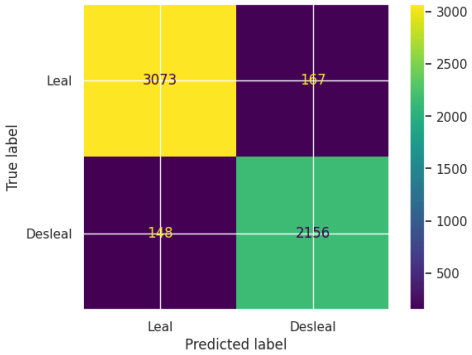


Fig. 7. Matriz de confusão referente a Rede Neural MLP

E. Random Florest

Semelhante a classificação anterior, usando a função RandomForestClassifier() da biblioteca sklearn, foi realizada a Classificação de random florest e após treinar o modelo. Logo em seguida, foi feita a matriz de confusão para o metodo de classificação (figura 8).

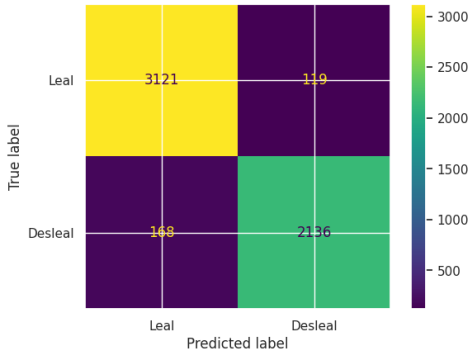


Fig. 8. Matriz de confusão referente a random florest

## III. Resultados e Discussão

A. Pré-processamento

- Quantos registros a base de dados possui? 25976 Registros.
- Quantos atributos? 25 atributos.
- O que quer dizer cada atributo? 6 são dados sobre o passageiro enquanto outros 13 são notas dadas por eles para os serviços das companhias aereas.
- A base de dados possui valores ausentes? Sim, 83 dados faltantes.
- A base de dados possui outliers? Sim, 4 atributos com outliers.
- Análise monovariada e bivariada dos atributos: Exemplo da analise bivariada feita ao longo do trabalho:

Fig. 9. tabela de correlação

- Existem atributos com elevada assimetria? Sim, 2 atributos com assimetria maior que 6.
- Qual a melhor forma de normalizar os dados? No trabalho como os valores eram muito proximos, não foram usados formas de normalização. Contudo ao longo dos testes a aplicação do log1p e sqrt para os valores se mostrou eficiente.
- Existe desbalanceamento entre as classes? Sim, porem é pequeno.
- É viável fazer seleção de atributos? É viavel uma vez que são menos atributos para tratar e alguns possuem alta correlação.

B. Processamento Treinamento dos modelos

- Ajuste dos hiperparâmetros: Qual método usar?
- Qual métrica utilizada para o ajuste? Foi utilizada a metrica accuracy, F1 score, kappa, precission e recall
- Naive-Bayes: Accuracy: 0.83 | F1 Score: 0.83
- Árvore de Decisão (simples): Accuracy: 1.0 | F1 Score: 1.0
- Rede Neural MLP: Accuracy: 0.59 | F1 Score: 0.63
- Random Florest Accuracy: 0.89 | F1 Score: 0.90

C. Pós-processamento: Interpretação dos resultados

- Qual modelo desempenhou melhor? Levando em consideração tempo de processamento (medido com cronometro) e valor de metricas, o melhor modelo foi o random florest.
- tabela comparativa com os resultados da validação cruzada

| | Naive-Bayes | Rede Neural MLP | Random Florest | Árvore de Decisão (simples) |
|---|---|---|---|---|
| Accuracy | 0,854256854257 | 0,943181818182 | 0,943181818182 | 1,000000000000 |
| F1 Score | 0,854331275544 | 0,943148704229 | 0,948865483540 | 1,000000000000 |
| Kappa | 0,699660152931 | 0,883168679283 | 0,894182659273 | 1,000000000000 |
| Precission | 0,821180555556 | 0,935763888889 | 0,926649305556 | 1,000000000000 |
| Recall | 0,826923076923 | 0,928110202325 | 0,948888888889 | 1,000000000000 |
| Processamento | 4,500000000000 | 20,000000000000 | 4,000000000000 | 60,000000000000 |
| Score/Tempo | 0,96 | -13,60 | 2,45 | -53,00 |
| Score | 5,46 | 6,40 | 6,45 | 7,00 |
| Final | -4,50 | -20,00 | -4,00 | -60,00 |

Fig. 10. tabela para comparação das metricas

- Curva ROC Final

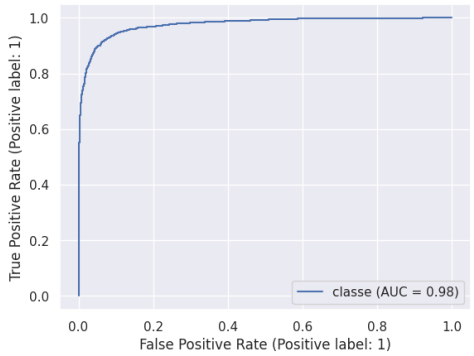


Fig. 11. Curva ROC Final

- Matriz de confusão do melhor modelo: figura 8

## IV. CONCLUSÕES

Em conclusão, o estudo sobre reconhecimento de padrões revelou pontos significativos sobre a eficácia de diversos classificadores. O classificador de árvore de decisões simples demonstrou uma performance excepcional, liderando em termos de acurácia e F1 Score, refletindo sua capacidade de aprender padrões complexos. Por outro lado, o Random Forest apresentou um desempenho sólido, figurando como uma alternativa eficiente com um tempo de execução aceitável. Destaca-se que, embora a Árvore de Decisão tenha revelado competência em classificação, sua demora na execução sugere a necessidade de considerações adicionais em relação à eficiência temporal, fator crucial em muitos contextos práticos. Além disso, a exploração de arquiteturas de redes neurais alternativas e ajustes nos parâmetros pode proporcionar melhorias adicionais em todos os modelos propostos.

Em suma, este trabalho não apenas aprofundou a compreensão dos métodos de reconhecimento de padrões, mas também forneceu orientações valiosas para a escolha prática de classificadores, considerando a dualidade entre desempenho e eficiência computacional.