UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE COMPUTAÇÃO

Trabalho 1 de Aprendizado de Máquina

Trabalho de predição de chuvas na Austrália

Áquila Oliveira, 759313

Letícia Bossatto Marchezi, 791003

Mateus Grota Nishimura Ferro, 771043

São Carlos, SP, 2022

1 Introdução

Este primeiro trabalho tem o objetivo dividido em três etapas, descritas por escolha do conjunto de dados, estudo do conjunto de dados e classificação usando modelos de aprendizado de máquina supervisionados.

O conjunto de dados escolhido foi o *dataset Rain in Australia*, que possui dados de quando choveu e não choveu nas cidades da Austrália, ou seja, nosso intuito com esse *dataset* é classificar se irá chover e mais especificamente chover amanhã (Rain Tomorrow).

O estudo do conjunto de dados será composto pela análise de atributos, assim como suas relações, distribuições e visualização dos dados.

O pré-processamento contém etapas essenciais para o tratamento dos dados e aprimoramento dos resultados do modelo. É necessário estabelecer uma estratégia para tratar valores nulos, normalizar os dados quantitativos, transformar as colunas categóricas e mitigar o desbalanceamento de classes.

Posteriormente ocorre a classificação, realizada com os algoritmos Naive Bayes, Decision Tree, Random Forest, Logistic Regression, SVM e AutoML. O objetivo é buscar o melhor desempenho ao acertar a classificação dos dados, por isso é realizada a análise dos resultados dos modelos.

2 Justificativa e Objetivos

O Dataset *Rain in Australia* foi escolhido devido a uma possível verossimilhança com a realidade, pois relaciona-se diretamente com a área da Metereologia e aspectos cotidianos: presença de chuva, variações de temperatura, pressão e nuvens no céu.

As conclusões desse estudo de caso podem confirmar conhecimentos prévios, ou seja, os dados mais relevantes podem coincidir com os critérios do mundo real, ou podem mostrar novas variáveis importantes para este espaço amostral específico. Além disso, o dataset possibilita a realização de diferentes análises e estratégias para alcançar o objetivo, estando aberto ao estudo de seus variados dados e distribuições.

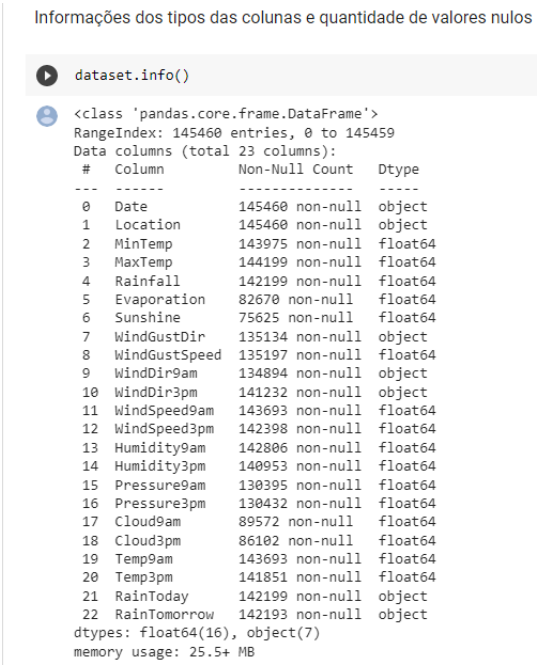
Este trabalho tem como principal objetivo a aplicação de diferentes métodos de classificação no *dataset Rain in Australia*, com dados metereológicos de diferentes localidades da Austrália. Para isso, são definidas as seguintes etapas:

- Extrair informações sobre o conjunto de dados e tais aplicações ao mundo real;
- Comparar resultados dos diferentes classificadores;
- Avaliar eficiência e desempenho de cada modelo proposto;
- Testar a otimização de hiperparâmetros e AutoML em busca de melhores resultados;

3 Pré-processamento

O pré-processento é a etapa onde validamos os dados e os transformamos para que possa ser utilizado nos classificadores de um modo coerente, ou seja, realizaremos processos de limpeza e regularização. Listaremos alguns dos processos que podem e vão ser usados para o nosso trabalho, como: eliminação de atributos sem alta relevância, tratamento de dados ausentes, encoding de dados nominais para atributos binários e normalização de atributos numéricos.

Visualização das informações dos atributos do *dataset*.


Informações dos tipos das colunas e quantidade de valores nulos

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Date           145460 non-null  object
 1   Location       145460 non-null  object
 2   MinTemp        143975 non-null  float64
 3   MaxTemp        144199 non-null  float64
 4   Rainfall       142199 non-null  float64
 5   Evaporation    82670 non-null   float64
 6   Sunshine       75625 non-null   float64
 7   WindGustDir    135134 non-null  object
 8   WindGustSpeed  135197 non-null  float64
 9   WindDir9am     134894 non-null  object
 10  WindDir3pm     141232 non-null  object
 11  WindSpeed9am   143693 non-null  float64
 12  WindSpeed3pm   142398 non-null  float64
 13  Humidity9am    142806 non-null  float64
 14  Humidity3pm    140953 non-null  float64
 15  Pressure9am    130395 non-null  float64
 16  Pressure3pm    130432 non-null  float64
 17  Cloud9am       89572 non-null   float64
 18  Cloud3pm       86102 non-null   float64
 19  Temp9am        143693 non-null  float64
 20  Temp3pm        141851 non-null  float64
 21  RainToday      142199 non-null  object
 22  RainTomorrow   142193 non-null  object
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

Figura 1: Informações sobre os atributos do *dataset*

Como descrito acima, o atributo *sunshine* possui 75625 valores não-nulos entre 145460, ou seja, quase metade dos dados desta coluna estão ausentes. Nestes casos, é importante estabelecer estratégias de tratamento para não ocorrer perda de informações.

Os dados nominais faltantes foram removidos do *dataset*, já que não representavam muitas amostras. Entre eles, destaca-se a coluna *"RainTomorrow"*, o atributo alvo da classificação, cujas amostras em que seu valor estava ausente foram excluídas. Já os atributos quantitativos faltantes foram substituidos por 0 e não removidos, como mostrado na Figura 2.

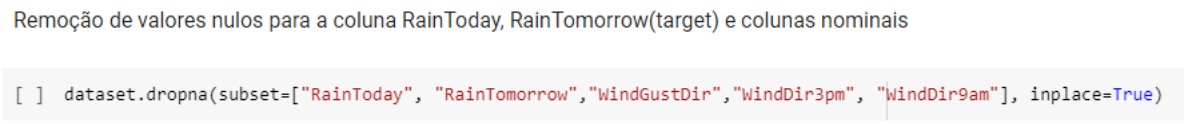


Figura 2: Removendo nulos de atributos nominais

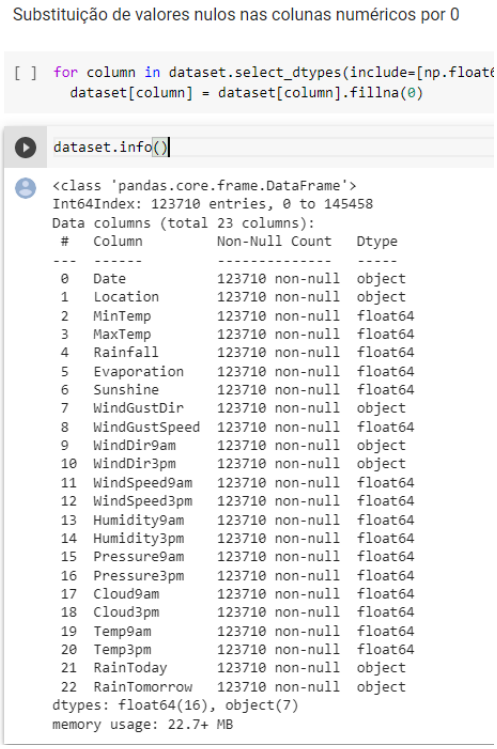


Figura 3: Informação dos atributos depois dos processamentos, sem nulos.

Com isso, diminuiu-se as linhas do *dataset* em quase 20000, mas permaneceram 120000 dados.

Após isso, mudamos o atributo alvo que é o *RainTomorrow* de *No* e *Yes* para 1 e 0. Mostrando a mudança na Figura 4.

Também foi realizada a transformação das colunas "RainTomorrow"e "RainToday"para valores binários, 1 ou 0.

A coluna de dia, mês e ano foi transformada em 2 novas colunas para mês e ano, pois é conveniente analisar a sazonalidade dos dados nesse caso.

O atributo alvo deste problema é 'RainTomorrow', ou seja, se ocorre chuva no dia seguinte. Os valores possíveis são 'No' e 'Yes'. Por questão de padronização, serão transformados em um atributo numérico binário(porém ainda nominal).

```
[ ] dataset["RainTomorrow"].unique()

    array(['No', 'Yes'], dtype=object)

[ ] dataset.loc[dataset.RainTomorrow=='Yes','RainTomorrow'] = 1.0
    dataset.loc[dataset.RainTomorrow=='No','RainTomorrow'] = 0.0

    dataset.loc[dataset.RainToday=='Yes','RainToday'] = 1.0
    dataset.loc[dataset.RainToday=='No','RainToday'] = 0.0

    dataset["RainTomorrow"] = dataset["RainTomorrow"].astype(float)
    dataset["RainToday"] = dataset["RainToday"].astype(float)
```

Figura 4: Passando *No* e *Yes* para 1 e 0.

Após a visualização dos dados, foi realizado o balanceamento das classes, já que a classe minoritária(Positivo) representava cerca de 1/4 dos registros. A técnica usada foi o SMOTE, conhecida como Técnica de sobreamostragem minoritária sintética. Assim, resultando numa proporção de 1/3, sem perda no desempenho dos classificadores.

Além disso, foi usado o método K-Best para selecionar as 60 melhores features dos dados.

4 Visualização dos Dados

Na visualizações de dados foram plotados gráficos que descrevem relações relevantes, desse modo é possível analisar os resultados.

Na Figura 5 mostra-se a quantidade de dados da classe 1 e classe 0 antes da técnica de Oversampling.

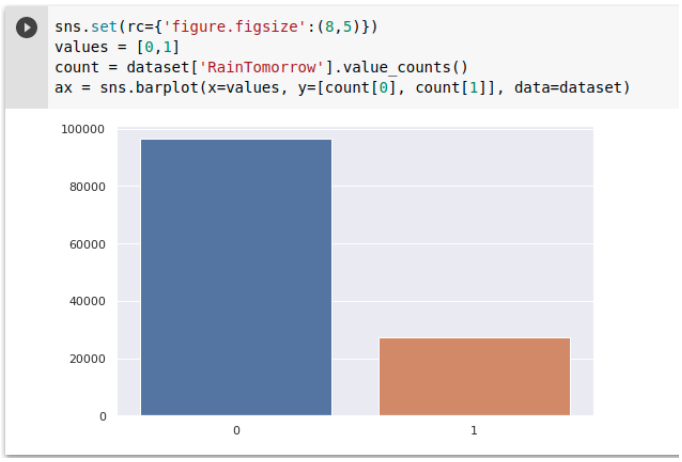


Figura 5: Quantidade de classes 0 e 1.

Na Figura 6 é possível ver a correlação de todos os atributos. É possível observar que o *target(Rain Tomorrow)* possui mais correlação com a umidade.

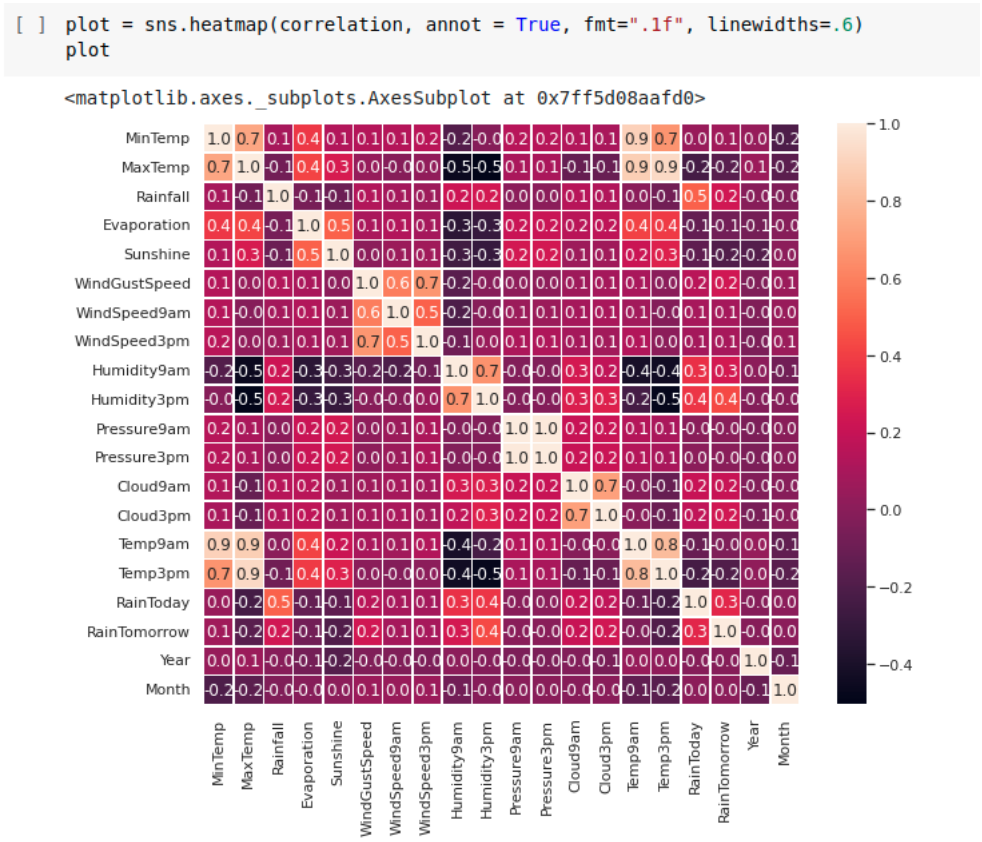


Figura 6: Correlação de atributos.

Na Figura 7 temos um gráfico de barras dos registros que temos depois do pré-processamento por ano.

A figura 8 mostra todos os registros que possui depois do pré-processamento das classes 1 e 0 por ano. A parte azul representa o atributo não choveu(0) e a parte laranja representa que choveu(1). É perceptível a estabilidade dos meses tanto de quando choveu quando não, uma das explicações é que a Austrália é um país tropical, onde as estações climáticas não influenciam tanto, diferentemente da Europa, Estados Unidos/Canadá e regiões acima do equador.

As figuras 9 e 8 mostram a quantidade de registros por ano e por mês, respectivamente, e que se mantém estáveis.

A figura 10 mostra é um gráfico de violino, do atributo de umidade as 9h da manhã em comparação com o RainTomorrow, é possível verificar que quanto maior a umidade, maior a densidade da classe 1(chuva) e quanto menor a umidade, mais classes 0(não chuva).

Na Figura 11 é mostrado o gráfico de violino, do atributo de umidade as 3h da tarde e é possível verificar a mesma situação.

Após a visuzalização dos gráficos, é possível observar que a umidade tem uma correlação com o atributo

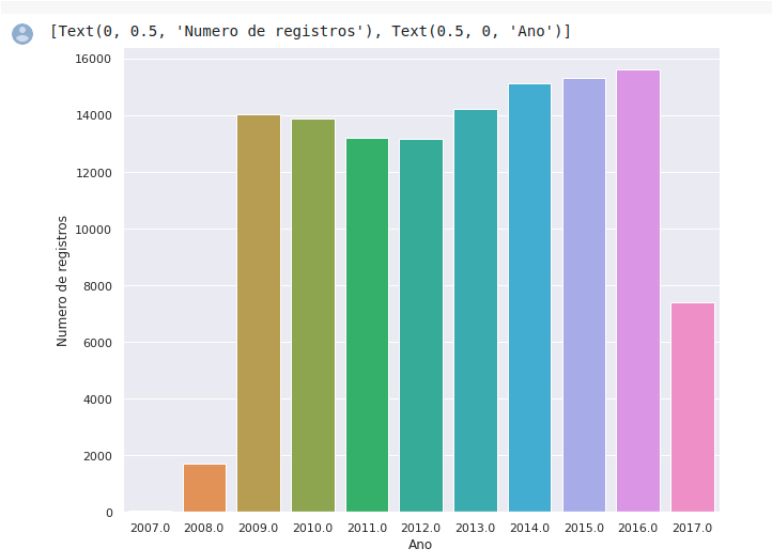


Figura 7: Registros que o dataset possuí por ano.

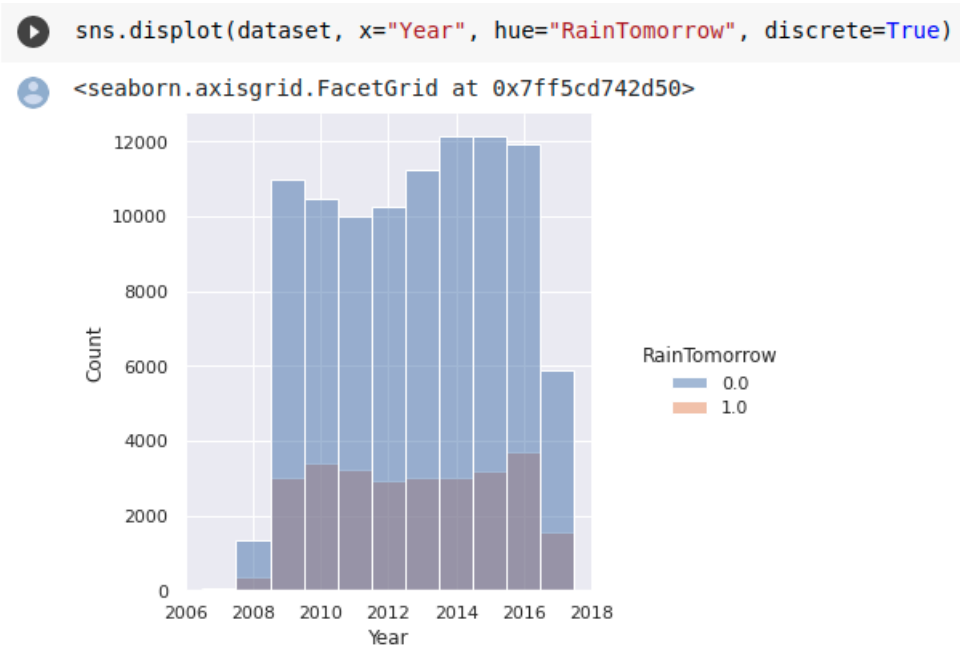


Figura 8: Registros que o dataset possuí da classe 1(chuva) e da classe 0(não chuva) por ano.

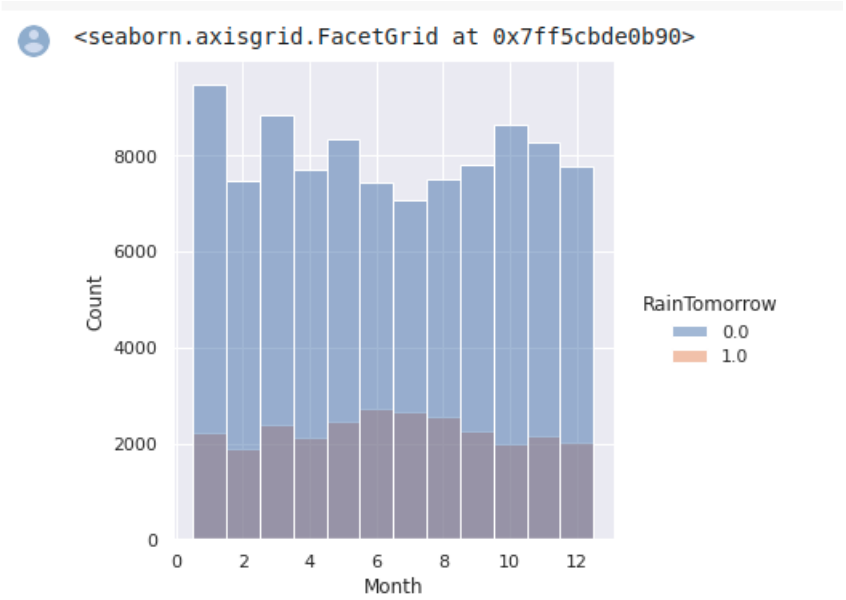


Figura 9: Registros que o dataset possuí da classe 1(chuva) e da classe 0(não chuva) por mês.

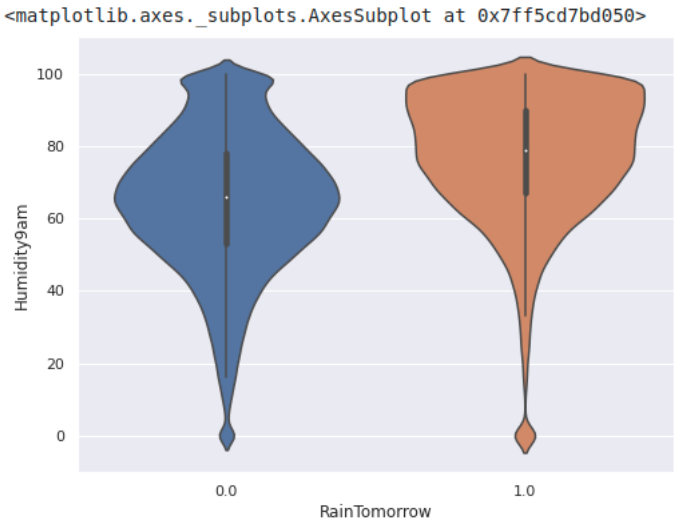


Figura 10: Quantidade de classes 0 e 1.

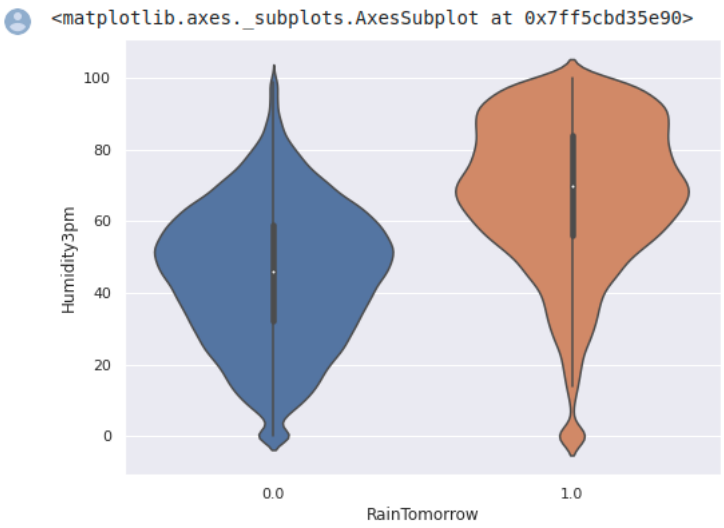


Figura 11: Gráfico de violino, da quantidade de umidade pelo target(RainTomorrow). Para umidade das 9h da manhã.

target(RainTomorrow), e foi decidido plotar os dados em função da umidade das 9h da manhâ e das 3h da tarde. Quanto mais para a parte superior e direita, mais temos dados da classe 1(chuva).

Após todos os gráficos, foi plotado alguns gráficos de atributos gerais para ter uma noção melhor dos dados em si.

5 Classificação

Para classificação foram utilizados sete modelos, que são: Naive Bayes, Árvore de Decisão, K-Nearest Neighbors(KNN), Support Vector Machines(SVM), Logistic Regression, Random Forest e AutoML.

Primeiramente foi executado os classificadores sem nenhum parâmetro, como o *dataset* possuí 123710 linhas, alguns classificadores como SVM e KNN demoraram mais para executar.

Após isso, foi utilizada a biblioteca *GridSearchCV* do *sklearn*, para a otimização de parâmetros. Infelizmente alguns modelos tiveram muito tempo de execução e mesmo diminuindo os *folds* e outros parâmetros que diminuem o tempo de execução, e assim teve a limitação de hardware. Alguns tempos de execução demorariam alguns dias para finalizar. Assim, foi utilizado o *GridSearchCV* apenas na Decision Tree que posteriormente irá ser um dos classificares de maior desempenho. Também foi aplicado o AutoML que é uma ferramenta para facilitar o e otimizar parâmetros do conjunto de dados.

Para Naive Bayes não foi utilizado nenhum parâmetro de otimização, deste modo foram utilizados os

Figura 12: Gráfico de violino da quantidade de umidade pelo target(RainTomorrow) para umidade das 3h da tarde.

parâmetros padrões da biblioteca.

Para o KNN foi utilizado o parâmetro de comparar com 5 vizinhos, e o resto dos parâmetros padrões que são *weights* igual a *uniform* para deixar todos os pontos com pesos equilibrados e *distance* para vizinhos mais próximos terem mais importância que os longes.

Para a Árvore de Decisão foi utilizado o parâmetro de número mínimo de amostras necessárias de um nó folha que foi estabelecido como 10. E random state como 0, que controla o processo de embaralhamento do classificador.

Para a Random Forest foi utilizado o parâmetro de número mínimo de amostras necessárias de um nó folha que foi estabelecido como 10. E random state como 1, que controla o processo de embaralhamento do classificador. Foi utilizado os mesmos parâmetros da Árvore de Decisão, já que são classificadores com a mesma ideia, que utiliza árvores.

Para o Logistic Regression utilizamos *sag* em *solver*, que é utilizado no problema de otimização, o valor *sag* é utilizado para *datasets* com maior quantidade de dados, igual ao nosso. E random state 0 que controla o embaralhamento do classificador.

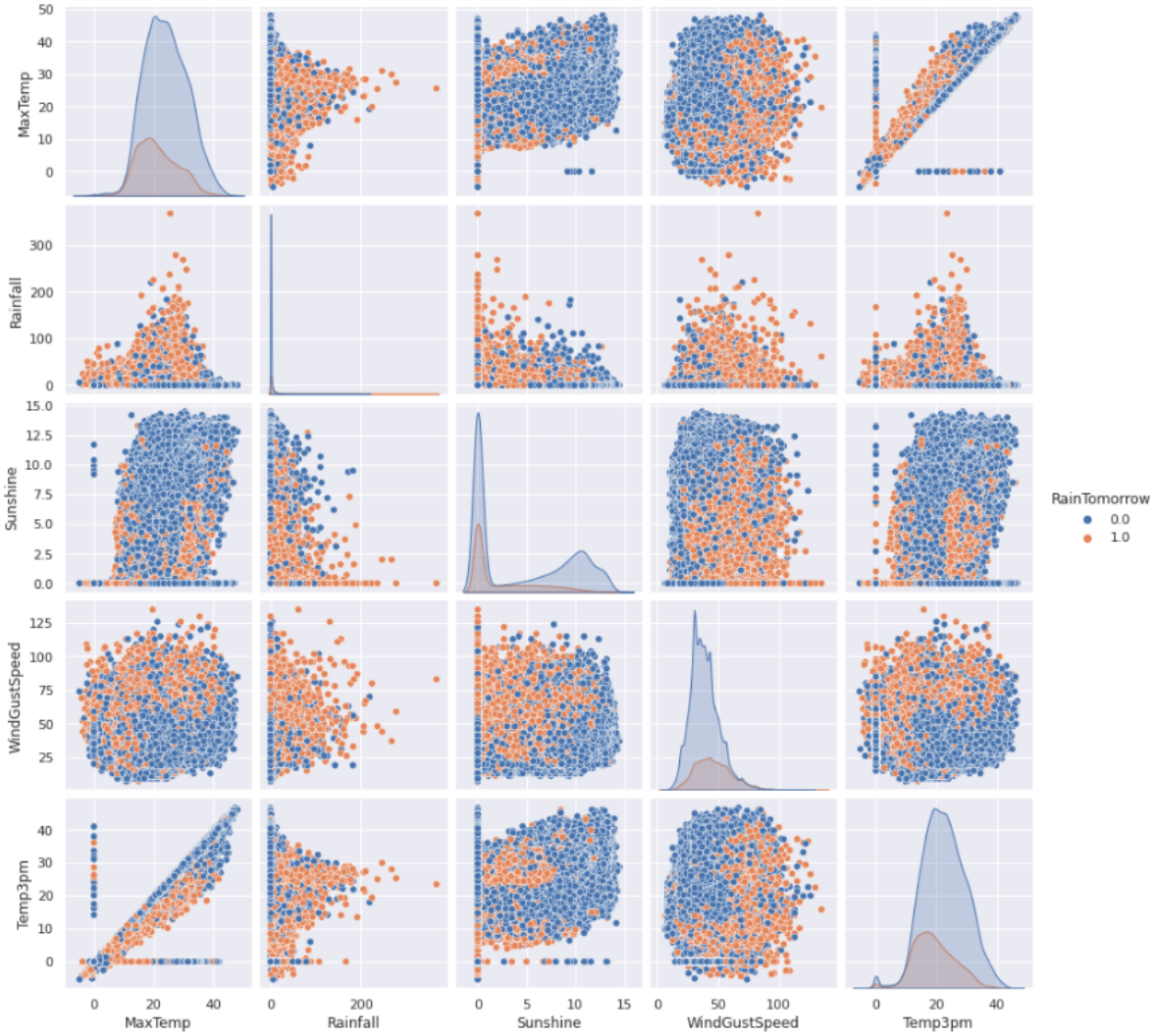


Figura 13: Plot de gráficos gerais.

Para o SVM foi utilizado apenas os parâmetros padrões que são gamma auto, C = 1 e kernel rbf. O C serve para controlar a margem de erro da reta traçado pelo SVM, ou seja, com C maiores temos uma margem de erro maior. O parâmetro Gamma é usado para decidir quanto de curvatura irá ser feita pelo SVM, serve para especificar e classificar melhor ou pior os dados quando estão maior perto entre si. Já o Kernel RBF usa o C e o Gamma para fazer contas que ajudam na classificação.

6 Análise do Desempenho e Resultado dos Classificadores

Para a análise do desempenho, foram usadas medidas relacionadas à tarefa de classificação, medidas como: acurácia, precisão, revocação, F1-score, suporte e curva ROC. A maioria das medidas são fornecidas pela função *classification_report()* da biblioteca *scikit-learn*. Além disso, também foi utilizada matrizes de confusão para uma melhor visualização das predições das classes realizadas pelos modelos.

6.1 *Naive Bayes*

O classificador de *Naive Bayes* obteve as seguintes medidas:

Modelo naiveBayes

	precision	recall	f1-score	support
0.0	0.77	0.86	0.81	31833
1.0	0.63	0.47	0.54	15845
accuracy			0.73	47678
macro avg	0.70	0.67	0.68	47678
weighted avg	0.72	0.73	0.72	47678

Figura 14: Medidas de desempenho do classificador de *Naive Bayes*

A sua matriz de confusão e curva ROC, respectivamente:

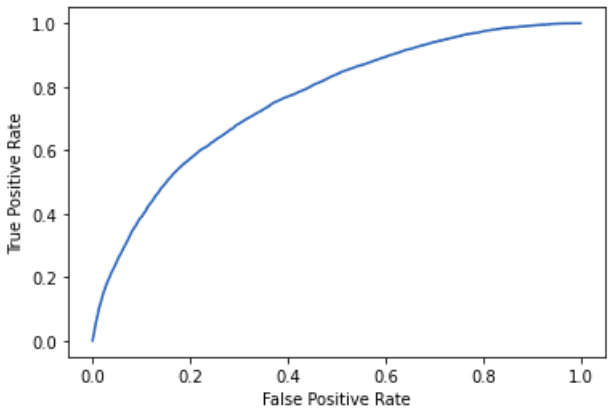


Figura 15: Curva ROC de *Naive Bayes*

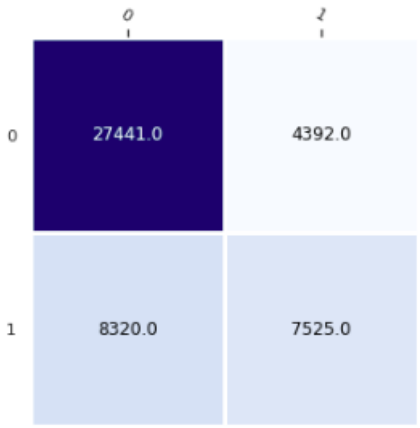


Figura 16: Matriz de confusão de *Naive Bayes*

6.2 *Decision Tree*

O classificador de *Decision Tree* obteve as seguintes medidas:

Modelo decisionTree

	precision	recall	f1-score	support
0.0	0.84	0.88	0.86	31833
1.0	0.74	0.67	0.70	15845
accuracy			0.81	47678
macro avg	0.79	0.78	0.78	47678
weighted avg	0.81	0.81	0.81	47678

Figura 17: Medidas de desempenho do classificador de *Decision Tree*

A sua matriz de confusão e curva ROC, respectivamente:

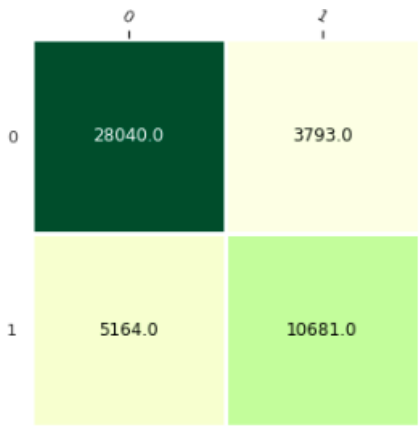


Figura 18: Matriz de confusão de *Decision Tree*

6.3 *KNN*

O classificador de KNN obteve as seguintes medidas:

A sua matriz de confusão e curva ROC, respectivamente:

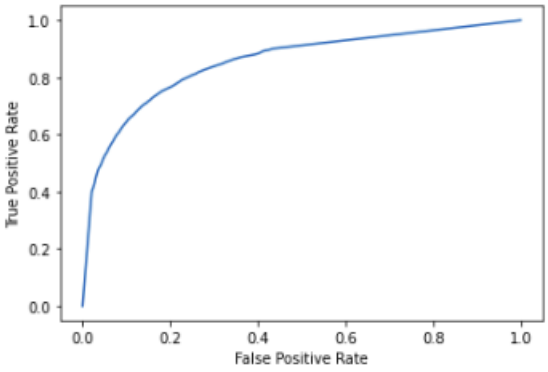


Figura 19: Curva ROC de *Decision Tree*

Modelo KNN

	precision	recall	f1-score	support
0.0	0.85	0.85	0.85	31833
1.0	0.69	0.70	0.70	15845
accuracy			0.80	47678
macro avg	0.77	0.77	0.77	47678
weighted avg	0.80	0.80	0.80	47678

Figura 20: Medidas de desempenho do classificador de KNN

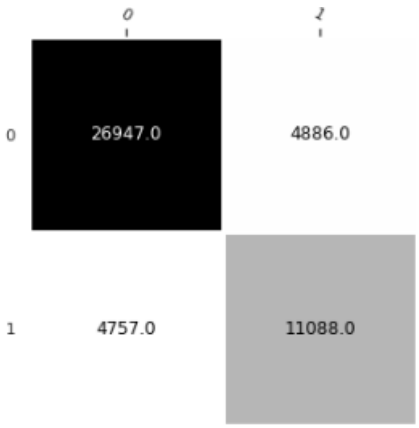


Figura 21: Matriz de confusão de KNN

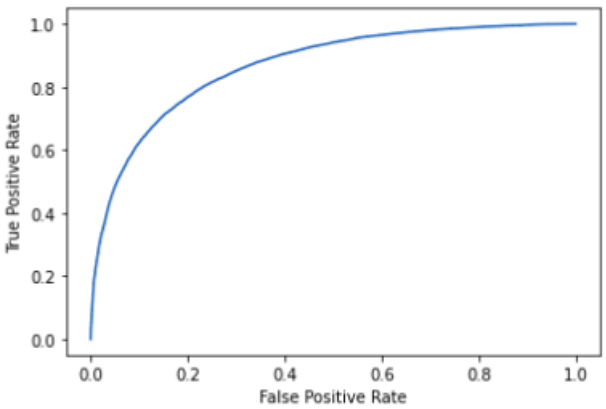


Figura 22: Curva ROC de KNN

6.4 *Regressão Logística*

O classificador de Regressão Logística obteve as seguintes medidas:

Modelo logisticRegression

	precision	recall	f1-score	support
0.0	0.79	0.95	0.86	31833
1.0	0.82	0.50	0.62	15845
accuracy			0.80	47678
macro avg	0.81	0.72	0.74	47678
weighted avg	0.80	0.80	0.78	47678

Figura 23: Medidas de desempenho do classificador de Regressão Logística

A sua matriz de confusão e curva ROC, respectivamente:

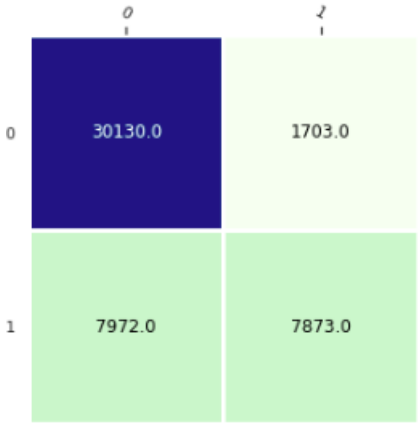


Figura 24: Matriz de confusão de Regressão Logística

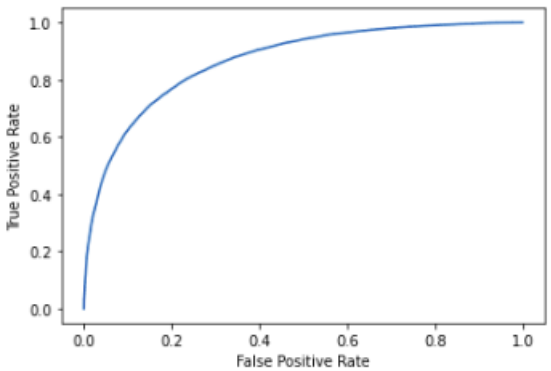


Figura 25: Curva ROC de Regressão Logística

6.5 *SVM*

O classificador de SVM obteve as seguintes medidas:

A sua matriz de confusão e curva ROC, respectivamente:

Modelo naiveBayes

	precision	recall	f1-score	support
0.0	0.77	0.86	0.81	31833
1.0	0.63	0.47	0.54	15845
accuracy			0.73	47678
macro avg	0.70	0.67	0.68	47678
weighted avg	0.72	0.73	0.72	47678

Figura 26: Medidas de desempenho do classificador de SVM

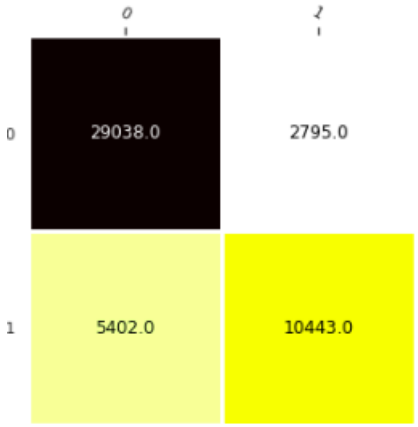


Figura 27: Matriz de confusão de SVM

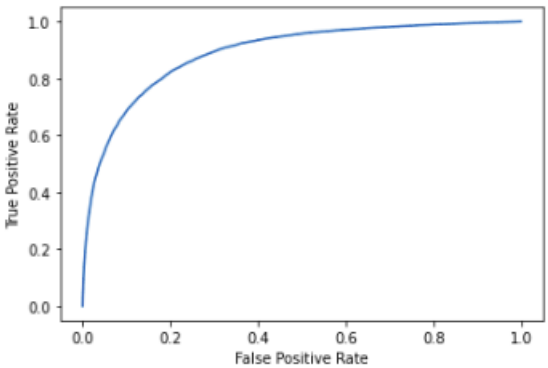


Figura 28: Curva ROC de SVM

6.6 *Random Forest*

O classificador de Random Forest obteve as seguintes medidas:

Modelo randomForest

	precision	recall	f1-score	support
0.0	0.86	0.92	0.89	31833
1.0	0.82	0.69	0.75	15845
accuracy			0.84	47678
macro avg	0.84	0.80	0.82	47678
weighted avg	0.84	0.84	0.84	47678

Figura 29: Medidas de desempenho do classificador de Random Forest

A sua matriz de confusão e curva ROC, respectivamente:

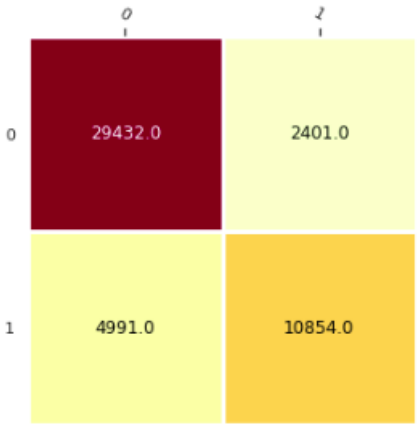


Figura 30: Matriz de confusão de Random Forest

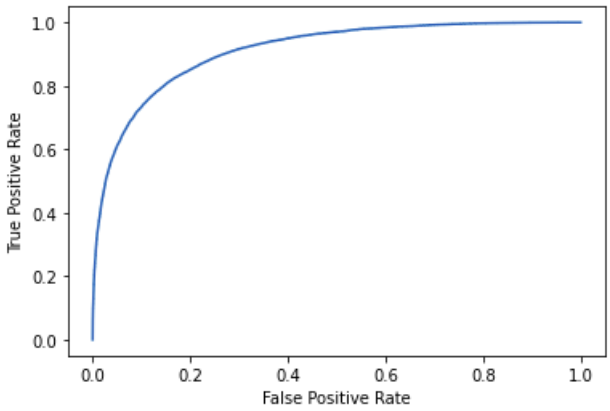


Figura 31: Curva ROC de Random Forest

6.7 *AutoML*

O classificador que utiliza a ferramenta automática auto-sklearn obteve as seguintes medidas:

A sua matriz de confusão e curva ROC, respectivamente:

Modelo Auto-ML

	precision	recall	f1-score	support
0.0	0.84	0.89	0.86	31833
1.0	0.75	0.66	0.70	15845
accuracy			0.81	47678
macro avg	0.79	0.77	0.78	47678
weighted avg	0.81	0.81	0.81	47678

Figura 32: Medidas de desempenho do classificador de Random Forest

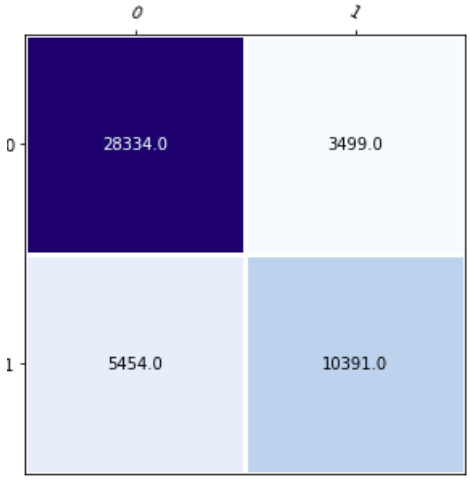


Figura 33: Matriz de confusão de Random Forest

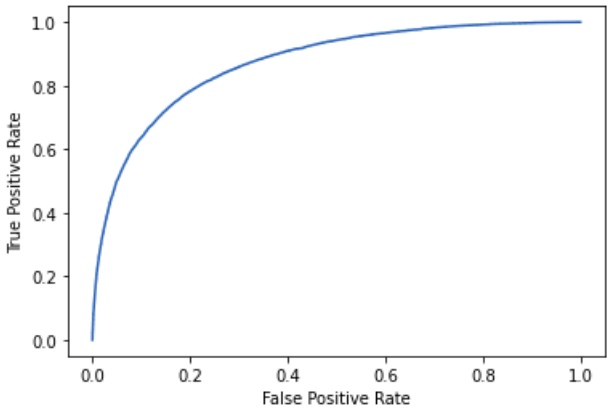


Figura 34: Curva ROC de Random Forest

7 Conclusão

Após o treinamento e testagem dos 7 modelos, foi possível observar que o modelo que obteve a melhor acurácia foi o que utiliza o método ensemble de random forest, alcançando 0,84. Além disso, foi possível observar que todos os modelos obtiveram bons resultados de acurácia, indicando que é possível extrair padrões da natureza relacionando variáveis metereológicas e que o aprendizado de máquina pode ser utilizado como instrumento para auxiliar nas previsões metereológicas.

O resultado de todo o trabalho pode ser verificado no Google Colab a seguir:

`https://colab.research.google.com/drive/1KUSiUVvrfmA3HIthjj_GNs2V6jhE3MNQ?usp=sharing`