Universidade Federal do Rio Grande do Norte
Departamento de Engenharia Elétrica
ELE0606 - TOPICOS ESPECIAIS EM INTELIGENCIA ARTIFICIAL

Classificações com Árvores de Decisão Utilizando o Modelo Random Forest

Aluno: Pedro Artur F. Varela de Lira
Professor: José Alfredo Ferreira Costa

Setembro
2023

Conteúdo

1 Introdução e Contexto

A classificação de dados é uma tarefa essencial em aprendizado de máquina e análise de dados, que desempenha um papel crucial em diversas aplicações, desde diagnósticos médicos até previsões financeiras. Neste relatório, apresentaremos os resultados de um experimento de classificação realizado utilizando a biblioteca tensorflow_decision_forests e o modelo Random Forest para analisar duas bases de dados: Wine, disponível na biblioteca Scikit-Learn, e Heart Disease, obtida a partir do Kaggle.

O objetivo fundamental deste experimento é explorar a eficácia do algoritmo Random Forest, uma técnica de aprendizado de máquina que se destaca pela sua capacidade de lidar com uma ampla variedade de tipos de dados e tarefas de classificação. Investigaremos como esse modelo se comporta em relação a duas bases de dados distintas e os insights que podemos obter a partir da sua aplicação.

As bases de dados escolhidas para este estudo possuem características únicas:

- Wine Dataset: Este conjunto de dados consiste em informações sobre vinhos, como teor alcoólico, acidez e outros atributos relacionados a análise química de vinhos. O objetivo é classificar os vinhos em três classes diferentes com base nessas características. Essa tarefa é relevante, por exemplo, para a indústria vinícola na identificação de vinhos de alta qualidade.

- Heart Disease Dataset: A base de dados de doenças cardíacas reúne informações clínicas e biomédicas sobre pacientes para prever a presença de doenças cardíacas. É uma tarefa crucial para a área médica, já que a detecção precoce de doenças cardíacas pode salvar vidas e melhorar a qualidade de vida dos pacientes.

Este relatório está organizado da seguinte forma: na seção seguinte, discutiremos brevemente a metodologia do trabalho, o pseudocódigo, os resultados obtidos e, por fim, serão apresentadas conclusões e discussões.

O estudo visa fornecer uma análise aprofundada das capacidades do modelo Random Forest, bem como uma compreensão mais profunda das características e desafios das bases de dados Wine e Heart Disease. Essa análise pode ser valiosa tanto para iniciantes quanto para profissionais experientes em aprendizado de máquina, fornecendo insights valiosos sobre a aplicação dessa técnica em problemas reais de classificação..

2 Metodologia

Utilizou-se a liguagem de progração Python, inserida em um ambiente de Jupyter Notebook disponibiizada pelo Kaggle. As principais etapas do código incluem:

- **Análise dos Dados e Correlação**: Carregar os conjuntos de dados Wine e Heart Disease; realizar uma análise exploratória dos dados para entender suas características, incluindo estatísticas descritivas; calcular a matriz de correlação entre os atributos dos dados para identificar possíveis relações entre eles.
- **Preparação dos Dados**: Pré-processar os dados, incluindo a remoção de valores ausentes e a codificação de variáveis categóricas, se necessário; dividir os dados em conjuntos de treinamento e teste para avaliar o modelo.
- **Treinamento do Modelo Random Forest**: Configurar um modelo Random Forest usando a biblioteca tensorflow_decision_forests; treinar o modelo com o conjunto de treinamento.
- **Análise de uma Árvore de Decisão**: Visualizar uma árvore de decisão do modelo Random Forest, se a biblioteca tensorflow_decision_forests permitir essa funcionalidade.
- **Avaliação do Modelo**: É necessário avaliar o modelo com as métricas acurácia e mean square error (mse)

A metodologia foi a mesma para a base de dados Wine.

3 Pseudocódigo

O pseudocódigo abaixo reflete as principais etapas do código implementado no experimento, desde a importação das bibliotecas até a análise dos resultados de acurácia e mse.

```
# Importar as bibliotecas necessarias
import tensorflow as tf
import tensorflow_decision_forests as tfdf
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

```

```
# Carregar o conjunto de dados "heart.csv"
heart_df = pd.read_csv('caminho/para/o/arquivo/heart.csv')

# Realizar analise exploratoria dos dados
# Verificar informacoes do dataset
heart_df.info()

# Gerar estatisticas descritivas
heart_df.describe()

# Calcular a matriz de correlacao e gerar um heatmap
corr = heart_df.corr()
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask)] = True
cmap = sns.diverging_palette(220, 10, as_cmap=True)
# Gerar o grafico de heatmap
# ...

# Remover as colunas 'chol' e 'fbs' do DataFrame
heart_df.drop(['chol', 'fbs'], axis=1, inplace=True)

# Separar os dados em conjuntos de treinamento e teste
train_ds_pd, valid_ds_pd = split_dataset(heart_df)

# Converter os DataFrames pandas em tensores
label = 'target'
train_ds = pd_dataframe_to_tf_dataset(train_ds_pd, label=
    label)
valid_ds = pd_dataframe_to_tf_dataset(valid_ds_pd, label=
    label)

# Criar e compilar o modelo Random Forest
rf = RandomFrestModel(num_trees=200)
rf.compile(metrics=["accuracy", "mse"])

# Treinar o modelo Random Forest
rf.fit(x=train_ds)

# Visualizar uma arvore de decisao especifica (indice 0)
plot_model_in_colab(rf, tree_idx=0, max_depth=3)

# Analisar a acuracia em funcao do numero de arvores
logs = rf.make_inspector().training_logs()
# Gerar grafico da acuracia vs. numero de arvores
# ...

# Avaliar a acuracia final do modelo
evaluation = rf.evaluate(x=valid_ds, return_dict=True)
for name, value in evaluation.items():
```

```
56  print(f"{name}: {value:.4f}")
```

Listing 1: Pseudocódigo

É importante ressaltar que o pseudocódigo não foi implementado para funcionar efetivamente como solução do problema, mas sim como um meio de representar facilmente as etapas principais realizadas no Jupyter Notebook. O código para o projeto completo e o resultados estão no link: `https://github.com/pedro-varela1/Arquivos_ELE-606/blob/main/Atividade_4-heart-desease-ra ipynb`.

Para a base de dados Wine, o pseudocódigo foi o mesmo, e o Jupyter Notebook está disponível em `https://github.com/pedro-varela1/Arquivos_ELE-606/blob/main/Atividade_3_DT_ELE606.ipynb`.

4 Resultados

Para realizar a validação geral do modelo, foram realocados 70% o dataset Wine para treinamento e 30% para teste e foram medidos a acurácia e mse.

No gráfico da Figura 1, vemos que, quanto maior o número de árvores, maior a acurácia do modelo para o base de dados *Heart Deasease*, isso também acontece para a base de dados *Wine*, como vemos na Figura 2. Porém, é importante ressaltar que isso não acontece indefinidamente, depois de uma certa quantidade de árvores, a acurácia vai se estabilizando. Portanto, para economizar custo computacional, é necessário fazer análises profundas da quantidades de árvores adequadas para o problema.

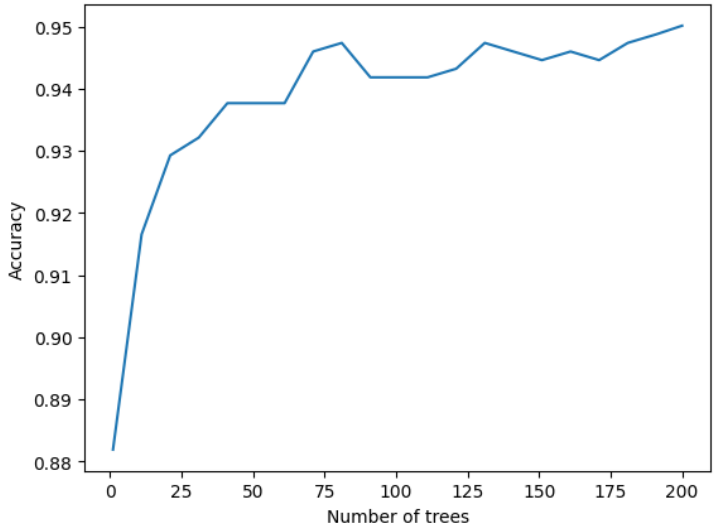

Figura 1: Acurácia do modelo com a base de dados Heart Desease em função do número de árvores

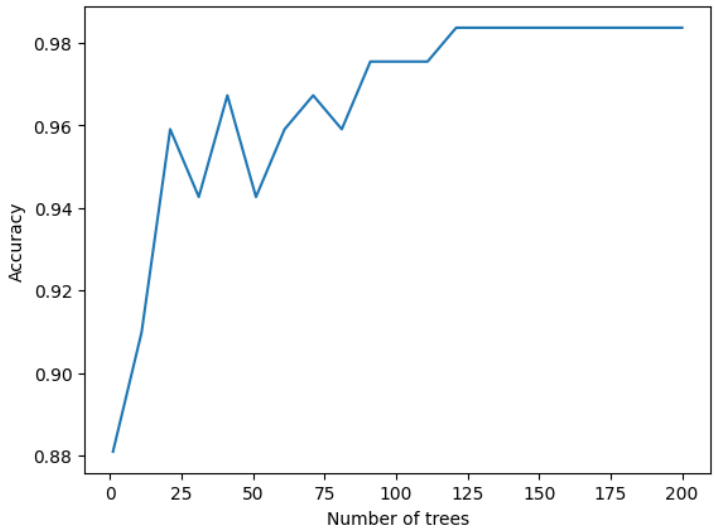

Figura 2: Acurácia do modelo com a base de dados Wine em função do número de árvores

Podemos analisar as árvores de decisão de índice 0, com uma profundidade de 3, para evitar poluir a imagem, nas Figuras 3 e 4.

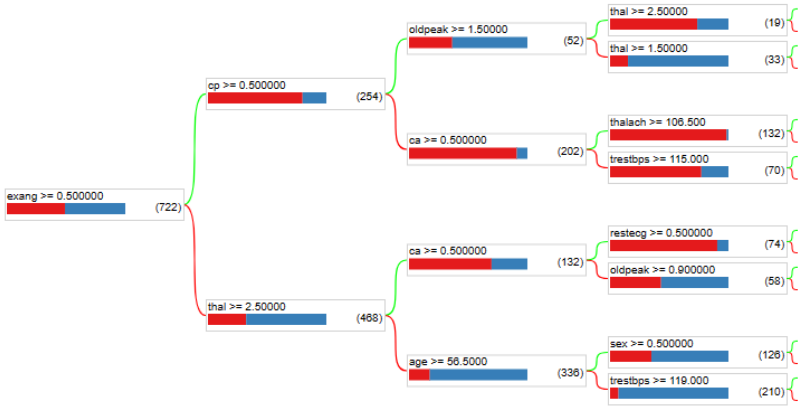

Figura 3: Árvore de Índice 0 e com profundidade 3 para análise Heart Deasease

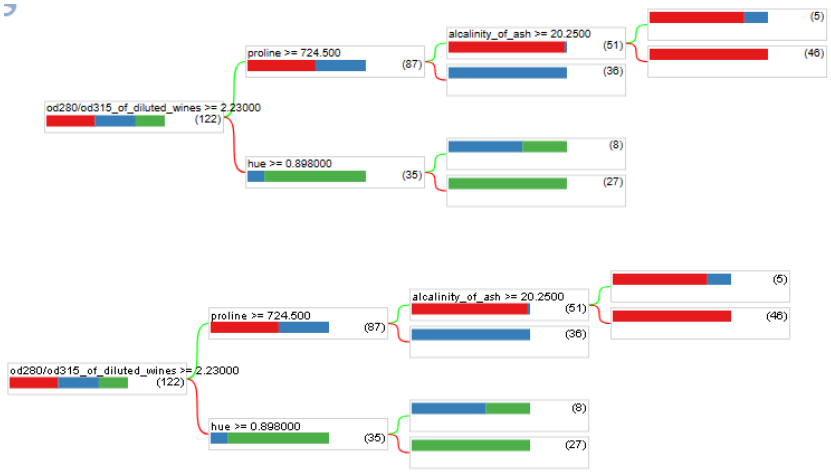

Figura 4: Árvore de Índice 0 e com profundidade 3 para análise Wine

Por fim, as métricas para análise dos resultados finais: acurácia e *mean square error* são dadas por:

```
1/1 [==============================]
loss: 0.0000
accuracy: 0.9703
mse: 0.0393
```

Figura 5: Métricas finais para o Heart Desease

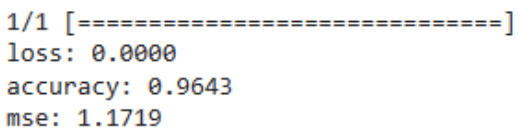

Figura 6: Métricas finais para o Wine

Para a ánalise da base de dados *Wine* a acurácia foi de 96.43% e para a base de dados *Heart Desease* foi de 97.03%. O *mean square error* foi de 1.1719 para a base de dados *Wine* e 0.0393 para a *Heart Desease*.

5 Conclusão e Discussões

Neste relatório, realizamos uma análise abrangente da aplicação do modelo Random Forest utilizando a biblioteca *tensorflow_decision_forests* em duas bases de dados distintas: Wine e Heart Disease. A seguir, destacamos as principais conclusões e discussões com base nos resultados obtidos.

Observamos que o aumento do número de árvores no modelo Random Forest teve um impacto positivo na acurácia para ambas as bases de dados Wine e Heart Disease. Esse comportamento é evidenciado nos gráficos da Figura 1 e Figura 2. No entanto, é importante notar que, após atingir uma certa quantidade de árvores, a melhoria na acurácia diminui e o modelo começa a se estabilizar. Essa informação é crucial para otimizar o custo computacional, uma vez que treinar um grande número de árvores pode ser dispendioso em termos de recursos.

Para melhor compreensão do modelo, analisamos as árvores de decisão de índice 0 com uma profundidade de 3 nas Figuras 3 e 4. Essa análise nos permitiu examinar as regras de decisão que o modelo estava usando para classificar os dados. No entanto, devido à limitação de espaço, não apresentamos as árvores completas neste relatório. Essa análise pode ser valiosa para insights específicos sobre como o modelo toma decisões.

As métricas de avaliação utilizadas para medir o desempenho do modelo foram a acurácia e o Mean Square Error (MSE). Para a base de dados Wine, o modelo alcançou uma acurácia de 96.43% e um MSE de 1.1719. Para a base de dados Heart Disease, a acurácia foi de 97.03% e o MSE foi de 0.0393. Esses resultados indicam que o modelo Random Forest foi capaz de realizar uma classificação altamente precisa em ambas as bases de dados.

O uso do modelo Random Forest e da biblioteca *tensorflow_decision_forests* mostrou-se eficaz na classificação de dados, demonstrando um alto desempenho em ambas as bases de dados Wine e Heart Disease. No entanto, é

importante lembrar que a escolha do número adequado de árvores é fundamental para equilibrar a precisão do modelo e os recursos computacionais necessários. Além disso, as métricas de avaliação destacam a qualidade do modelo em fazer previsões precisas para problemas de classificação e regressão.

Em resumo, este experimento demonstra a versatilidade e eficácia do modelo Random Forest na solução de problemas de classificação de dados, proporcionando insights valiosos para futuros projetos e aplicações práticas. A análise detalhada das métricas e das árvores de decisão pode servir como base para otimizações futuras e aprimoramento do desempenho do modelo em cenários do mundo real.

Bibliografia

IBM. (s.d.). What is random forest?. Disponível em: https://www.ibm.com/topics/random-forest. Acesso em: 18/09/2023.

TENSORFLOW. (s.d.). TensorFlow Decision Forests. Disponível em: https://www.tensorflow.org/decision_forests?hl=pt-br. Acesso em: 18/09/2023.