Universidade do Rio de Janeiro
Tópicos Especiais em Matemática e Computação
Trabalho 1

Matheus Lopes Marendino

05/05/2024

1 Descrição dos Dados

1.1 Informações Relevantes

O conjunto de dados do Titanic contém informações sobre os passageiros do famoso navio Titanic, incluindo características como idade, sexo, classe de passageiro, tarifa paga, número de parentes a bordo, entre outros, bem como se o passageiro sobreviveu ou não ao naufrágio. A base utilizada contém 1309 amostras e cada uma representa os dados de cada passageiro embarcado no cruzeiro.

1.2 Atributos do banco de dados

PassengerId: Identificação única de cada passageiro.
Survived: Indica se o passageiro sobreviveu (1) ou não (0).
Pclass: Classe de passageiro (1ª, 2ª ou 3ª).
Name: Nome do passageiro.
Sex: Sexo do passageiro.
Age: Idade do passageiro.
SibSp: Número de irmãos/cônjuges a bordo.
Parch: Número de pais/filhos a bordo.
Ticket: Número do bilhete.
Fare: Tarifa paga pelo passageiro.
Cabin: Número da cabine do passageiro.
Embarked: Porto de embarque do passageiro (C = Cherbourg, Q = Queenstown, S = Southampton).
A variável-alvo (target) é ”Survived”, que indica se o passageiro sobreviveu (1) ou não (0).

1.3 Desbalanceamento

É importante notar que a variável-alvo ”Survived”é desbalanceada, ao ultilizar a função value.counts() temos que 494 passageiros sobreviveram e 815 não sobreviveram, ou seja, mais passageiros não sobreviveram. Isso pode afetar a performance de certos algoritmos de aprendizado de máquina.

1.4 Valores Nulos e/ou Faltantes

Durante a análise dos dados, percebe-se que o atributo ’Age’ contém 263 valores faltantes. Para lidar com os valores faltantes das idades, foi utilizada a média da idade dos passageiros, que era

de 29 anos.
O atributo 'Cabin' contém 1014 valores faltantes. Devido à grande quantidade de valores faltantes e ao fato de o atributo conter o número da cabine em que o passageiro estava hospedado, é impossível determinar a cabine de cada passageiro. Portanto, o atributo foi excluído da base de dados.
O atributo 'Embarked' contém 2 valores faltantes. Após imprimir quais passageiros estavam com valores faltantes no atributo 'Embarked', pesquisei manualmente na Wiki seus respectivos nomes e constatei que os 2 passageiros tinham embarcado no porto de Southampton. Assim, pude completar os valores manualmente com o caractere 'S'.
E 'Fare' contém 13 valores faltantes, que também foi preenchida com a média dos valores.

1.5 Outliers (boxplot)

Para descobrir se existem outliers, foi utilizada o Boxplot no código. Gerando o seguinte gráfico da figura 1:

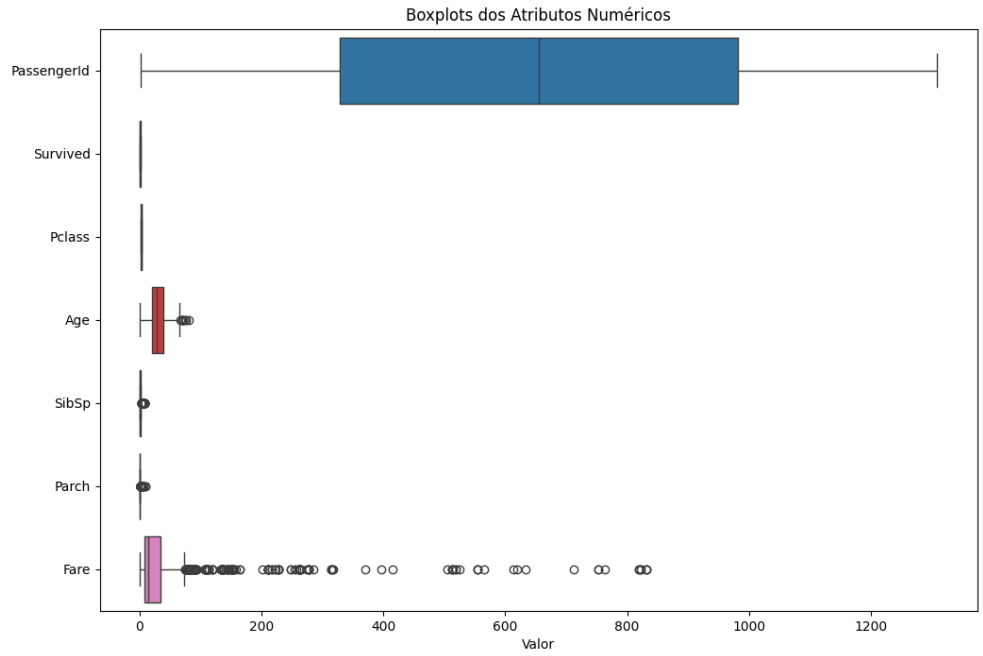

Figura 1:Gráfico Outliers

Analisando o gráfico do Boxplot na Figura 1, percebe-se que existem outliers nos atributos numéricos 'Age', 'Fare', 'Parch' e 'SibSp'

2 Análise dos Dados

2.1 Resumo estatístico

A tabela 1 contém os valores estastíticos dos atributos da base de dados, como valor médio, desvio padrão e variância de cada atributos. Pode ser um forte aliado ao fazer uma primeira avaliação da

base de dados.

Variável	Valor médio	Desvio padrão	Variância
PassengerId	655.000000	378.020061	142899.166667
Survived	0.377387	0.484918	0.235146
Pclass	2.294882	0.837836	0.701969
Age	29.881138	14.413493	207.748787
SibSp	0.498854	1.041658	1.085052
Parch	0.385027	0.865560	0.749195
Fare	48.624269	104.436107	10906.900384

Tabela 1: Valores da Série

2.2 Matriz de correlação

Na Tabela 2, temos a matriz de correlação e podemos identificar como e quais atributos se relacionam entre si. Como exemplo, temos 'Fare' e 'Pclass'. Uma vez que um passageiro(a) pagou por uma classe superior, a tarifa do seu bilhete foi mais cara, e por isso a correlação entre os atributos. Outro exemplo seriam os atributos 'SibSp' e 'Parch', indicando que passageiros que embarcaram com seus irmãos e/ou cônjuges tendem a viajar junto aos pais e/ou filhos."

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.020370	-0.038354	0.028814	-0.055224	0.008942	0.183446
Survived	-0.020370	1.000000	-0.264710	-0.053695	0.002370	0.108919	0.144137
Pclass	-0.038354	-0.264710	1.000000	-0.408106	0.060832	0.018322	-0.420944
Age	0.028814	-0.053695	-0.408106	1.000000	-0.243699	-0.150917	0.143882
SibSp	-0.055224	0.002370	0.060832	-0.243699	1.000000	0.373587	0.101077
Parch	0.008942	0.108919	0.018322	-0.150917	0.373587	1.000000	0.140362
Fare	0.183446	0.144137	-0.420944	0.143882	0.101077	0.140362	1.000000

Tabela 2: Matriz de Correlação

2.3 Tabela de valor máximo e valor mínimo

A tabela 3, referente aos valores máximos e mínimos foi um forte aliado para a avaliação dos outliers que foram identificados durante o uso do boxplot, ajudando a analisar os valores de cada atributo separadamente.

Variável	Valor Mínimo	Valor Máximo
PassengerId	1.00	1309.000
Survived	0.00	1.000
Pclass	1.00	3.000
Age	0.17	80.000
SibSp	0.00	8.000
Parch	0.00	9.000
Fare	0.00	831.583

Tabela 3: Valores Mínimos e Máximos

3 Metodologia

3.1 PSO e DE

Para otimizar o desempenho do método proposto, foi utilizada as técnicas de otimização Differential Evolution (DE) e Particle Swarm Optimization (PSO). Evolução diferencial é um método que utiliza estratégias de mutação, recombinação e seleção para encontrar soluções ótimas em espaços de busca complexos. Por outro lado, Enxame de partículas é uma técnica baseada em modelos de comportamento de enxame, onde partículas movem-se no espaço de busca para encontrar a melhor solução global.

3.2 Kfold

Foi empregado a técnica de Validação Cruzada K-Fold. Essa abordagem divide o conjunto de dados em K partes (ou folds), treinando o modelo K vezes. Em cada iteração, um fold diferente é utilizado como conjunto de validação, enquanto os demais são usados para treinamento. Isso nos permite avaliar a capacidade de generalização do modelo de forma mais abrangente do que uma simples divisão treino-teste.

3.3 Smote

Fou utilizado a técnica de SMOTE (Synthetic Minority Over-sampling Technique) para lidar com o desbalanceamento de classes nos dados. SMOTE gera novas amostras sintéticas da classe minoritária, melhorando assim a capacidade do modelo de aprender com dados desbalanceados.

3.4 RFE

Para garantir uma seleção robusta de features e otimizar, foi empregado o método de Recursive Feature Elimination (RFE). RFE é uma abordagem que recursivamente remove as features menos importantes, treinando o modelo com o subconjunto remanescente de features até que o número desejado de features seja alcançado. Isso ajuda a melhorar a eficiência computacional e pode levar a um modelo mais simples. O número de características selecionadas foram 5 e as principais são 'Pclass', 'Age', 'SibSp', 'Fare', 'MaleCheck'.

3.5 Árvore de Decisão

A Árvore de Decisão é um algoritmo de aprendizado de máquina supervisionado que é utilizado para classificação e para regressão. Como proposto, foi ultilizado a Árvore de Decisão como modelo base para a análise dos dados.
Após a aplicação das técnicas anteriores, encontramos os melhores hiperparâmetros para a Árvore de Decisão, permitindo assim uma avaliação confiável do seu desempenho.
Avaliamos o desempenho da Árvore de Decisão utilizando métricas como acurácia, recall e F1-score garantindo uma boa análise do modelo.

4 Experimentos Computacionais

4.1 Parâmetros do Modelo de Árvore de Decisão

Para utilizar o modelo de árvore de decisão, foram selecionados de acordo com a tabela 4, os parâmetros max_depth, min_samples_split, min_samples_leaf .

Parâmetro	Limite Inferior	Limite Superior
max_depth	2	10
min_samples_split	2	20
min_samples_leaf	1	20
Parâmetros da DE e PSO	**Nome**	**Limite**
Número de gerações	maxiter	100
Tamanho população	popsize	50

Tabela 4: Limites dos parâmetros do modelo de árvore de decisão e Parâmetros do PSO e DE

O parâmetro max_depth foi configurado com valores variando de 2 a 10. Este parâmetro controla a profundidade máxima da árvore, ou seja, o número máximo de níveis que a árvore pode ter. A profundidade de uma árvore de decisão se refere ao número de camadas que a árvore possui, que vai desde o nó da raiz até as suas folhas. Cada camada representa uma decisão baseada em uma característica específica dos dados. Uma profundidade maior significa que a árvore pode fazer mais divisões, capturando mais detalhes nos dados de treinamento.

O parâmetro min_samples_split foi configurado com valores variando de 2 a 20. Este parâmetro define o número mínimo de amostras necessárias para dividir um nó. Isso significa que um nó interno precisa ter pelo menos esse número de amostras para ser considerado para divisão. Valores maiores podem resultar em árvores mais gerais, enquanto valores menores podem fazer com que a árvore se ajuste muito aos dados de treinamento.

O parâmetro min_samples_leaf foi configurado com valores variando de 1 a 20. Este parâmetro especifica o número mínimo de amostras que um nó folha deve ter. Um valor maior pode ajudar a suavizar a árvore, evitando que ela se ajuste muito aos dados de treinamento e melhorando a generalização.

4.2 Resultados

4.2.1 Média e Desvio padrão da Evolução Diferencial

De acordo com a tabela 5, percebe-se que houve uma ligeira melhora nos resultados dos dados pré-processados em relação aos dados que não foram processados. Tendo uma média de 88% para as métricas com os dados pré-processados, e para os dados originais tivemos uma média de 86% para todas as métricas.

Métrica	Tipo de Dados	Média $\pm$ Desvio Padrão
ACCURACY	Pré-processados	0.8880 ± 0.0025
F1	Pré-processados	0.8880 ± 0.0024
RECALL	Pré-processados	0.8880 ± 0.0025
ACCURACY	Originais	0.8698 ± 0.0018
F1	Originais	0.8684 ± 0.0018
RECALL	Originais	0.8698 ± 0.0018

Tabela 5: Resultados Médios e Desvios Padrão

4.2.2 Melhores resultados Evolução Diferencial

De acordo com a tabela 6, após o pré-processamento dos dados, o melhor modelo alcançou uma acurácia de 89.33%, um F1 Score de 89.31% e um recall de 89.33%. Esse modelo foi ajustado com os seguintes parâmetros: max_depth = 8, min_samples_split = 4 e min_samples_leaf = 1 .
Em comparação, o modelo com os dados originais mostrou resultados ligeiramente menores, uma acurária de 87.47%, um F1 Score de 87.29% e um recall de 87.47%. Este modelo utilizou os seguintes parâmetros: max_depth = 6, min_samples_split = 4 e min_samples_leaf = 3 .

Métrica	**Pré-processados**	**Originais**
Parâmetros	{'max_depth': 8, 'min_samples_split': 4, 'min_samples_leaf': 1 }	{'max_depth': 6, 'min_samples_split': 4, 'min_samples_leaf': 3}
Acurácia	0.8933	0.8747
F1 Score	0.8931	0.8729
Recall	0.8933	0.8747

Tabela 6: Melhores Modelos

4.2.3 Média e desvio padrão Enxame de Partículas

De acordo com a tabela 7, após o pré-processamento dos dados, tivemos que a média das métricas acurácia, f1 e Recall foi de 88.7%. Enquanto no modelo sem pré-processamento de dados tivemos médias de 87% para acurácia e Recall e de 86.9% para o F1. Mostrando que após o pré-processamento de dados houve uma ligeira melhora nos resultados

Métrica	**Tipo de Dados**	**Média $\pm$ Desvio Padrão**
ACCURACY	Pré-processados	0.8871 ± 0.0009
F1	Pré-processados	0.8871 ± 0.0009
RECALL	Pré-processados	0.8871 ± 0.0009
ACCURACY	Originais	0.8709 ± 0.0022
F1	Originais	0.8696 ± 0.0021
RECALL	Originais	0.8709 ± 0.0022

Tabela 7: Resultados Médios e Desvios Padrão

4.2.4 Melhores resultados Enxame de Partículas

De acordo com a tabela 8, após o pré-processamento dos dados, o melhor modelo encontrou o resultado de 88.7% para todas as métricas avaliadas, ele foi ajustado com os seguintes parâmetros: max_depth = 10, min_samples_split = 2 e min_samples_leaf = 3.
Em comparação, o modelo com os dados originais mostrou resultados piores, uma acurária de 87.2%, um F1 Score de 87.1% e um recall de 87.2%, foi utilizado os seguintes parâmetros: max_depth = 4, min_samples_split = 16 e min_samples_leaf = 1 .

Métrica	Pré-processados	Originais
Parâmetros	{'max_depth': 10, 'min_samples_split': 2, 'min_samples_leaf': 3}	{'max_depth': 4, 'min_samples_split': 16, 'min_samples_leaf': 1 }
Acurácia	0.8877	0.8724
F1 Score	0.8877	0.8711
Recall	0.8877	0.8724

Tabela 8: Melhores Modelos

5 Conclusão

Neste trabalho, utilizamos as técnicas de Evolução Diferencial (DE) e Enxame de Partículas (PSO) para otimizar os hiperparâmetros de um modelo de Árvore de Decisão aplicado ao conjunto de dados Titanic, e comparamos os resultados dos dados pré-processados utilizando o kfold, o método RFE para seleção de características e o SMOTE para balancear dos dados. Avaliamos o desempenho dos modelos em termos de acurácia, F1 Score e Recall, tanto para dados pré-processados quanto para dados originais.

Os resultados mostram que o pré-processamento dos dados levou a uma pequena melhoria dos resultados das métricas avaliadas.
Para a técnica de Evolução Diferencial, a média das métricas nos dados pré-processados foi de 88%, enquanto nos dados originais foi de 86%. O melhor modelo para os dados pré-processados encontrou o melhor valor de 89.3% para todas as métricas e para os dados originais o melhor modelo teve 87% para acurácia e f1, e 86.9%.

Para a técnica de Enxame de Partículas, observamos uma média de 88.7% para as métricas nos dados pré-processados, em comparação com 87% para acurácia e Recall e 86.9% para F1 nos dados originais. O melhor modelo obteve o resultado de 88.7% para todas as métricas avaliadas, em contrapartida o melhor modelo dos dados originais obteve um resultado de 87.2%

O uso de DE e PSO demonstrou ser eficaz na otimização dos hiperparâmetros do modelo de Árvore de Decisão, obtendo um desempenho satisfatório. Além disso, o pré-processamento dos dados provou trzer melhorias para os resultados, mostrando que um bom preparo dos dados pode vir a trazer melhoria das métricas. Estes resultados reforçam a validade dessas técnicas de otimização e a importância do pré-processamento no desenvolvimento dos modelos preditivos.

Referências

https://www.kaggle.com/c/titanic/data
https://www.encyclopedia-titanica.org/titanic-survivor/amelia-icard.html
https://www.encyclopedia-titanica.org/titanic-survivor/martha-evelyn-stone.html
https://quantdare.com/decision-trees-gini-vs-entropy/