Análise de Modelos de Classificação em Datasets misteriosos : Árvore de decisão, KNN e MLP.

Eduardo Schvinn
Joinville, Brasil
Email: eschvinn@gmail.com

I. Introdução

Neste trabalho, foram avaliados três algoritmos de classificação (K-Nearest Neighbors, Perceptron Multicamadas e Árvore de Decisão) utilizando o conjunto de dados '12.csv', fornecido sem informações adicionais. Aplicou-se técnicas de aprendizado de máquina e foi realizada uma comparação entre os classificadores, utilizando métricas de desempenho como acurácia, precisão, recall e F1-score. A seleção de hiperparâmetros foi otimizada com GridSearchCV, e uma análise exploratória foi conduzida para identificar outliers e padrões nos dados. Nosso objetivo é identificar o classificador com melhor desempenho e analisar as vantagens e limitações de cada método, contribuindo para a escolha informada de algoritmos em futuros projetos de aprendizado de máquina.

II. Pré-processamento

Durante o pré-processamento,foi verificado se existiam colunas nulas e dados duplicados, após isso foi testou-se a presença de outliers em cada linha do conjunto de dados.E foi identificado que as Linhas 19 e 70 apresentavam outliers, afetando mais de 15% das colunas. Essas linhas foram então removidas. Antes da remoção, o conjunto de dados continha 47 instâncias da Classe 1 e 25 instâncias da Classe 2. Após a remoção dos outliers, a distribuição foi ajustada para 46 instâncias da Classe 1 e 24 instâncias da Classe 2 como mostra a tabela a seguir:

Classe	Antes da Remoção	Após a Remoção
Classe 1	47	46
Classe 2	25	24

TABLE I: Distribuição de Classes antes e depois da remoção de outliers

Para normalizar os dados, foi utilizada a função normaliza_dataframe que emprega o StandardScaler da biblioteca scikit-learn. Primeiramente, os dados de treino (X_treino) são ajustados ao scaler para calcular a média e o desvio padrão de cada atributo. Em seguida, aplicou-se essa transformação aos dados de treino para centralizá-los em torno de zero e escalá-los para que tenham variância unitária. Para os dados de teste (X_test), foi utilizado o mesmo scaler ajustado com base nos dados de treino, garantindo que a escala aplicada seja consistente entre os conjuntos de treino e teste.

Para selecionar os melhores atributos, utilizou-se a função seleciona_melhores, que emprega o método SelectKBest do scikit-learn com o score function f_classif para selecionar os melhores atributos baseados na análise de variância. Primeiramente, os dados de treino normalizados (X_treino_normalizado) são ajustados ao seletor para calcular os scores de cada atributo em relação à variável alvo (y_treino) e selecionar os 54 melhores atributos , o valor 'k = 54' na função foi definido apartir de testes com diferentes valores de 'k'. Em seguida, foi aplicada essa seleção tanto aos dados de treino quanto aos dados de teste normalizados (X_test_normalizado).

III. Base Line

Para estabelecer um baseline a ser superado, inicialmente foi implementado um classificador dummy que prediz a classe mais frequente no conjunto de dados, atribuindo essa classe a todos os exemplos. Esse método resultou em uma taxa de acerto de 57.14%. Além disso, como segundo baseline, empregamos um modelo de Random Forest sem nenhum pré-processamento adicional dos dados, alcançando uma taxa de acerto de 76.19%. Esses resultados iniciais servem como referência para avaliar o desempenho de modelos mais sofisticados que serão desenvolvidos.

IV. GridSearchCv

Para encontrar os melhores parâmetros para ambos os modelos foi utilizado um GridSearchCV, e foi constatado que os melhores parametros para cada modelo foram os seguintes:

- **Árvore de Decisão:**
  - criterion: 'gini'
  - max_depth: 6
  - max_features: 'sqrt'
  - min_samples_leaf: 5
  - min_samples_split: 5
  - splitter: 'best'
- **KNN:**
  - metric: 'euclidean'
  - n_neighbors: 3
  - weights: 'uniform'
- **MLP:**
  - activation: 'tanh'
  - alpha: 0.001
  - hidden_layer_sizes: (100, 50)
  - learning_rate: 'constant'
  - solver: 'adam'

V. Treinamento do modelo e resultado

Após encontrar os melhores parâmetros para cada modelo, os modelos foram treinados e posteriormente foi feita uma avaliação dos resultados.

A. *Árvore de decisão*

Para a árvore de decisão foi realizada uma validação cruzada que mostrou um desempenho satisfatório com uma acurácia de: 0.96 ± 0.10.Posteriormente foram avaliados, também a precisão o recall e o F1-Score:

Classe	Precisão	Recall	F1-Score
1	0.67	0.83	0.74
2	0.67	0.44	0.53

TABLE II: Relatório de Classificação

Classe 1: O modelo obteve uma precisão de 67%, indicando que 67% das instâncias previstas como Classe 1 eram realmente dessa classe. O recall de 83% mostra que o modelo identificou corretamente 83% de todas as instâncias da Classe 1. O F1-score para esta classe foi de 0.74.

Classe 2: Para a Classe 2, o modelo apresentou uma precisão de 67%, o que significa que 67% das instâncias classificadas como Classe 2 eram realmente dessa classe. No entanto, o recall foi de 44%, indicando que o modelo identificou corretamente apenas 44% de todas as instâncias verdadeiras da Classe 2. O F1-score para esta classe foi de 0.53.

Os resultados mostram um desempenho satisfatório do modelo para a Classe 1, com boa precisão e recall, indicando que ele conseguiu identificar corretamente a maioria das instâncias dessa classe. No entanto, para a Classe 2, o recall mais baixo sugere que o modelo pode não estar identificando todas as instâncias dessa classe tão eficazmente quanto para a Classe 1.

B. *KNN*

Para o KNN foi realizada uma validação cruzada que mostrou um desempenho satisfatório com uma acurácia de: 0.98 ± 0.08.Posteriormente foram avaliados, também a precisão o recall e o F1-Score:

Classe	Precision	Recall	F1-score	Support
1	0.75	1.00	0.86	12
2	1.00	0.56	0.71	9

TABLE III: Resultados de Precisão, Recall e F1-score por classe

Classe 1: O modelo apresentou uma precisão de 75%, indicando que 75% das instâncias previstas como Classe 1 eram realmente dessa classe. O recall de 100% mostra que o modelo identificou corretamente todas as instâncias da Classe 1. O F1-score para esta classe foi de 0.86.

Classe 1: Para a Classe 2, o modelo alcançou uma precisão de 100%, o que significa que todas as instâncias classificadas como Classe 2 eram realmente dessa classe. No entanto, o recall foi de 56%, indicando que o modelo identificou corretamente apenas 56% de todas as instâncias verdadeiras da Classe 2. O F1-score para esta classe foi de 0.71.

Os resultados mostram um desempenho satisfatório do modelo para ambas as classes, com alta precisão. No entanto, o recall mais baixo para a Classe 2 sugere que o modelo pode não estar identificando todas as instâncias dessa classe tão eficazmente quanto para a Classe 1.

C. *MLP*

Para o MLP foi realizada uma validação cruzada que mostrou um desempenho satisfatório com uma acurácia de: 0.96 ± 0.10.Posteriormente foram avaliados, também a precisão o recall e o F1-Score:

Classe	Precision	Recall	F1-score	Support
1	0.80	1.00	0.89	12
2	1.00	0.67	0.80	9

TABLE IV: Resultados de Precisão, Recall e F1-score por classe

Classe 1: O modelo alcançou uma precisão de 80%, com recall de 100% e F1-score de 0.89, indicando excelente desempenho na identificação das instâncias da Classe 1.

Classe 2: O modelo obteve precisão de 100% para a Classe 2, com recall de 67% e F1-score de 0.80, mostrando um desempenho robusto, porém com recall inferior à Classe 1.

Os resultados indicam um desempenho geral muito bom do modelo, com altas precisões para ambas as classes. O recall perfeito para a Classe 1 evidencia que o modelo identificou todas as instâncias dessa classe. No entanto, o recall mais baixo para a Classe 2 sugere que algumas instâncias dessa classe não foram identificadas tão eficazmente quanto para a Classe 1, apesar da alta precisão.

VI. Conclusão

Após a análise dos resultados, os modelos demonstraram desempenho superior à linha de base estabelecida. Contudo, enfrentaram dificuldades na classificação da classe minoritária. Desse modo, os modelos foram testados sem alterações no dataset '14.csv', no qual apresenta duas classes uma majoritária e uma minoritária. Observou-se um recall inferior para a classe minoritária, indicando a necessidade de melhorias para aumentar tanto o recall da minoritária quanto a precisão das previsões em ambos os datasets.

References

[1] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2005. *Introduction to Data Mining* (First Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[2] Aurélien Géron. 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc.