Análise de Problemas de Probabilidade, Estudo de Banco de Dados e Implementação de Rede Neural para Previsão de Desempenho Acadêmico

Francisco Gabriel Alves Nunes
Universidade Federal do Ceará

Friday 30th August, 2024

Contents

1 Introdução

1.1 Problemática

Este trabalho consiste em 3 principais partes. Parte 1: Resolução de 200 questões em python sobre probabilidade e estatística onde poderíamos retirar essas questões tanto dos livros (Barbetta[1] e Morgado[3]) usados nas aulas, quanto de fontes externas referente ao assunto. Parte 2: Fazer a escolha de um banco de dados povoado e fazer a analise básica dos dados. Parte 3: Utilizar o banco de dados anteriormente selecionado para fazer o tratamento de dados e treinar uma rede neural.

1.2 Tecnologias utilizadas

Para Realização das atividades utilizaremos as seguintes tecnologias: Python na versão 3, Jupyter lab, dados salvos em arquivo CSV.

2 Resolução de problemas

A resolução dos problemas será tratada de maneira mais simplista possível, porém mantendo um padrão entre as perguntas e resposta para facilitar a compreensão. Em um arquivo "questoes.py" esta localizado a resolução das questões.

3 Análise de Banco de Dados

3.1 Escolha dos dados

A escolha dos dados no projeto foi feito na plataforma Kaggle[2] que consiste em um repositório de datasets de diversos tipos, um espaço para compartilhar notebooks interativos em Python e R, e uma comunidade onde os usuários podem aprender e trocar conhecimentos sobre ciência de dados e machine learning.

O Dataset selecionado foi sobre a performance de estudantes em exames de matemática, leitura e escrita. O dataset contem os seguintes dados:

Gender	race/ethnicity	Parental level of education	lunch	Test preparation course	math score	reading score	writing score

3.2 Tratamento de dados

Os dados foram analisados para que não tivesse falta de dados ou dados de inconsistentes. Onde os dados que estavam faltando foram atribuido a media dos que estava presente.

4 Aplicação do Algoritmo Supervisionado (K-Nearest Neighbors - KNN)

Para o aprendizado supervisionado, o algoritmo K-Nearest Neighbors (KNN) foi aplicado. O objetivo era prever a pontuação do exame (*exam_score*) com base nas características dos alunos.

4.1 Passos Executados

- **Divisão dos Dados:** Os dados foram divididos em conjuntos de treino e teste para avaliar o desempenho do modelo.
- **Treinamento do Modelo:** O modelo KNN foi treinado com os dados de treino, utilizando as características como entradas e a pontuação no exame como saída.
- **Avaliação:** O modelo foi avaliado utilizando o conjunto de teste, e a acurácia foi calculada para determinar a eficácia do modelo em prever as pontuações dos exames.

4.2 Resultados

O modelo KNN conseguiu prever as pontuações com uma acurácia razoável, indicando que as características fornecidas têm uma relação significativa com o desempenho dos alunos nos exames.

5 Aplicação do Algoritmo Não Supervisionado (K-Means)

Para o aprendizado não supervisionado, o algoritmo K-Means foi utilizado. O objetivo era agrupar os alunos em *clusters* com base nas características fornecidas, sem usar rótulos ou variáveis alvo.

5.1 Passos Executados

- **Escolha do Número de Clusters (K):** Um número arbitrário de *clusters* (neste caso, 3) foi escolhido para explorar como os alunos poderiam ser agrupados com base em suas características.
- **Treinamento do Modelo:** O modelo K-Means foi aplicado aos dados, que foram agrupados em 3 *clusters*.
- **Interpretação dos Resultados:** Os *clusters* formados foram analisados para entender as características dos grupos de alunos que o algoritmo identificou.

5.2 Resultados

O algoritmo K-Means agrupou os alunos em 3 *clusters* distintos, com base nas similaridades das suas características. Isso poderia ser útil para identificar diferentes perfis de alunos, que compartilham atributos comuns, como nível de educação dos pais ou tipo de refeição.

6 Conclusão

A aplicação dos dois tipos de aprendizado de máquina em um conjunto de dados permitiu explorar diferentes abordagens e objetivos:

- **Aprendizado Supervisionado (KNN):** Útil para prever resultados específicos com base em entradas rotuladas. No caso do conjunto de dados, o KNN foi usado para prever a pontuação de exames dos alunos.
- **Aprendizado Não Supervisionado (K-Means):** Focado em descobrir padrões e estruturas nos dados sem a necessidade de rótulos. Com o K-Means, foi possível identificar grupos de alunos que compartilham características semelhantes.

Cada abordagem tem suas vantagens e aplicações específicas, dependendo do objetivo final. Enquanto o aprendizado supervisionado é ideal para tarefas de previsão com dados rotulados, o aprendizado não supervisionado é valioso para análise exploratória e descoberta de padrões em dados não rotulados.

7 Implementação de Rede Neural

7.1 Tratamento de dados

Para a implementação da Rede Neural foi necessario fazer um novo tratamento de dados onde as notas individuais foram removidas para dar lugar a media das notas que sera a coluna de previsão da rede neural, ja os dados alfabeticos foram subistituidos por dados numericos para realizar o treinamento, seguindo a regra descrita:

1. Gender:**female**foi substituido por **0**, **male** foi substituido por **1**
2. Race/Ethnicity:
   (a) **Grupo A** tornasse **1**
   (b) **Grupo B** tornasse **2**
   (c) **Grupo C** tornasse **3**
   (d) **Grupo D** tornasse **4**
   (e) **Grupo E** tornasse **5**

3. arental level of education:
   (a) **high school** é subistituido por **1**
   (b) **some high school** é subistituido por **2**
   (c) **some college** é subistituido por **3**
   (d) **associate's degree** é subistituido por **4**
   (e) **bachelor's degree** é subistituido por **5**
   (f) **master's degree** é subistituido por **6**
4. lunch: **standard** foi trocado por **1**, **free/reduced** foi trocado por **0**
5. test preparation course: **none** foi trocado por **0**, **completed** foi trocado por **1**

7.2 Algoritmo Utilizado

Rede Neural Multicamadas (Multilayer Perceptron - MLP): Esse código implementa uma rede neural feedforward, também conhecida como Perceptron Multicamadas (MLP). É composta por várias camadas densas (fully connected layers), onde cada neurônio em uma camada está conectado a todos os neurônios da camada anterior. O modelo é otimizado usando o otimizador Adam, um algoritmo de otimização que combina as vantagens dos métodos AdaGrad e RMSProp, ajustando a taxa de aprendizado durante o treinamento. A função de perda utilizada é o Mean Squared Error (MSE), uma métrica comum para problemas de regressão, que mede o erro quadrático médio entre as previsões do modelo e os valores reais. A métrica adicional usada é o Mean Absolute Error (MAE), que mede o erro absoluto médio, oferecendo uma visão complementar ao MSE.

7.3 Divisão dos Dados em Treino e Teste

```
train_test_split(X, y, test_size=0.3, random_state=42)
```

Aqui, o conjunto de dados é dividido em duas partes: 70% dos dados são usados para treinamento (X_train, y_train) e 30% para teste (X_test, y_test). Treinamento: O modelo é treinado nos dados de treino. Durante esse processo, ele ajusta seus pesos para minimizar a função de perda. Teste: Após o treinamento, o modelo é avaliado nos dados de teste, que não foram usados durante o treinamento. Isso ajuda a avaliar a capacidade do modelo de generalizar para novos dados.

7.4 Normalização dos Dados

```
scaler = StandardScaler()
```

E

```
X_train = scaler.fit_transform(X_train)
```

A normalização (ou padronização) é uma etapa crucial, especialmente para redes neurais. O StandardScaler normaliza os dados para que cada feature tenha média 0 e desvio padrão 1. Isso ajuda a acelerar o treinamento e a evitar que features com valores maiores dominem a otimização.

7.5 Construção do Modelo

Arquitetura da Rede Neural: Camada de Entrada: input_shape=(X_train.shape[1],). define o número de features de entrada. Camadas Ocultas: A primeira camada oculta tem 128 neurônios com função de ativação ReLU. A segunda camada oculta tem 64 neurônios, também com ReLU. A terceira camada oculta tem 32 neurônios, novamente com ReLU. Camada de Saída: Possui 1 neurônio, sem função de ativação, já que estamos tratando de um problema de regressão, onde a saída pode ser qualquer valor contínuo.

7.6 Compilação do Modelo

```
optimizer = Adam(learning_rate=0.001)
```

O otimizador Adam é configurado com uma taxa de aprendizado de 0.001. Adam é eficiente e requer menos ajuste de hiperparâmetros. loss='mean_squared_error': A função de perda MSE é usada para calcular o erro quadrático médio, comum em regressão. metrics=['mean_absolute_error']: A métrica MAE é usada como uma métrica adicional para monitorar o desempenho do modelo.

7.7 Treinamento do Modelo

```
model.fit(X_train, y_train, epochs=100, validation_split=0.2)
```

O modelo é treinado por 100 épocas. Em cada época, o modelo passa por todos os dados de treino. validation_split =0.2: 20% dos dados de treino são usados como um conjunto de validação para monitorar o desempenho do modelo durante o treinamento e evitar overfitting.

7.8 Avaliação do Modelo

```
model.evaluate(X_test, y_test)
```

Após o treinamento, o modelo é avaliado no conjunto de teste. Isso fornece o valor final da função de perda (MSE) e da métrica (MAE) no conjunto de teste, indicando o quão bem o modelo está performando em dados não vistos.

8 Conclusão

Este relatório abordou três principais aspectos do projeto: resolução de problemas de probabilidade e estatística, análise de um banco de dados de desempenho acadêmico e implementação de uma rede neural para previsão de desempenho dos alunos.

Na primeira etapa, foram resolvidas questões de probabilidade e estatística usando Python, o que reforçou os conceitos teóricos abordados nas aulas e ampliou a habilidade de aplicar esses conceitos em situações práticas. Essa prática permitiu consolidar o conhecimento e testar diferentes abordagens para problemas comuns em estatística.

Na segunda parte, uma análise exploratória detalhada foi conduzida em um conjunto de dados de desempenho acadêmico de estudantes, extraído do Kaggle. O tratamento dos dados incluiu a manipulação de valores ausentes, que foram preenchidos com a média dos dados disponíveis, garantindo a integridade do conjunto de dados. A análise demonstrou como o pré-processamento de dados é essencial para garantir a qualidade dos resultados em qualquer projeto de ciência de dados.

Por fim, na terceira etapa, foi desenvolvida uma Rede Neural Multicamadas (MLP) para prever a média de desempenho dos alunos com base em diversas variáveis, como gênero, etnia, nível educacional dos pais, tipo de almoço e participação em cursos de preparação para testes. A aplicação de algoritmos de aprendizado de máquina supervisionado (KNN) e não supervisionado (K-Means) permitiu explorar diferentes abordagens para prever resultados e identificar padrões. A implementação da rede neural demonstrou um desempenho satisfatório, evidenciando que a escolha correta de métodos e o tratamento adequado dos dados são cruciais para alcançar previsões precisas.

No geral, o trabalho integrou conceitos teóricos e práticos na solução de problemas complexos, demonstrando a importância de uma abordagem multidisciplinar na análise de dados e na aplicação de técnicas de machine learning. A comparação entre métodos supervisionados e não supervisionados também destacou a importância de escolher a técnica certa para cada problema específico, mostrando que tanto a previsão de resultados quanto a análise exploratória de dados têm seu valor em diferentes contextos.

Este projeto poderia ser ampliado no futuro com a implementação de técnicas adicionais de aprendizado profundo, como redes neurais convolucionais (CNN) ou redes recorrentes (RNN), dependendo da natureza dos dados e do objetivo do estudo. Além disso, uma análise mais aprofundada de hiperparâmetros e a aplicação de técnicas de validação cruzada poderiam melhorar ainda mais a acurácia e robustez dos modelos.

References

[1] P. A. Barbetta. *Estatística Para Cursos De Engenharia E Informática*. Atlas, 2010.

[2] Kaggle. *Students Performance in Exams Dataset*. Acessado em: 10 de agosto de 2024. 2024. URL: `https://www.kaggle.com/datasets/spscientist/students-performance-in-exams/data`.

[3] M. Morgado. *Análise Combinatória e Probabilidade*. SBM, 2016.