INSTITUTO INFNET

Faculdade de Engenharia e Desenvolvimento de Software

Dados para Machine Learning: Feature Engineering

[25E2_2]

TP1

Aluno: JEAN MICHAEL ESTEVEZ ALVAREZ

E-mail: jean.alvarez@al.infnet.edu.br

Matrícula:

Professor: Diego da Silva Rodrigues

Distrito Federal

Abril, 2025

SUMÁRIO

1 EXERCÍCIOS

1. **Escreva um parágrafo explicando o que é Machine Learning e por que é útil em diversas aplicações do mundo real.**

R:Machine Learning é um ramo da inteligência artificial que desenvolve algoritmos capazes de aprender padrões a partir de dados, ajustando-se automaticamente sem necessidade de programação explícita para cada tarefa.

Essa abordagem é útil em diversas aplicações do mundo real, como diagnósticos médicos, sistemas de recomendação, previsão de demanda e detecção de fraudes, pois permite que sistemas adaptem-se a novos dados e tomem decisões com base em evidências empíricas, melhorando continuamente seu desempenho.

2. **Liste e explique brevemente os dois principais tipos de Machine Learning: Aprendizado Supervisionado e Não-Supervisionado.**

R: Tipos de Machine Learning

1. Aprendizado Supervisionado

Nesse tipo, o modelo é treinado com um conjunto de dados rotulado, ou seja, os dados de entrada já vêm com as respostas corretas (rótulos). O objetivo é aprender a relação entre entrada e saída para prever ou classificar novos dados. Exemplos: regressão linear, árvores de decisão e redes neurais para classificação.

2. Aprendizado Não-Supervisionado

Aqui, o modelo trabalha com dados sem rótulos. Ele busca identificar padrões ou estruturas ocultas nos dados, como agrupamentos ou associações. É útil para explorar dados, segmentar clientes ou reduzir dimensionalidade. Exemplos: k-means, análise de componentes principais (PCA) e algoritmos de agrupamento hierárquico.

3. **Pesquise e liste cinco bibliotecas de Python 3 que são comumente usadas em Machine Learning, destacando a importância do Scikit-Learn.**

R: Bibliotecas Python Comuns em Machine Learning

1. Scikit-Learn

Uma das bibliotecas mais populares para Machine Learning em Python. Oferece ferramentas simples e eficientes para análise e modelagem de dados, como classificação, regressão, clustering e redução de dimensionalidade. Sua API intuitiva e ampla documentação a tornam ideal para iniciantes e projetos profissionais.

2. TensorFlow

Desenvolvida pelo Google, é usada principalmente para criar e treinar redes neurais profundas. Suporta computação distribuída e é altamente escalável, sendo amplamente utilizada em produção.

3. Keras

Uma interface de alto nível para redes neurais, que roda sobre o TensorFlow. Permite a criação rápida e fácil de modelos de deep learning com poucas linhas de código.

4. PyTorch

Desenvolvida pelo Facebook, é uma biblioteca poderosa para deep learning que se destaca pela flexibilidade e por seu modo dinâmico de construção de redes neurais, facilitando a depuração e o desenvolvimento experimental.

5. Pandas

Embora não seja exclusivamente para Machine Learning, é fundamental na etapa de pré-processamento, permitindo manipulação eficiente de dados estruturados com DataFrames, filtragens, agrupamentos e transformações

4. Explique a diferença entre aprendizado baseado em instâncias e aprendizado baseado em modelos, dando um exemplo de cada um.

R: Diferença entre Aprendizado Baseado em Instâncias e em Modelo

- Aprendizado Baseado em Instâncias

Nesse tipo, o algoritmo memoriza os dados de treinamento e toma decisões com base na similaridade entre novas entradas e exemplos anteriores. Ele não cria um modelo generalizado, mas depende diretamente dos dados armazenados.

Exemplo: o algoritmo K-Nearest Neighbors (KNN), que classifica uma nova amostra com base nas classes dos vizinhos mais próximos.

- Aprendizado Baseado em Modelos

Aqui, o algoritmo constrói um modelo abstrato a partir dos dados de treinamento, aprendendo uma função que mapeia entradas para saídas. Esse modelo pode então ser usado para prever novos casos.

Exemplo: a Regressão Linear, que ajusta uma equação matemática para prever valores contínuos com base em variáveis de entrada.

5. **Enumere e descreva três desafios comuns enfrentados ao criar modelos de Machine Learning.**

R: Três Desafios Comuns em Modelos de Machine Learning

1. Overfitting (Sobreajuste)

Ocorre quando o modelo aprende tão bem os dados de treinamento que perde a capacidade de generalizar para novos dados. Ele memoriza ruídos e padrões específicos, resultando em baixa performance em testes.

2. Dados Insuficientes ou Desequilibrados

Modelos de ML precisam de grandes quantidades de dados representativos. Quando os dados são escassos ou há classes com pouca representatividade (desequilíbrio), o desempenho do modelo pode ser comprometido.

3. Escolha e Engenharia de Features

A seleção inadequada de variáveis (features) pode limitar a capacidade do modelo de aprender padrões relevantes. A engenharia de features — processo de criar e transformar variáveis — é crucial para melhorar a acurácia do modelo.

6. **Acesse a base de dados "Pinguins" disponível em PalmerPenguins GitHub. Identifique as 'features' e o 'target' na base de dados.**

https://github.com/EstevezCodando/ia_ml_model/

target: **species**

features: **island, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g, sex, year**

7. **Com a base de dados "Pinguins", escreva um código em Python para separar os dados em conjuntos de treino e validação. Observe que iremos criar um classificador que diferencia a espécie "Adelie" das outras duas.**

https://github.com/EstevezCodando/ia_ml_model/

```
Criar target binário: 1 para Adelie, 0 para as outras espécies
penguins['target'] = (penguins['species'] == 'Adelie').astype(int)
# Selecionar features numéricas
features = ['bill_length_mm', 'bill_depth_mm', 'flipper_length_mm', 'body_mass_g']
X = penguins[features]
y = penguins['target']
# Separar conjuntos de treino e validação (80% treino, 20% validação)
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.25, random_state=42,
stratify=y)
```

8. **Utilizando a biblioteca Scikit-Learn, escreva um código em Python para construir um modelo de Machine Learning usando o algoritmo de K-Nearest Neighbors com a base de dados "Pinguins".**

   https://github.com/EstevezCodando/ia_ml_model/

9. **Após treinar o modelo de K-Nearest Neighbors com a base de dados "Pinguins", escreva um código para avaliar a acurácia do seu modelo nos dados de validação.**

   https://github.com/EstevezCodando/ia_ml_model/

```
# Treinar o modelo com os dados de treino
knn.fit(X_train_scaled, y_train)
# Fazer previsões com os dados de validação
y_pred = knn.predict(X_val_scaled)
# Avaliar a acurácia do modelo (percentual de acertos no conjunto de validação)
print("Acurácia do modelo:", accuracy_score(y_val, y_pred))
```

10. **Baseado nos exercícios anteriores, escreva um breve texto discutindo os desafios encontrados ao criar um modelo de Machine Learning e como cada etapa do processo é crucial para o sucesso do projeto.**

Durante a construção de um modelo de Machine Learning com a base Palmer Penguins, A primeira dificuldade foi lidar com dados ausentes, que, se não tratados, podem comprometer totalmente o desempenho do modelo.

A etapa de seleção de features também se mostrou crítica: variáveis irrelevantes ou colineares podem confundir o algoritmo, enquanto boas combinações (como features derivadas) aumentam significativamente a capacidade preditiva.

A padronização dos dados foi essencial, especialmente para o K-Nearest Neighbors, pois esse algoritmo é sensível à escala das variáveis. Sem isso, variáveis com magnitudes maiores dominariam a métrica de distância. Outro ponto relevante foi a escolha de métricas de avaliação adequadas, como a matriz de confusão, que forneceu uma visão detalhada dos tipos de erro, além da simples acurácia.

Por fim, técnicas como PCA e t-SNE ajudaram na visualização da separabilidade entre as classes, auxiliando na interpretação do modelo, embora não contribuam diretamente para

sua melhoria. Assim, cada etapa — desde o pré-processamento até a avaliação — é interdependente e indispensável para garantir a robustez e a interpretabilidade do modelo.

11. Baseado nos exercícios anteriores, crie um novo modelo de classificação binário, agora para classificar os pinguins da classe Gentoo versus as outras espécies. O desempenho deste modelo é pior ou melhor do que o primeiro modelo criado?

O modelo Gentoo vs Outras geralmente tem desempenho melhor, pois Gentoo apresenta características morfológicas (como massa corporal e nadadeira) bem distintas das outras espécies, o que facilita a separação pelo KNN.

12. Seria possível criar um modelo capaz de diferenciar as três espécies de pinguim? Escreva sua resposta destacando quais as diferenças seriam observadas na análise?

Sim, com base na matriz de confusão e na visualização por PCA, observam-se as seguintes diferenças na análise multiclasse:

- Gentoo é a espécie mais facilmente distinguível: aparece como um cluster isolado no gráfico de PCA e foi classificada com 100% de acerto (29/29) pelo modelo KNN.
- Adelie também apresenta excelente separação, com 40 acertos e nenhum erro.
- Chinstrap é a espécie com maior sobreposição com Adelie, tanto no gráfico quanto na matriz, onde houve 1 erro de classificação como Adelie — reflexo da semelhança morfológica entre essas duas espécies.

Essas diferenças destacam a importância da distribuição das features no espaço de decisão, que favorece a separação de Gentoo, mas impõe desafios para diferenciar Adelie e Chinstrap.

2 LINK GITHUB

https://github.com/EstevezCodando/ia_ml_model/blob/master/TP1_PalmerPenguins.ipynb