TP1

August 10, 2024

1 TP1 - Inteligência Artificial: Model LifeCycle [24E3_1] - Alberto F. Pluecker

1.1 1: Escreva um parágrafo explicando o que é Machine Learning e por que é útil em diversas aplicações do mundo real.

Machine Learning (ML) é uma subárea da inteligência artificial que envolve o desenvolvimento de algoritmos que permitem que computadores aprendam a partir de dados, reconhecendo padrões e tomando decisões com o mínimo de intervenção humana. A utilidade do ML está em sua capacidade de lidar com grandes volumes de dados, detectar tendências e realizar previsões, o que é crucial em diversas aplicações do mundo real, como recomendações de produtos, diagnósticos médicos, análise de sentimentos em mídias sociais, e muito mais.

1.2 2: Liste e explique brevemente os dois principais tipos de Machine Learning: Aprendizado Supervisionado e Não-Supervisionado.

- Aprendizado Supervisionado: Envolve o uso de um conjunto de dados rotulado para treinar o modelo. Cada exemplo no conjunto de treinamento inclui a entrada e a saída desejada, de forma que o algoritmo aprenda a mapear entradas para saídas. Exemplos incluem regressão linear e árvores de decisão.
- Aprendizado Não-Supervisionado: Neste caso, o modelo é treinado usando dados que não são rotulados, e o objetivo é inferir a estrutura subjacente dos dados. Algoritmos não supervisionados tentam encontrar padrões nos dados. Exemplos incluem clustering (agrupamento) e redução de dimensionalidade.

1.3 3: Pesquise e liste cinco bibliotecas de Python 3 que são comumente usadas em Machine Learning, destacando a importância do Scikit-Learn.

- Scikit-Learn: Fornece uma variedade de algoritmos para classificação, regressão, clustering e redução de dimensionalidade, além de ferramentas para avaliação e pré-processamento de dados.
- TensorFlow: Usado para construir e treinar redes neurais, muito popular em Deep Learning.
- PyTorch: Outra biblioteca útil para Deep Learning, oferecendo flexibilidade e suporte para aprendizado dinâmico de gráficos.
- Pandas: Frequentemente usada para manipulação e análise de dados, prática essencial para ML.
- Keras: Uma API de alto nível para redes neurais que pode ser executada em cima do TensorFlow e facilita a criação de modelos complexos.

1.4 4: Explique a diferença entre aprendizado baseado em instâncias e aprendizado baseado em modelos, dando um exemplo de cada um.

- Aprendizado Baseado em Instâncias: Este tipo de aprendizado memoriza exemplos do conjunto de treinamento e faz previsões comparando novos exemplos diretamente com esses exemplos memorizados. Um exemplo é o algoritmo K-Nearest Neighbors (K-NN), que classifica um ponto com base nos rótulos dos pontos de dados mais próximos.
- Aprendizado Baseado em Modelos: Aqui, um modelo geral é criado a partir do conjunto de treinamento. O modelo faz previsões com base nas características inferidas a partir dos dados. Um exemplo é a regressão linear, que tenta ajustar uma linha através dos dados para prever a saída.

1.5 5: Enumere e descreva três desafios comuns enfrentados ao criar modelos de Machine Learning.

- Qualidade dos Dados: Dados incompletos, imprecisos ou enviesados podem afetar negativamente o desempenho do modelo.
- Overfitting e Underfitting: Overfitting ocorre quando o modelo aprende os detalhes e ruídos dos dados de treinamento, prejudicando seu desempenho em novos dados. Underfitting acontece quando o modelo é muito simples para capturar a relação subjacente entre os dados de entrada e saída.
- Escalabilidade: À medida que a quantidade de dados cresce, o modelo deve ser capaz de processá-los de forma eficiente sem comprometer o desempenho.

1.6 6: Acesse a base de dados "Pinguins" disponível em PalmerPenguins GitHub. Identifique as 'features' e o 'target' na base de dados.

[14]:
```
import pandas as pd
from sklearn.model_selection import train_test_split

url = "https://raw.githubusercontent.com/allisonhorst/palmerpenguins/master/
 ↪inst/extdata/penguins.csv"
data = pd.read_csv(url).dropna()

data.head()
```

[14]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	2007
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	2007
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female	2007
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female	2007
5	Adelie	Torgersen	39.3	20.6	190.0			

```
5       3650.0    male  2007
```

- Features: `bill_length_mm`, `bill_depth_mm` , `flipper_length_mm`, `body_mass_g`, `sex`
- Target: `species` (Adelie, Gentoo, Chinstrap).

1.7 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.

[15]:
```
features = data[['bill_length_mm', 'bill_depth_mm', 'flipper_length_mm',␣
 ↪'body_mass_g']]
target = data['species'] == 'Adelie'

features_train, features_val, target_train, target_val =␣
 ↪train_test_split(features, target, test_size=0.2, random_state=42)
```

1.8 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".

[9]:
```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(features_train, target_train)
```

[9]:
```
KNeighborsClassifier(n_neighbors=3)
```

1.9 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.

[10]:
```
from sklearn.metrics import accuracy_score

target_pred = knn.predict(features_val)
accuracy = accuracy_score(target_val, target_pred)
print(f'Acurácia do modelo: {accuracy:.2f}')
```

```
Acurácia do modelo: 0.73
```

1.10 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.

Os desafios incluem a escolha e preparação dos dados corretos, a seleção do(s) algoritmo(s) apropriado(s), e o ajuste fino do modelo para evitar overfitting ou underfitting. Cada etapa do processo, desde a limpeza de dados até a avaliação do modelo, contribui para a precisão e generalização do modelo. As etapas de pré-processamento são especialmente importantes, pois afetam diretamente

a qualidade das previsões. A escolha do modelo também é importante, pois diferentes algoritmos têm diferentes são mais ou menos adequados dependendo da natureza dos dados.

1.11 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?

[13]:
```
target = data['species'] == 'Gentoo'

features_train, features_val, target_train, target_val =␣
 ↪train_test_split(features, target, test_size=0.2, random_state=42)

knn.fit(features_train, target_train)

target_pred = knn.predict(features_val)
accuracy = accuracy_score(target_val, target_pred)
print(f'Acurácia do modelo Gentoo vs outras: {accuracy:.2f}')
```

```
Acurácia do modelo Gentoo vs outras: 0.94
```

Comparando os desempenhos, podemos observar que o segundo modelo é “melhor” do que o primeiro, dado a sua maior acurácia.

1.12 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, é possível. Isso envolveria um problema de classificação multiclasses, onde o alvo possui três rótulos (Adelie, Gentoo, Chinstrap). A principal diferença estaria nos desafios adicionais de garantir que o modelo seja capaz de capturar as características únicas de cada espécie sem confundi-las. Isso pode envolver técnicas como one-vs-rest ou one-vs-one, além de uma avaliação com métricas específicas para problemas multiclasses.