FUNDAMENTOS DE BIG DATA E DATA ANALYTICS COM PYTHON

OBJETIVO DA AULA: Introdução ao Machine Learning com Scikit-Learn

Exercício 01

Problema de Negócio: Usando dados históricos é possível prever o salário de uma pessoa com base no tempo de estudo em horas por mês?

1. **Importar as bibliotecas necessárias:**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

2. Carregar os dados do arquivo horasDeEstudos.csv;
3. **Visualizar algumas informações básicas do dataframe:**
    a. Cabeçalho;
    b. Final do dataframe;
    c. Tipo de dados;
4. **Gerar uma análise exploratória dos dados:**
    a. Verificar se há dados nulos e mostrar a quantidade;
    b. Resumo estatístico do dataframe;
    c. Estudo estatístico das variáveis preditoras (X);
    d. Analisar a correlação entre as variáveis;

e. Crie os gráficos de dispersão, jointplot e pairplot para demonstrar a correlação entre as variáveis;
f. Analisar a distribuição das informações da variável preditora (X) com um histograma;

5. **Preparação dos dados:**
   a. Preparar a variável de entrada (X), convertendo-a em um array. Utilize a função np.array.
   b. Verifique o tipo de informação, utilizando a função type;
   c. Ajuste o formato da matrix, transformando cada dado em uma matriz 1x1. Utilize o comando X.reshape(-1,1);
   d. Crie uma variável chamada y e atribua a ela a coluna salário do dataframe.
6. **Separação dos dados em treino e teste:**
   a. Separe os dados em treino e teste, utilizando o comando train_teste_split. Adote como tamanho da amostra de treino como sendo 20% e Random_state igual a 42;
   b. Analise o tamanho das variáveis de treino e teste.
7. **Treinamento do modelo:**
   a. Crie o modelo de regressão linear simples. *modelo = LinearRegression ()*;
   b. Treino o modelo com as variáveis X_train e y_train. *modelo.fit(X_train, y_train)*;
   c. Visualize a reta de regressão linear (previsões) e os dados reais utilizados no treinamento. Utilize os comandos abaixo:

```
plt.scatter(X, y, color = "blue", label = "Dados Reais Históricos")
plt.plot(X, modelo.predict(X), color = "red", label = "Reta de
Regressão com as Previsões do Modelo")
plt.xlabel("Horas de Estudo")
plt.ylabel("Salário")
plt.legend()
plt.show()
```

d. Avaliar o modelo nos dados de teste, calculando o coeficiente R^2. Utilize o comando *modelo.score(X_teste, y_teste)*.

e. Calcule os coeficientes a e b da equação. Utilize os comandos modelo.intercept_ e modelo.coef_.

8. **Deploy do modelo:**

a. Faça três testes com os valores a seguir: 48, 65 e 73.

b. Crie um código para solicitar ao usuário a inserção da quantidade horas estudadas e retorne a previsão do salário.

Exercício 02 – Previsão de Vendas

Construa um modelo que preveja vendas com base no dinheiro gasto em diferentes plataformas de marketing.

Use o conjunto de dados de publicidade fornecido no ISLR e analise a relação entre 'publicidade na TV' e 'vendas' usando um modelo de regressão linear simples.

Neste caderno, construiremos um modelo de regressão linear para prever vendas usando uma variável preditora apropriada.

1. **Importe as bibliotecas:**
    a. Pandas
    b. Numpy
    c. Matplotlib
    d. Seaborn
    e. OS
    f. Scikit Learn
2. **Carregue os dados do arquivo advertising.csv para um dataframe.**
3. **Visualizar algumas informações básicas do dataframe:**
    a. Cabeçalho;
    b. Final do dataframe;
    c. Tipo de dados;
4. **Limpeza e análise exploratória dos dados:**
    a. Verificar se há dados nulos e mostrar a quantidade;
    b. Resumo estatístico do dataframe;
    c. Analise se há outliers no conjunto de dados. Crie o seguinte comando para analisarmos os dados:

```
# Outlier Analysis
fig, axs = plt.subplots(3, figsize = (5,5))
plt1 = sns.boxplot(advertising['TV'], ax = axs[0])
plt2 = sns.boxplot(advertising['Newspaper'], ax = axs[1])
plt3 = sns.boxplot(advertising['Radio'], ax = axs[2])
plt.tight_layout()
```

d. Análise univariada da coluna sales. Utilize o comando plt.boxplot para exibir o comportamento dos dados.

e. Verifique como as variáveis se relacionam, utilizando o gráfico de dispersão.

```
# Let's see how Sales are related with other variables using scatter plot.
sns.pairplot(advertising, x_vars=['TV', 'Newspaper', 'Radio'], y_vars='Sales', height=4, aspect=
1, kind='scatter')
plt.show()
```

f. Verifique a correlação existente entre as variáveis, utilizando o gráfico mapa de calor.

```
# Let's see the correlation between different variables.
sns.heatmap(advertising.corr(), cmap="YlGnBu", annot = True)
plt.show()
```

5. **Separação dos dados em treino e teste:**

   a. Selecione a variável que têm maior correlação com as vendas e atribua ela à variável X.

   b. Separe os dados em treino e teste, utilizando o comando train_teste_split. Adote como tamanho da amostra de treino como sendo 30% e Random_state igual a 0;

   c. Analise o tamanho das variáveis de treino e teste.

6. **Treinamento do modelo e Deploy:**

   a. Crie o modelo de regressão linear simples. *modelo = LinearRegression ()*;

b. Treino o modelo com as variáveis X_train e y_train. *modelo.fit(X_train, y_train)*;
c. Visualize a reta de regressão linear (previsões) e os dados reais utilizados no treinamento.
d. Faça três testes com os valores a seguir: 48, 65 e 73.
e. Crie um código para solicitar ao usuário a inserção da quantidade horas estudadas e retorne a previsão do salário.