### **Task 9**

**Machine Learning**

Upload .py or Ipynb extension file on GitHub public repo “100DaysofBytewise" and share the link in the submission form by 4 July 2024.

**Exercise: Load a dataset (e.g., the Boston Housing dataset from Scikit-Learn) and prepare the data for linear regression (e.g., split into training and testing sets).**

**Code:** from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split

import pandas as pd

boston = fetch\_california\_housing()

df = pd.DataFrame(boston.data, columns=boston.feature\_names)

df['MEDV'] = boston.target

print(df.head())

X = df.drop('MEDV', axis=1)

y = df['MEDV']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

print(f"Training features shape: {X\_train.shape}")

print(f"Testing features shape: {X\_test.shape}")

print(f"Training target shape: {y\_train.shape}")

print(f"Testing target shape: {y\_test.shape}")

**Exercise: Implement linear regression using Scikit-Learn. Fit the model to the training data.**

**Code:** import pandas as pd

from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

california = fetch\_california\_housing()

df = pd.DataFrame(california.data, columns=california.feature\_names)

df['MedHouseVal'] = california.target

X = df.drop('MedHouseVal', axis=1)

y = df['MedHouseVal']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

print("Coefficients:", model.coef\_)

print("Intercept:", model.intercept\_)

**Exercise: Predict the target variable for the test set using the fitted linear regression model.**

**Code:** import pandas as pd

from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

california = fetch\_california\_housing()

df = pd.DataFrame(california.data, columns=california.feature\_names)

df['MedHouseVal'] = california.target

X = df.drop('MedHouseVal', axis=1)

y = df['MedHouseVal']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

# Print the first few predictions

print("First few predictions:", y\_pred[:5])

print("First few actual values:", y\_test.values[:5])

**Exercise: Calculate the Mean Squared Error (MSE) of the linear regression model on the test set.**

**Code:** import pandas as pd

from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error

from sklearn.linear\_model import LinearRegression

california = fetch\_california\_housing()

df = pd.DataFrame(california.data, columns=california.feature\_names)

df['MedHouseVal'] = california.target

X = df.drop('MedHouseVal', axis=1)

y = df['MedHouseVal']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse= mean\_squared\_error(y\_pred,y\_test)

print("Mean Squared Error:", mse)

**Exercise: Calculate the R-squared value of the linear regression model on the test set.**

**Code:** import pandas as pd

from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import r2\_score

from sklearn.linear\_model import LinearRegression

california = fetch\_california\_housing()

df = pd.DataFrame(california.data, columns=california.feature\_names)

df['MedHouseVal'] = california.target

X = df.drop('MedHouseVal', axis=1)

y = df['MedHouseVal']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

r=r2\_score(y\_pred,y\_test)

print("R-Squared Error :", r)

**Exercise: Plot the regression line along with the actual data points to visually assess the model's performance.**

**Code:** import pandas as pd

import matplotlib.pyplot as plt

from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import r2\_score

from sklearn.linear\_model import LinearRegression

california = fetch\_california\_housing()

df = pd.DataFrame(california.data, columns=california.feature\_names)

df['MedHouseVal'] = california.target

X = df.drop('MedHouseVal', axis=1)

y = df['MedHouseVal']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

plt.figure(figsize=(6,6))

plt.plot([y\_test.min(),y\_test.max()],[y\_test.min(),y\_test.max()],'--b')

plt.grid(True)

plt.show()

**Exercise: Evaluate the model's performance by comparing the predicted values with the actual values. Create a scatter plot of the predicted vs. actual values.**

**Code:** import pandas as pd

import matplotlib.pyplot as plt

from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import r2\_score

from sklearn.linear\_model import LinearRegression

california = fetch\_california\_housing()

df = pd.DataFrame(california.data, columns=california.feature\_names)

df['MedHouseVal'] = california.target

X = df.drop('MedHouseVal', axis=1)

y = df['MedHouseVal']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

plt.figure(figsize=(6,6))

plt.scatter(y\_test,y\_pred,alpha=0.46)

plt.plot([y\_test.min(),y\_test.max()],[y\_test.min(),y\_test.max()],'--b')

plt.grid(True)

plt.show()

**Exercise: Interpret the coefficients of the linear regression model. Explain the impact of each feature on the target variable.**

**Answer:** c=pd.DataFrame(model.coef\_,X.columns,columns=['coefficients'])

print(c)