Machine Learning

1- SIMPLE REGRESSION

```
pip install scikit-learn
```

```
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
    Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.22.4)
    Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.10.1)
    Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.2.0)
    Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.1.0)
```

STEP-1 IMPORT LIBRARY

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
```

STEP-2 IMPORT DATA

```
df = pd.read_csv('salary_data.csv')
df.head()
```

	YearsExperience	Salary
0	1.1	39343
3	2.0	43525
4	2.2	39891

Cut 1 cells. You can now paste them in this or a different notebook. ✕

STEP-3 SELECTING INPUT AND OUTPUT VARIABLES

```
X = df[["YearsExperience"]]
y = df["Salary"]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=0)
```

STEP-4 MAKING LINEAR REGRESSION MODEL

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
```

STEP-5 FITTING THE MODEL

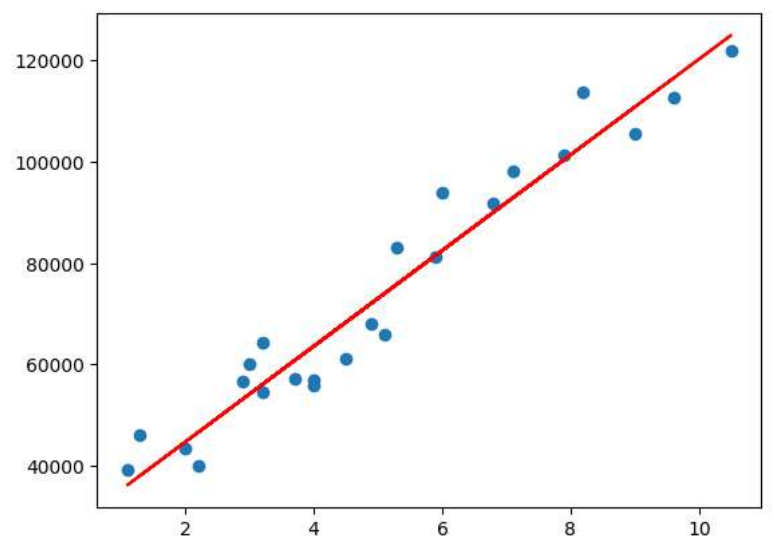
```
model = model.fit(X,y)
model
```

```
▾ LinearRegression
LinearRegression()
```

STEP-6 Plotting

```
plt.scatter(X_train,y_train)
plt.plot(X_train.values, model.predict(X_train), color="red")
```

```
[<matplotlib.lines.Line2D at 0x7f18d324ed70>]
```



STEP-6 Predicting the Model

```
model.predict([[50]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegressior
  warnings.warn(
array([498290.31627142])
```

Cut 1 cells. You can now paste them in this or a different notebook. ✕

```
from sklearn.model_selection import train_test_split
X_train , y_train , X_test , y_test = train_test_split(X,y,test_size=0.2, random_state=0)
```

TO CHECK THE ACCURACY SCORE AND SPLIT DATA IN 80/20 RATIO

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LinearRegression

# Assuming X and y are your feature matrix and target vector respectively
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Create and fit the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Compute the accuracy score (Note: Linear Regression is not typically used for classification tasks, so accuracy score might not be app
accuracy = model.score(X_test, y_test)

print("Accuracy score: {:.2f}".format(accuracy))
```

```
Accuracy score: 0.99
```

Splitting the average of the model

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LinearRegression

# Assuming X and y are your feature matrix and target vector respectively
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Create and fit the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)
accuracy = model.score(X_test, y_test)

print("Accuracy score: {:.2f}".format(accuracy))
```

```
Accuracy score: 0.90
```

✓ 0s completed at 6:02 AM

Cut 1 cells. You can now paste them in this or a different notebook.