MACHINE LEARNING

Step-1 Import Libraries

```
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
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

Step 3 Spliting dataset into training and testing data

```
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 Fit Linear Regression Model

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model = model.fit(X_train, y_train)
model
```

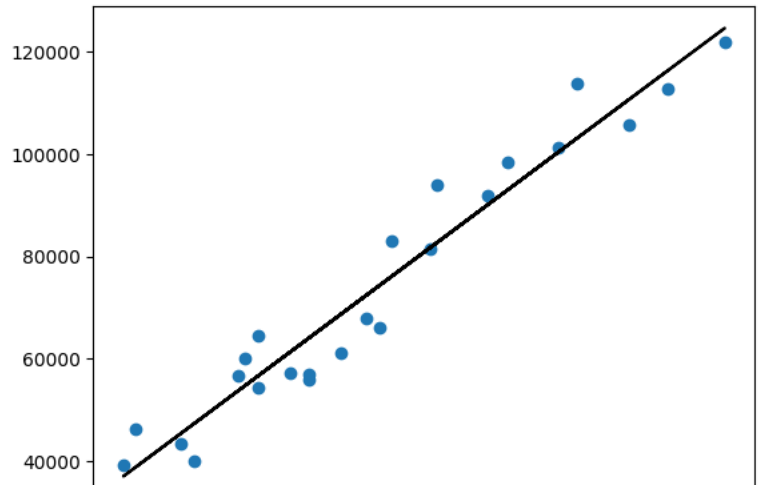
```
▾ LinearRegression
LinearRegression()
```

5 Plotting

```
import matplotlib.pyplot as plt
plt.scatter(X_train,y_train)
plt.plot(X_train.values, model.predict(X_train), color="black")

```

```
[<matplotlib.lines.Line2D at 0x7f066367e770>]
```



6 Evaluating Model Fitness

```
print("Score for training data =" ,model.score(X_train, y_train))
print("Score for test data =" ,model.score(X_test, y_test))
```

```
Score for training data = 0.9411949620562126
Score for test data = 0.988169515729126
```

7 Prediction of unknown values

```
model.predict([[20],[25],[30]])

```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was
  warnings.warn(
array([213031.60168521, 259594.47731886, 306157.3529525 ])
```

8 Measure the Accuracy

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
# Generate sample data
np.random.seed(42)
X = np.random.rand(100, 5) # Features (100 samples, 5 features)
y = np.random.rand(100) # Target values
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=100)
# Train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
# Make predictions on the test set
y_pred = model.predict(X_test)
# Calculate the R-squared score as a measure of accuracy
accuracy = r2_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.0035824605319503267
```