1 Machine Learning

1.1 Simple Linear Regression

Step 0 Import Libraries

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

Step-1 Import Dataset

```
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-2 Splitting 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-3 Fit Linear Regression Model

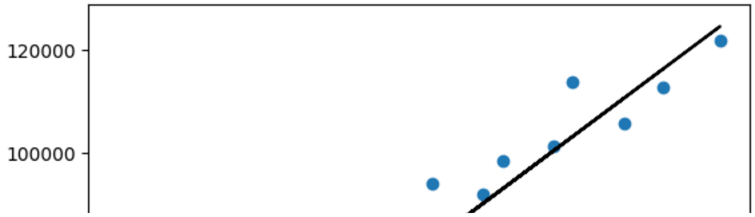
```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model=model.fit(X_train, y_train)
model
```

```
▾ LinearRegression
LinearRegression()
```

Step-4 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 0x7f04c86b4d90>]
```



Step-5 Evaluating Model Fitness

```
#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
```

Step-6 Prediction of unknown values

```
model.predict([[10], [15], [20]])
```

```
     /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([119905.85041792, 166468.72605157, 213031.60168521])
```

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Assuming you have your data stored in X and y arrays

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Training the linear regression model
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting on the testing set
y_pred = regressor.predict(X_test)

# Calculating the mean squared error (MSE)
mse = mean_squared_error(y_test, y_pred)

# Calculating the accuracy (R-squared score)
accuracy = regressor.score(X_test, y_test)

print("Mean Squared Error: ", mse)
print("Accuracy (R-squared): ", accuracy)
```

```
     Mean Squared Error:  49830096.85590839
     Accuracy (R-squared):  0.9024461774180497
```