1) Delivery\_time -> Predict delivery time using sorting time

2) Salary\_hike -> Build a prediction model for Salary\_hike

------------------------------------------------------------

Build a simple linear regression model by performing EDA and do necessary transformations and select the best model using R or Python.

==================================================================================

1)Q

**Ans: delivery\_time\_csv (file)**

**# Import necessary librarie**

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

**# Load the dataset**

file\_path = ('F:\delivery\_time.csv') # Update with your actual file path

data = pd.read\_csv(file\_path)

# Display the first few rows of the dataset

print(data.head())

**# Visualize the data if applicable**

**# For example, if you have columns 'Sorting Time' and 'Delivery Time'**

plt.scatter(data['Sorting Time'], data['Delivery Time'])

plt.xlabel('Sorting Time')

plt.ylabel('Delivery Time')

plt.title('Sorting Time vs Delivery Time')

plt.show()

**# Prepare the data for modeling**

X = data[['Sorting Time']]

y = data['Delivery Time']

**# Split 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)

**# Train a linear regression model**

model = LinearRegression()

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

# Make predictions on the test set

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

**# Evaluate the model**

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

print(f'Mean Squared Error: {mse}')

**# Plot the regression line**

plt.scatter(X\_test, y\_test, label='Actual')

plt.plot(X\_test, y\_pred, color='red', linewidth=2, label='Predicted')

plt.xlabel('Sorting Time')

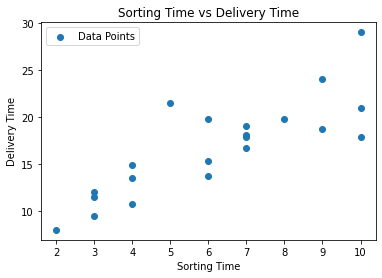
plt.ylabel('Delivery Time')

plt.title('Linear Regression Model')

plt.legend()

plt.show()

**OUTPUT:**

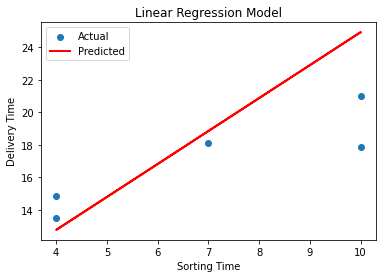
**1)** 

**2) Intercept: [4.68229796]**

**Coefficient: [2.02424455]**

**3)** **Mean Squared Error: 14.046738956635016**

**4)**



**-------------------------------**

**2) Q.**

**Ans: Salary\_data.csv(file).**

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

**# Load the dataset**

file\_path = r'F:\Salary\_data.csv'

data = pd.read\_csv(file\_path)

# Display the first few rows of the dataset

print(data.head())

**# EDA and Data Visualization**

plt.scatter(data['YearsExperience'], data['Salary'])

plt.xlabel('Years of Experience')

plt.ylabel('Salary')

plt.title('Scatter Plot of Years of Experience vs. Salary')

plt.show()

# Split the data into training and testing sets

X = data[['YearsExperience']]

y = data['Salary']

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

**# Train a simple linear regression model**

model = LinearRegression()

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

# Make predictions on the test set

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

**# Evaluate the model**

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

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

print(f'Mean Squared Error: {mse}')

print(f'R-squared: {r2}')

**# Plot the regression line**

plt.scatter(X\_test, y\_test, label='Actual')

plt.plot(X\_test, y\_pred, color='red', linewidth=2, label='Predicted')

plt.xlabel('Years of Experience')

plt.ylabel('Salary')

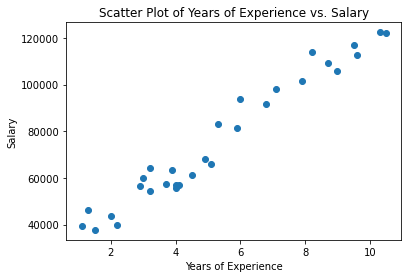
plt.title('Simple Linear Regression Model')

plt.legend()

plt.show()

**OUTPUT**

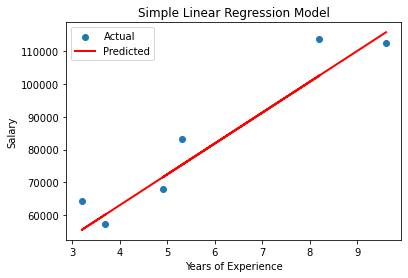
1)



2) Mean Squared Error: 49830096.85590839

R-squared: 0.9024461774180497

3)



---------------------------------------------------------------------------------------------------------------------------------