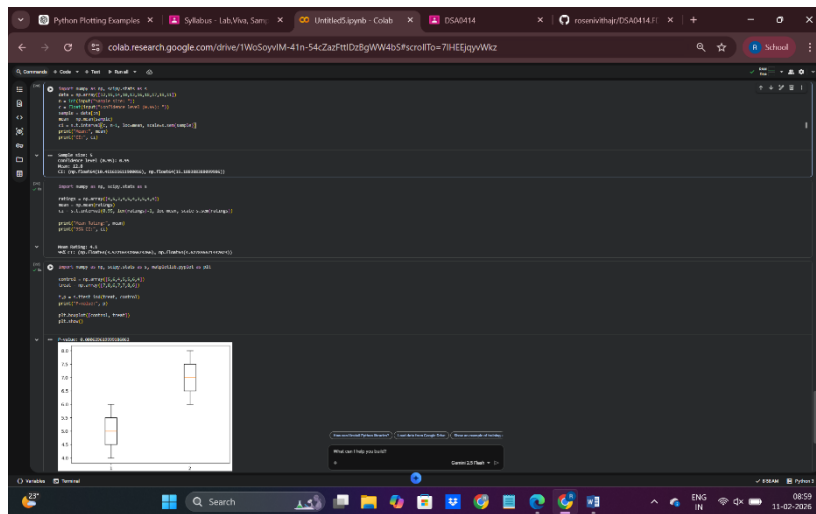


EXP 21,22,23



EXP 24,25,26

```
from sklearn.neighbors import KNeighborsClassifier
X = [[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10], [10, 11]]
y = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1]
k = 3
model = KNeighborsClassifier(n_neighbors=k)
model.fit(X, y)
print(model.predict([10, 11]))
```

```
from sklearn.neighbors import KNeighborsClassifier
X = [[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10], [10, 11]]
y = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1]
k = 3
model = KNeighborsClassifier(n_neighbors=k)
model.fit(X, y)
print(model.predict([10, 11]))
```

```
from sklearn.linear_model import LogisticRegression
X = [[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10], [10, 11]]
y = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1]
model = LogisticRegression()
model.fit(X, y)
print(model.predict([10, 11]))
```

EXP 27,28,29

```
from sklearn.linear_model import LogisticRegression
X = [[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10], [10, 11]]
y = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1]
model = LogisticRegression()
model.fit(X, y)
print(model.predict([10, 11]))
```

```
from sklearn.cluster import KMeans
X = [[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10], [10, 11]]
y = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1]
k = 3
model = KMeans(n_clusters=k)
model.fit(X)
print(model.predict([10, 11]))
```

```
from sklearn.metrics import accuracy_score
X = [[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10], [10, 11]]
y = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1]
model = LogisticRegression()
model.fit(X, y)
print(model.predict([10, 11]))
```

EXP 30

```
Python Plotting Examples x Syllabus - Lab.Viva, Sam x Untitled5.ipynb - Colab x DS40414 x rosenwithag/DS40414 x + x
colab.research.google.com/drive/1WoSoyvIM-41n-54cZarFtIDzBgWW4b5#scrollTo=q7z2Dw-0uTVQ

from sklearn.tree import DecisionTreeRegressor
import numpy as np

# Create some data
X = [[10000, 1], [10000, 2], [10000, 3], [10000, 4]]
y = [100000, 200000, 300000, 400000]

model = DecisionTreeRegressor(random_state=1)

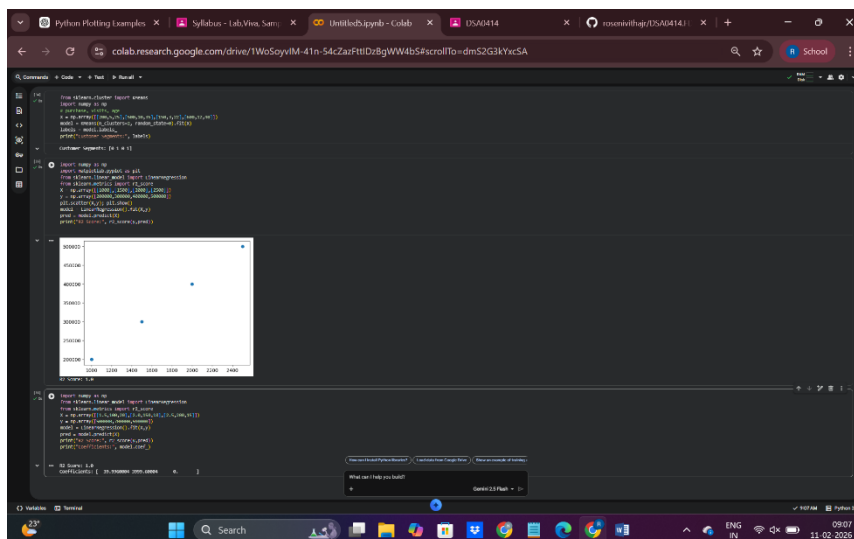
# Fit the model
model.fit(X, y)

# Make predictions
X_test = [[10000, 1], [10000, 2], [10000, 3], [10000, 4]]
y_test = [100000, 200000, 300000, 400000]

# Predict the target values
y_pred = model.predict(X_test)

# Print the predicted values
print("Predicted Prices: ", y_pred)
```

EXP 31,32,33



EXP 34,35

```
Python Plotting Examples x Syllabus - Lab.Viva, Sam x Untitled5.ipynb - Colab x DS40414 x rosenwithag/DS40414 x + x
colab.research.google.com/drive/1WoSoyvIM-41n-54cZarFtIDzBgWW4b5#scrollTo=nsEXMeCwYH

from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Create some data
X = [[10000, 1], [10000, 2], [10000, 3], [10000, 4]]
y = [100000, 200000, 300000, 400000]

# 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=1)

# Create a Decision Tree Regressor
model = DecisionTreeRegressor(random_state=1)

# Fit the model
model.fit(X_train, y_train)

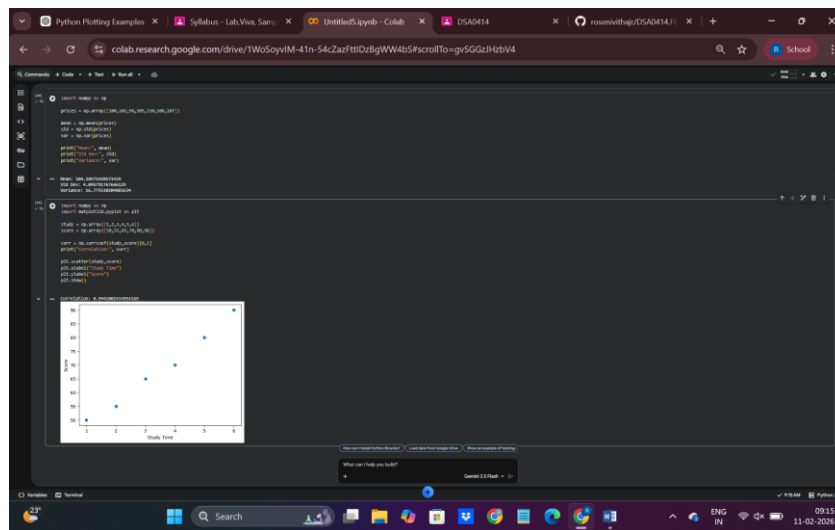
# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)

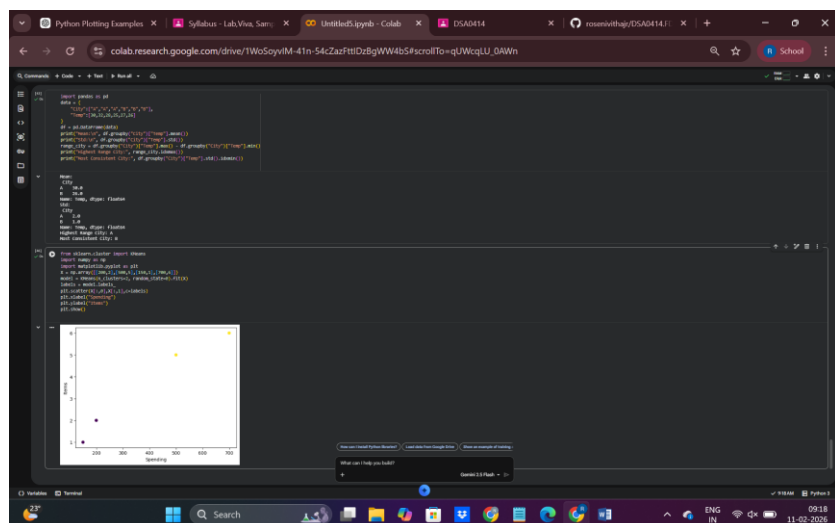
# Calculate the R-squared score
r2 = r2_score(y_test, y_pred)

# Print the MSE and R-squared score
print("MSE: ", mse)
print("R-squared: ", r2)
```

EXP 36,37



EXP 38,39



EXP 40

