Spring 2024 CS5720

Neural Networks & Deep Learning - Assignment 4

Name: Sravya Reddy Pilli

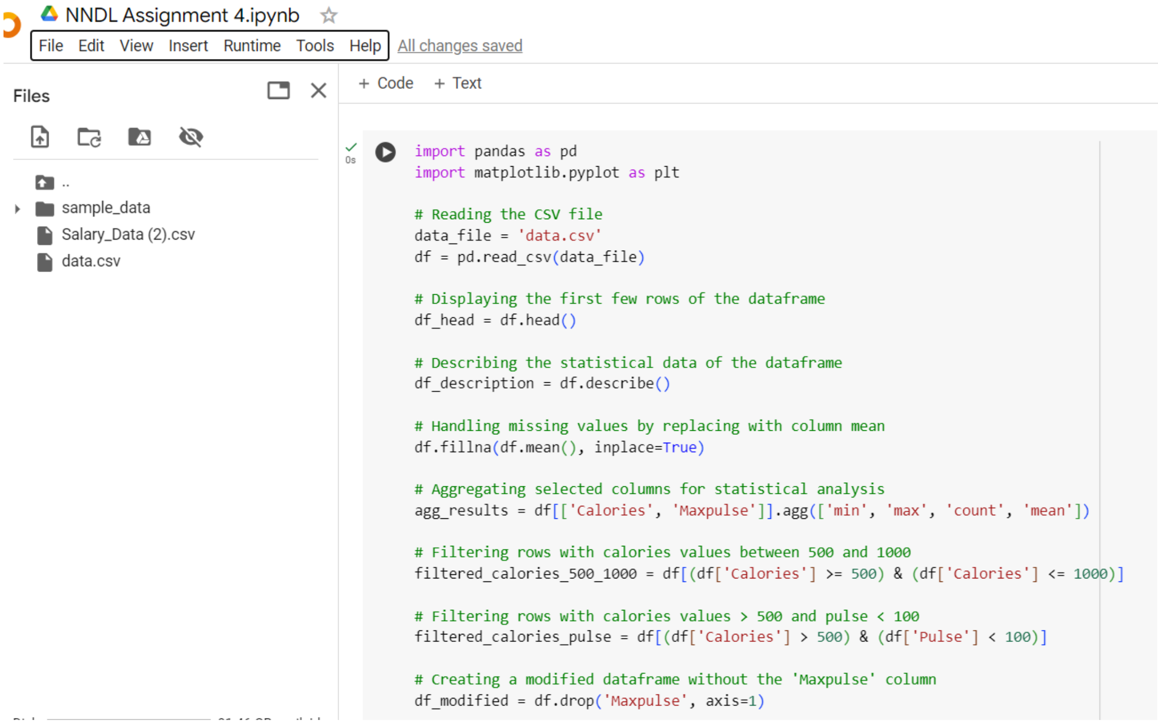
Student Id: 700747154

Github link : https://github.com/09sravyareddy/NNDL_Assignment4

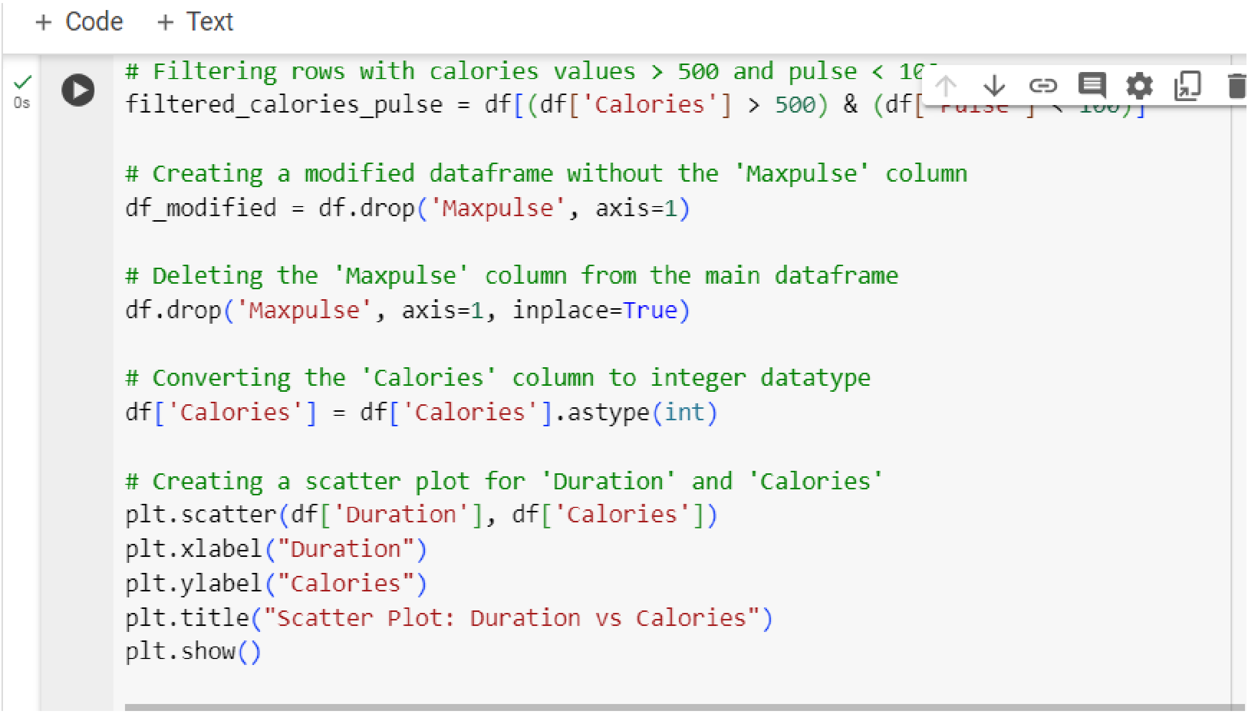
Recording Link :
https://drive.google.com/file/d/161W6yLQHap_z7M_u2JYkJBsPyat3igUy/view?usp=drive_link

Code & Output:

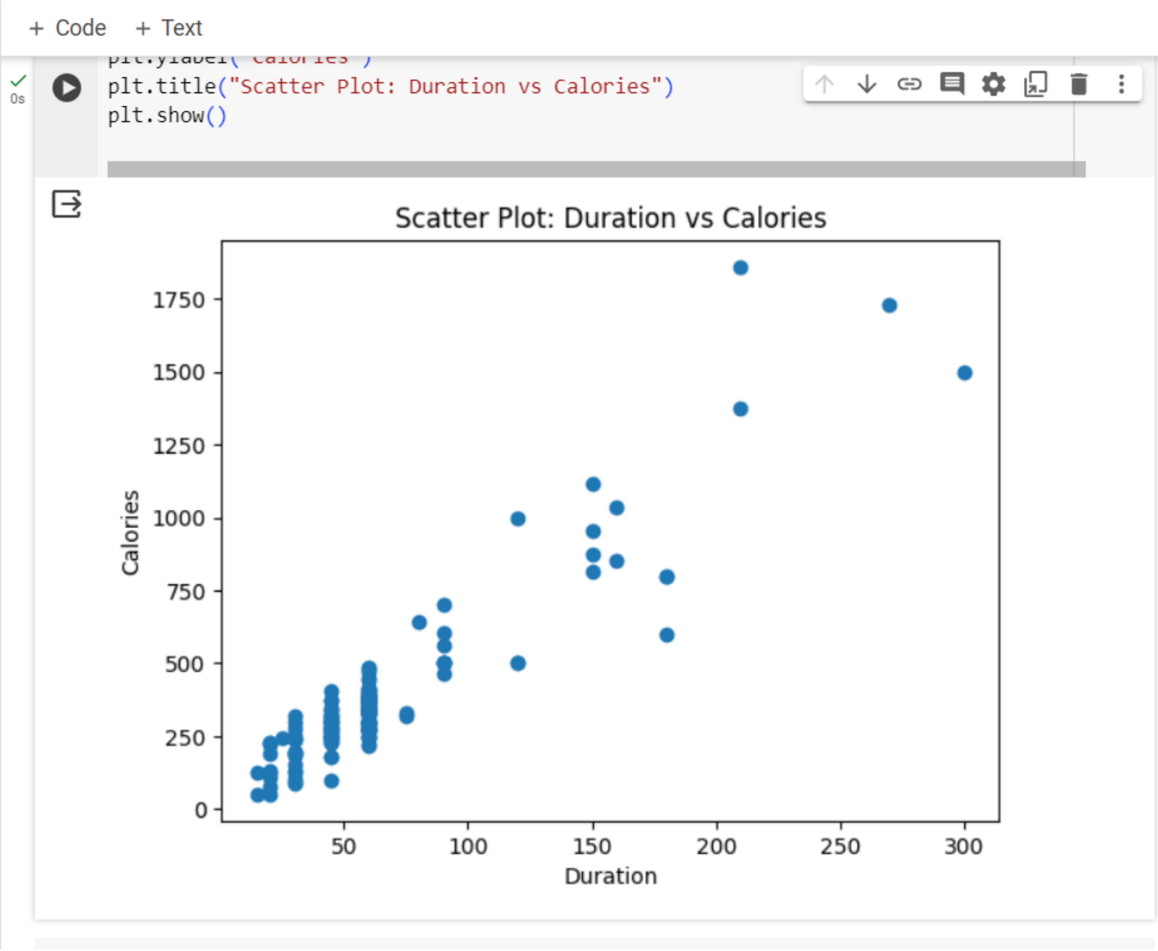
1)



```
import pandas as pd
import matplotlib.pyplot as plt

# Reading the CSV file
data_file = 'data.csv'
df = pd.read_csv(data_file)

# Displaying the first few rows of the dataframe
df_head = df.head()

# Describing the statistical data of the dataframe
df_description = df.describe()

# Handling missing values by replacing with column mean
df.fillna(df.mean(), inplace=True)

# Aggregating selected columns for statistical analysis
agg_results = df[['Calories', 'Maxpulse']].agg(['min', 'max', 'count', 'mean'])

# Filtering rows with calories values between 500 and 1000
filtered_calories_500_1000 = df[(df['Calories'] >= 500) & (df['Calories'] <= 1000)]

# Filtering rows with calories values > 500 and pulse < 100
filtered_calories_pulse = df[(df['Calories'] > 500) & (df['Pulse'] < 100)]

# Creating a modified dataframe without the 'Maxpulse' column
df_modified = df.drop('Maxpulse', axis=1)
```

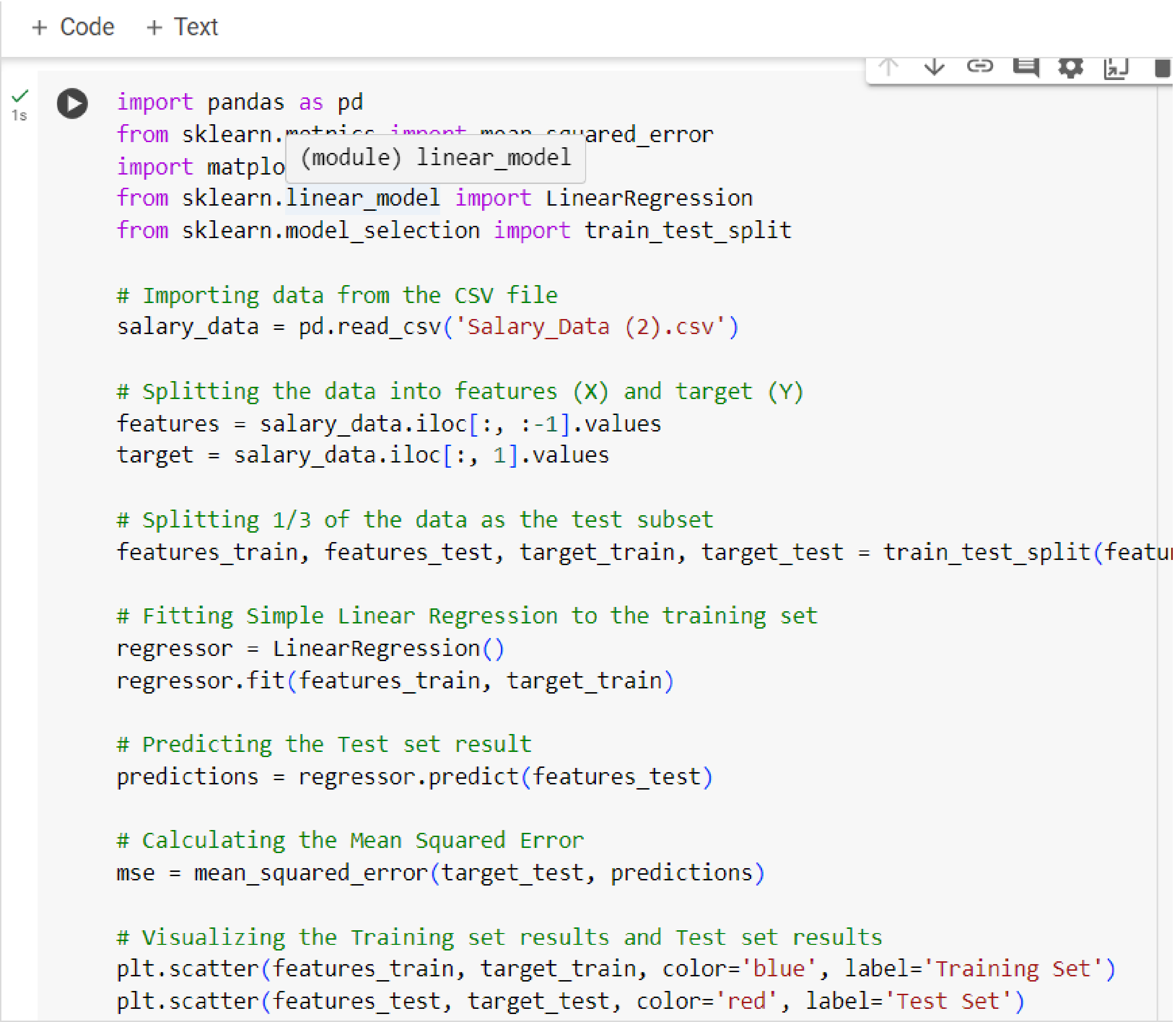


```
# Filtering rows with calories values > 500 and pulse < 10
filtered_calories_pulse = df[(df['Calories'] > 500) & (df['Pulse'] < 100)]

# Creating a modified dataframe without the 'Maxpulse' column
df_modified = df.drop('Maxpulse', axis=1)

# Deleting the 'Maxpulse' column from the main dataframe
df.drop('Maxpulse', axis=1, inplace=True)

# Converting the 'Calories' column to integer datatype
df['Calories'] = df['Calories'].astype(int)

# Creating a scatter plot for 'Duration' and 'Calories'
plt.scatter(df['Duration'], df['Calories'])
plt.xlabel("Duration")
plt.ylabel("Calories")
plt.title("Scatter Plot: Duration vs Calories")
plt.show()
```

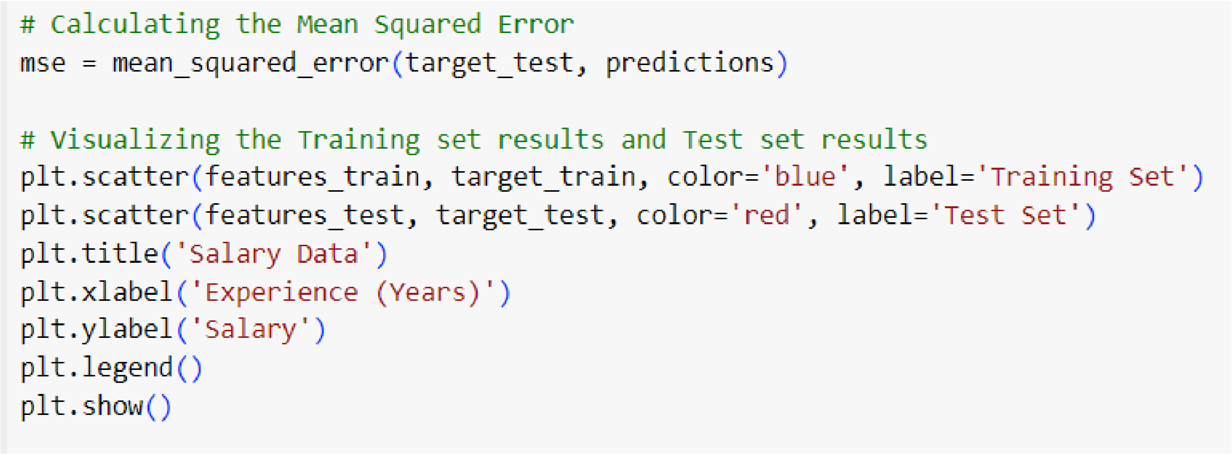


```
plt.ylabel("Calories")
plt.title("Scatter Plot: Duration vs Calories")
plt.show()
```

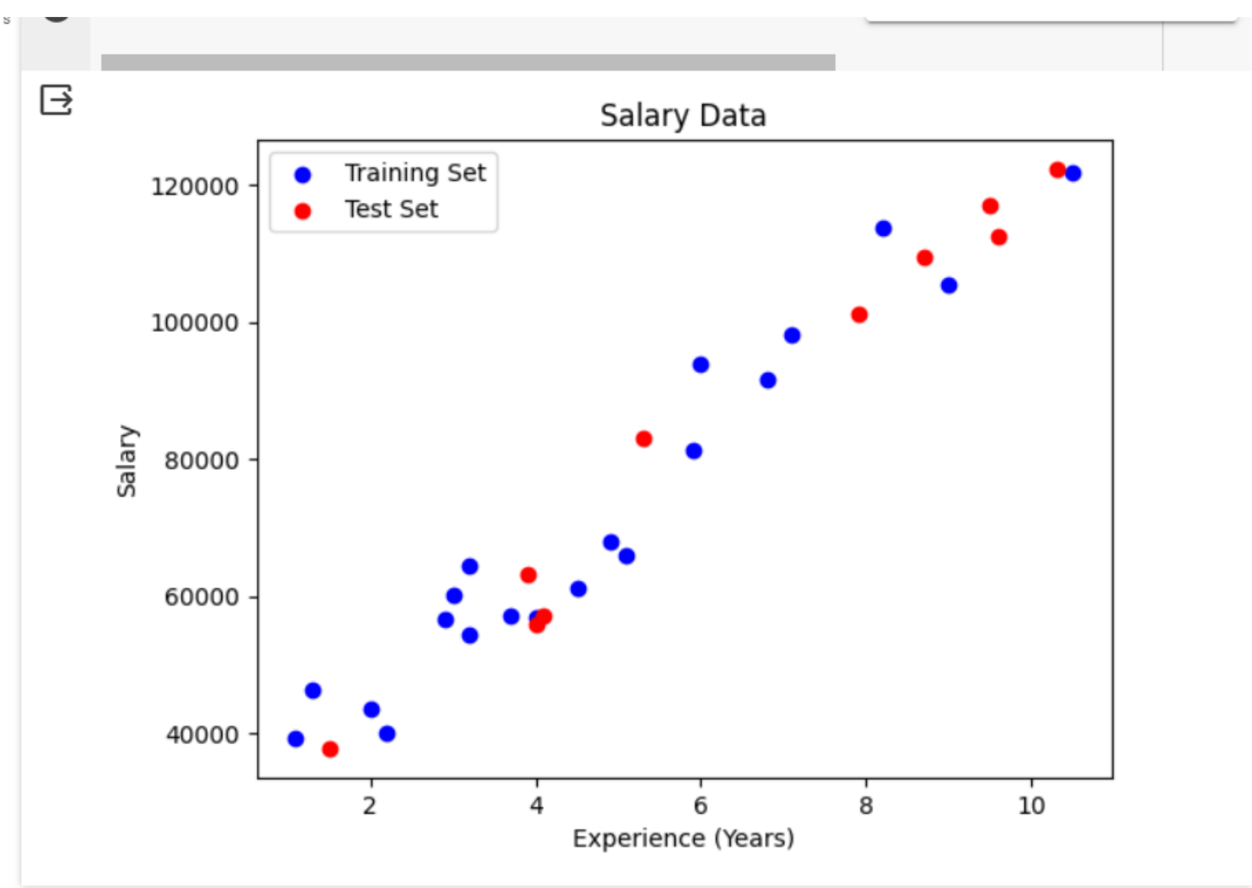
2)



```
import pandas as pd
from sklearn.metrics import mean_squared_error
import matplo
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Importing data from the CSV file
salary_data = pd.read_csv('Salary_Data (2).csv')

# Splitting the data into features (X) and target (Y)
features = salary_data.iloc[:, :-1].values
target = salary_data.iloc[:, 1].values

# Splitting 1/3 of the data as the test subset
features_train, features_test, target_train, target_test = train_test_split(featu

# Fitting Simple Linear Regression to the training set
regressor = LinearRegression()
regressor.fit(features_train, target_train)

# Predicting the Test set result
predictions = regressor.predict(features_test)

# Calculating the Mean Squared Error
mse = mean_squared_error(target_test, predictions)

# Visualizing the Training set results and Test set results
plt.scatter(features_train, target_train, color='blue', label='Training Set')
plt.scatter(features_test, target_test, color='red', label='Test Set')
```



```
# Calculating the Mean Squared Error
mse = mean_squared_error(target_test, predictions)

# Visualizing the Training set results and Test set results
plt.scatter(features_train, target_train, color='blue', label='Training Set')
plt.scatter(features_test, target_test, color='red', label='Test Set')
plt.title('Salary Data')
plt.xlabel('Experience (Years)')
plt.ylabel('Salary')
plt.legend()
plt.show()
```


Salary Data
● Training Set
● Test Set
Salary
Experience (Years)