

Assignment – 3

Name : Ganesh konagalla

Student Id : 700756412

Github Link :

<https://github.com/ganeshkonagalla123/Neural-Networks/upload/main/Assignment3>

1st Program Code :

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
file_path = "data.csv"
```

```
# Read the CSV file into a DataFrame
```

```
df = pd.read_csv(file_path)
```

```
# Show basic statistical description about the data
```

```
print("Statistical Description of the Data:")
```

```
print(df.describe())
```

```
# Check for null values and replace them with the mean of their respective column
```

```
df.fillna(df.mean(), inplace=True)
```

```
# Convert the datatype of Calories column to int after replacing null values
```

```
df['Calories'] = df['Calories'].astype(int)
```

```
# Aggregate selected columns
```

```
aggregated_data = df[['Duration', 'Calories']].agg(['min', 'max', 'count', 'mean'])
```

Filter the DataFrame for specific conditions

```
filter1 = df[(df['Calories'] >= 500) & (df['Calories'] <= 1000)]
```

```
filter2 = df[(df['Calories'] > 500) & (df['Pulse'] < 100)]
```

Create a modified DataFrame without the "Maxpulse" column

```
df_modified = df.drop(columns=['Maxpulse'])
```

Delete the "Maxpulse" column from the main DataFrame

```
df.drop(columns=['Maxpulse'], inplace=True)
```

Creating a scatter plot for Duration vs Calories

```
plt.figure(figsize=(10, 6))
```

```
plt.scatter(df['Duration'], df['Calories'], color='blue', alpha=0.5)
```

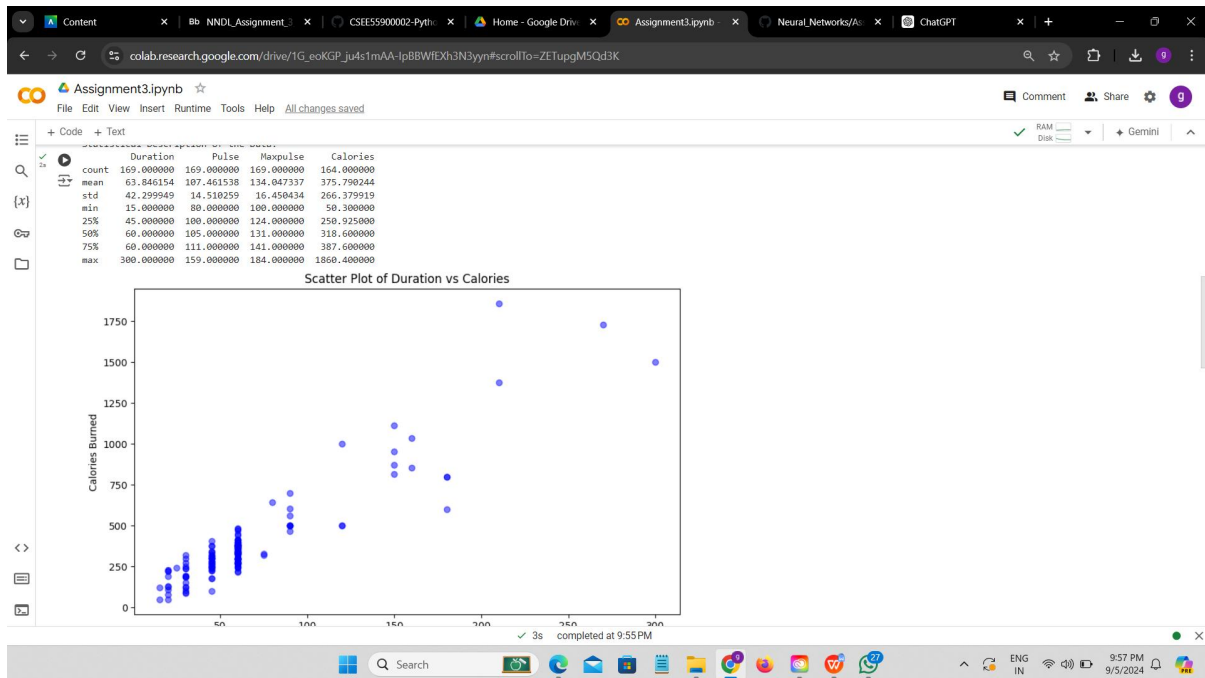
```
plt.title('Scatter Plot of Duration vs Calories')
```

```
plt.xlabel('Duration (minutes)')
```

```
plt.ylabel('Calories Burned')
```

```
plt.show()
```

Output :



2nd Program Code :

```
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Step a: Load the dataset

file_path = "Salary_Data.csv" # Update this path
df_salary = pd.read_csv(file_path)

# Step b: Splitting the data into training and testing sets

X = df_salary.iloc[:, :-1].values # Independent variable is the first column
y = df_salary.iloc[:, 1].values # Dependent variable (Salary) is the second column

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3, random_state=0)

# Step c: Linear Regression model on the training set

regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Step d: Predict the Test set results

y_pred = regressor.predict(X_test)

# Step e: Calculating the mean squared error

mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

Step f: Visualize both train and test data using scatter plot

```
plt.scatter(X_train, y_train, color='blue', label='Training data')
```

```
plt.scatter(X_test, y_test, color='red', label='Test data')
```

```
plt.plot(X_test, y_pred, color='black', label='Prediction')
```

```
plt.title('Salary vs. Experience (Training and Test set)')
```

```
plt.xlabel('Years of Experience')
```

```
plt.ylabel('Salary')
```

```
plt.legend()
```

```
plt.show()
```

Output :

