**Spring 2024: CS5720 Neural Networks & Deep Learning - ICP- 4**

**Sai Preetham Nagaswaram (700743350)**

GitHub Link:[https://github.com/Saipreethamnagaswaram/Neural-Network-Deep-Learning/tree/main/Assignment4](https://github.com/Saipreethamnagaswaram/Neural-Network-Deep-Learning/tree/main/Assignment1)

**1. Data Manipulation**

a. Read the provided CSV file ‘data.csv’.

b. https://drive.google.com/drive/folders/1h8C3mLsso-R-sIOLsvoYwPLzy2fJ4IOF?usp=sharing

c. Show the basic statistical description about the data.

d. Check if the data has null values.

i. Replace the null values with the mean

e. Select at least two columns and aggregate the data using: min, max, count, mean.

f. Filter the dataframe to select the rows with calories values between 500 and 1000.

g. Filter the dataframe to select the rows with calories values > 500 and pulse < 100.

h. Create a new “df\_modified” dataframe that contains all the columns from df except for

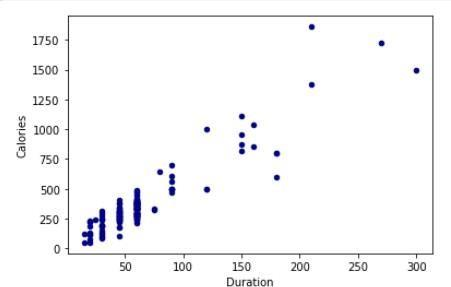
“Maxpulse”.

i. Delete the “Maxpulse” column from the main df dataframe

j. Convert the datatype of Calories column to int datatype.

k. Using pandas create a scatter plot for the two columns (Duration and Calories).

Example:



**Solution:**

**import** pandas

**from** matplotlib **import** pyplot **as** plt

df **=** pandas.read\_csv('data.csv')

description **=** df.describe()

print(" DATAFRAME DESCRIPTION ")

print("")

print(description)

print("")

null\_columns **=** df.columns[df.isnull().any()]

**for** c **in** null\_columns:

mean\_value **=** df[c].mean()

df[c].fillna(mean\_value, inplace**=True**)

selected\_columns **=** ['Calories', 'Duration']

aggregated\_data **=** df[selected\_columns].agg(['min', 'max', 'count', 'mean'])

print(" AGGREGATED DATA ")

print("")

print(aggregated\_data)

print("")

filtered\_df1 **=** df[(df['Calories'] **>=** 500) **&** (df['Calories'] **<=** 1000)]

filtered\_df2 **=** df[(df['Calories'] **>** 500) **&** (df['Pulse'] **<** 100)]

df\_modified **=** df.drop(columns**=**['Maxpulse'])

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

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

plt.scatter(df['Duration'], df['Calories'])

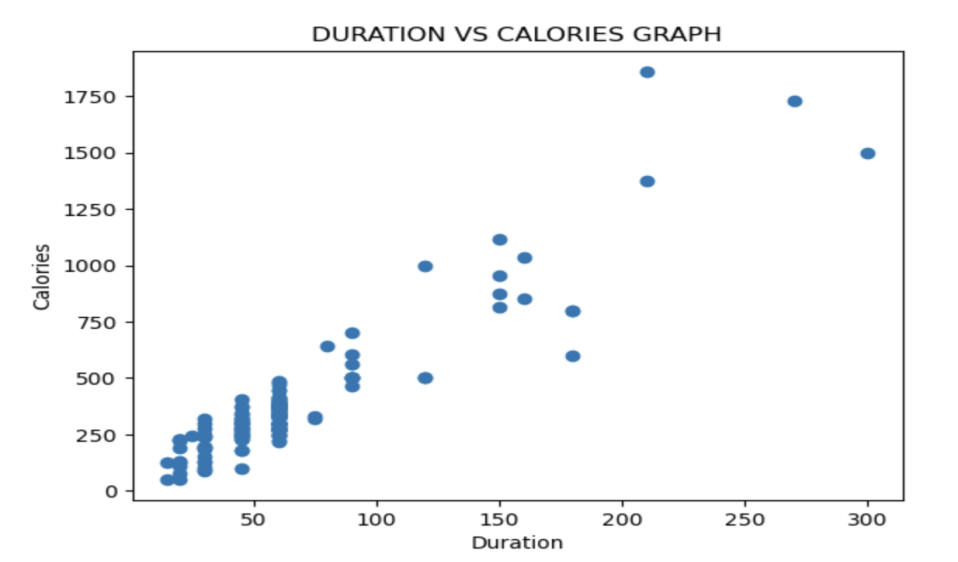
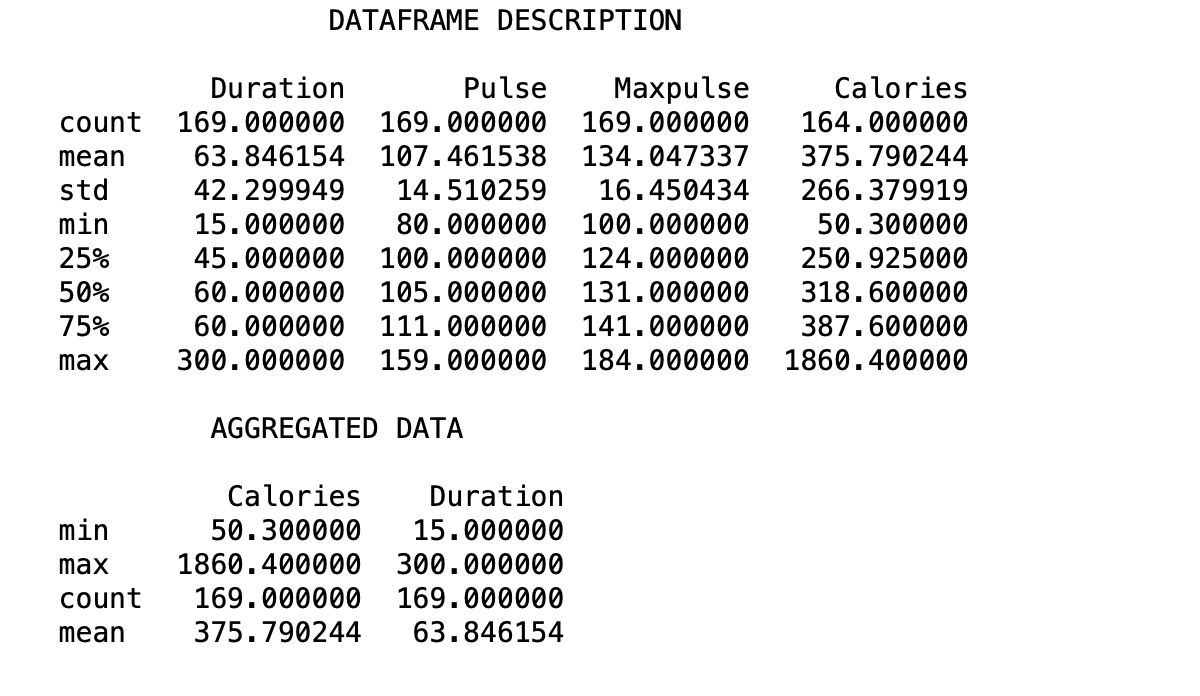
plt.xlabel('Duration')

plt.ylabel('Calories')

plt.title('DURATION VS CALORIES GRAPH')

plt.show()

**Output:**

****

**2. Linear Regression**

a) Import the given “Salary\_Data.csv”

b) Split the data in train\_test partitions, such that 1/3 of the data is reserved as test subset.

c) Train and predict the model.

d) Calculate the mean\_squared error

e) Visualize both train and test data using scatter plot.

**Solution:**

**import** pandas **as** pd

**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

**from** matplotlib **import** pyplot **as** plt

data **=** pd.read\_csv("Salary\_Data.csv")

x **=** data[['YearsExperience']]

y **=** data[['Salary']]

x\_train, x\_test, y\_train, y\_test **=** train\_test\_split(x, y, test\_size**=**1**/**3, random\_state**=**0)

regressor **=** LinearRegression()

regressor.fit(x\_train, y\_train)

y\_pred **=** regressor.predict(x\_test)

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

print("Mean Squared Error:", mse)

plt.scatter(x\_train, y\_train, label**=**"Training Data")

plt.scatter(x\_test, y\_test, label**=**"Test Data")

plt.plot(x\_test, y\_pred, label**=**"Linear Regression")

plt.xlabel("Years of Experience")

plt.ylabel("Salary")

plt.legend()

plt.title("Linear Regression: Salary vs Years of Experience")

plt.show()

**Output:**

****