Neural Networks Assignment – 4

ID:700739769

Name: Anjani Priya Marthati

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 thetwo columns (Duration and Calories).

In [5]:

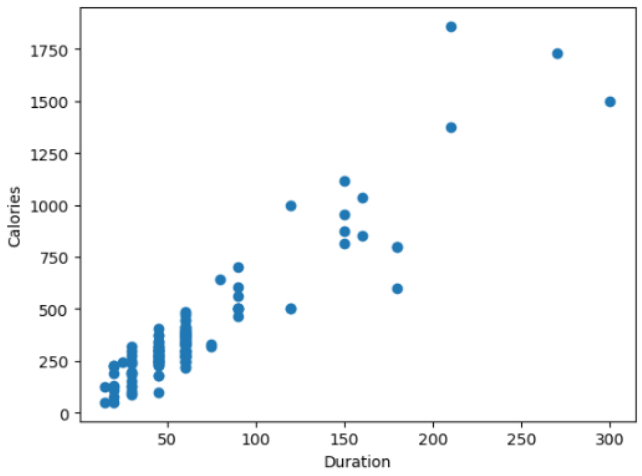
```python
import pandas as pd
import matplotlib.pyplot as plt
import os

# Read the CSV file into a Pandas dataframe
df = pd.read_csv('C:\\Neural networks\\data.csv')
#Show the basic statistical description about the data
print("Statistics of Data:\n{} \n".format(df.describe()))
# Check for null values
print("Number of null Values in data per column: \n{} \n".format(df.isnull().sum()))
# Replace null values with the mean
df.fillna(df.mean(), inplace=True)
#Select at least two columns and aggregate the data using: min, max, count, mean.
cols = ['Duration', 'Calories']
agg = df[cols].agg(['min', 'max', 'count', 'mean'])
print("Aggrigate data of two columns (Duration, Calories) : \n {} \n".format(agg))
# Filter data with calories between 500 and 1000
df_500_1000 = df[(df['Calories'] >= 500) & (df['Calories'] <= 1000)]
print("Data with calories between 500 and 1000: \n {} \n".format(df_500_1000))
# Filter data with calories > 500 and pulse < 100
df_500_pulse = df[(df['Calories'] > 500) & (df['Pulse'] < 100)]
print("Data with calories > 500 and pulse < 100: \n {} \n".format(df_500_pulse))
# Create new dataframe without "Maxpulse" column
df_modified = df.drop('Maxpulse', axis=1)
# Delete "Maxpulse" column from the main df dataframe
df.drop('Maxpulse', axis=1, inplace=True)
# Convert "Calories" column to int datatype
df['Calories'] = df['Calories'].astype(int)
# Scatter plot for "Duration" and "Calories"

plt.scatter(df['Duration'], df['Calories'])
plt.xlabel('Duration')
plt.ylabel('Calories')
plt.show()
```


```python
df.drop('Maxpulse', axis=1, inplace=True)
# Convert "Calories" column to int datatype
df['Calories'] = df['Calories'].astype(int)
# Scatter plot for "Duration" and "Calories"

plt.scatter(df['Duration'], df['Calories'])
plt.xlabel('Duration')
plt.ylabel('Calories')
plt.show()
```


```
Statistics of Data:
        Duration       Pulse    Maxpulse     Calories
count  169.000000  169.000000  169.000000   164.000000
mean    63.846154  107.461538  134.047337   375.790244
std     42.299949   14.510259   16.450434   266.379919
min     15.000000   80.000000  100.000000    50.300000
25%     45.000000  100.000000  124.000000   250.925000
50%     60.000000  105.000000  131.000000   318.600000
75%     60.000000  111.000000  141.000000   387.600000
max    300.000000  159.000000  184.000000  1860.400000

Number of null Values in data per column:
Duration    0
Pulse       0
Maxpulse    0
Calories    5
dtype: int64

Aggrigate data of two columns (Duration, Calories) :
         Duration     Calories
min     15.000000    50.300000
max    300.000000  1860.400000
count  169.000000   169.000000
mean    63.846154   375.790244
```

```
Data with calories between 500 and 1000:
     Duration  Pulse  Maxpulse  Calories
51         80    123       146     643.1
62        160    109       135     853.0
65        180     90       130     800.4
66        150    105       135     873.4
67        150    107       130     816.0
72         90    100       127     700.0
73        150     97       127     953.2
75         90     98       125     563.2
78        120    100       130     500.4
83        120    100       130     500.0
90        180    101       127     600.1
99         90     93       124     604.1
101        90     90       110     500.0
102        90     90       100     500.0
103        90     90       100     500.4
106       180     90       120     800.3
108        90     90       120     500.3

Data with calories > 500 and pulse < 100:
     Duration  Pulse  Maxpulse  Calories
65        180     90       130     800.4
70        150     97       129    1115.0
73        150     97       127     953.2
75         90     98       125     563.2
99         90     93       124     604.1
103        90     90       100     500.4
106       180     90       120     800.3
108        90     90       120     500.3
```

```
75         90     98       125     563.2
99         90     93       124     604.1
103        90     90       100     500.4
106       180     90       120     800.3
108        90     90       120     500.3
```



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.

In [9]:

```
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
import matplotlib.pyplot as plt

# Import the data
df = pd.read_csv("C:\\Neural networks\\Salary_Data (2).csv")
#Split the data in train_test partitions, such that 1/3 of the data is reserved as test subset
X = df[['YearsExperience']]
y = df[['Salary']]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3, random_state=0)
# Train and predict the model
reg = LinearRegression()
reg.fit(X_train, y_train)
y_pred = reg.predict(X_test)

# Calculate the mean squared error
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error: ", mse)

# Visualize the train and test data using scatter plot
plt.scatter(X_train, y_train, color='black')
plt.scatter(X_test, y_test, color='red')
plt.plot(X_train, reg.predict(X_train), color='orange')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.title('Training and Test data')
plt.show()
```

```
Mean Squared Error:  21026037.329511296
```

In [ ]:

Video Link:

https://vimeo.com/manage/videos/908491809/e83b72fb19?studio_recording=true&record_session_id=ca777c5e-9963-46ab-942b-73c9fa1c3d31

Github Link:

https://github.com/Priyamarthati/Assignment4