

Spring 2024: CS5720 Neural Networks & Deep Learning - ICP-4
Bhanu Chandrika Lakkimsetti (700747439)

GitHub Link: https://github.com/bhanuchandrika99/NNDL_ICP_4

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.

```
In [4]: import pandas as pd
import matplotlib.pyplot as plt

# Read the CSV file into a Pandas dataframe
df = pd.read_csv(r'C:\Users\bhanu\OneDrive\Documents\UCM\Spring24\Neural Networks\ICP_4_Spring24\data.csv')
print("Statistics of Data:\n{} \n".format(df.describe()))
```

```
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     90.000000   159.000000   184.000000  1860.400000
```

- d. Check if the data has null values.
 - i. Replace the null values with the mean

```
In [6]: # 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)
```

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

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

```
In [8]: # Aggregate data for two columns
cols = ['Duration', 'Calories']
agg = df[cols].agg(['min', 'max', 'count', 'mean'])
print("Aggregate data of two columns (Duration, Calories) : \n {} \n".format(agg))

Aggregate 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
```

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

```
In [9]: # 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))

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
```

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

```
In [10]: # 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))

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
```

- 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

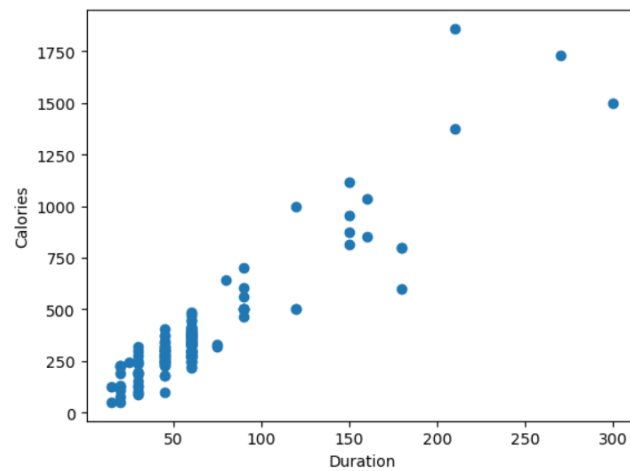
```
In [11]: # Create new dataframe without "Maxpulse" column  
df_modified = df.drop('Maxpulse', axis=1)
```

```
In [12]: # Delete "Maxpulse" column from the main df dataframe  
df.drop('Maxpulse', axis=1, inplace=True)
```

- j. Convert the datatype of Calories column to int datatype.
- k. Using pandas create a scatter plot for the two columns (Duration and Calories).

```
In [14]: # Scatter plot for "Duration" and "Calories"
```

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



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 [3]: 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(r'C:\Users\bhanu\OneDrive\Documents\UCM\Spring24\Neural Networks\ICP_4_Spring24\Salary_Data.csv')

# Split the data into train and test partitions
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

