

Spring 2024: CS5720 – NN &DL

In-Class Programming Assignment-4

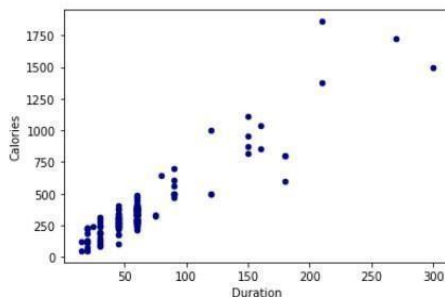
NAME: Sai Deva Pranay Kumar Rao, Guddity

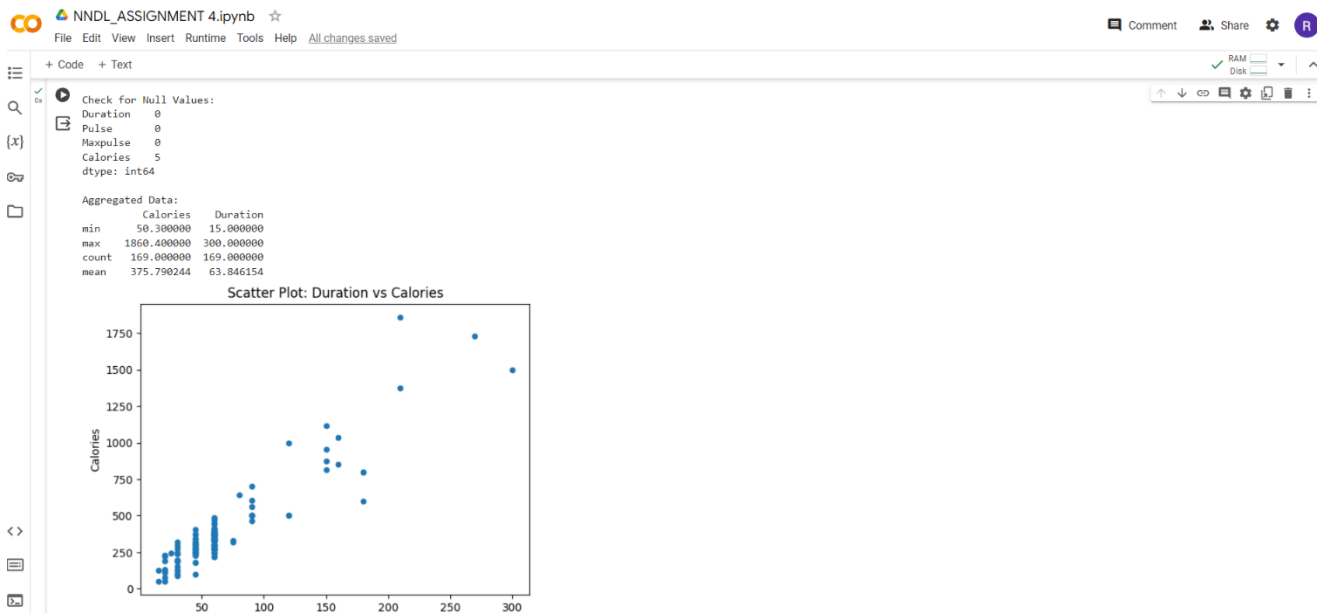
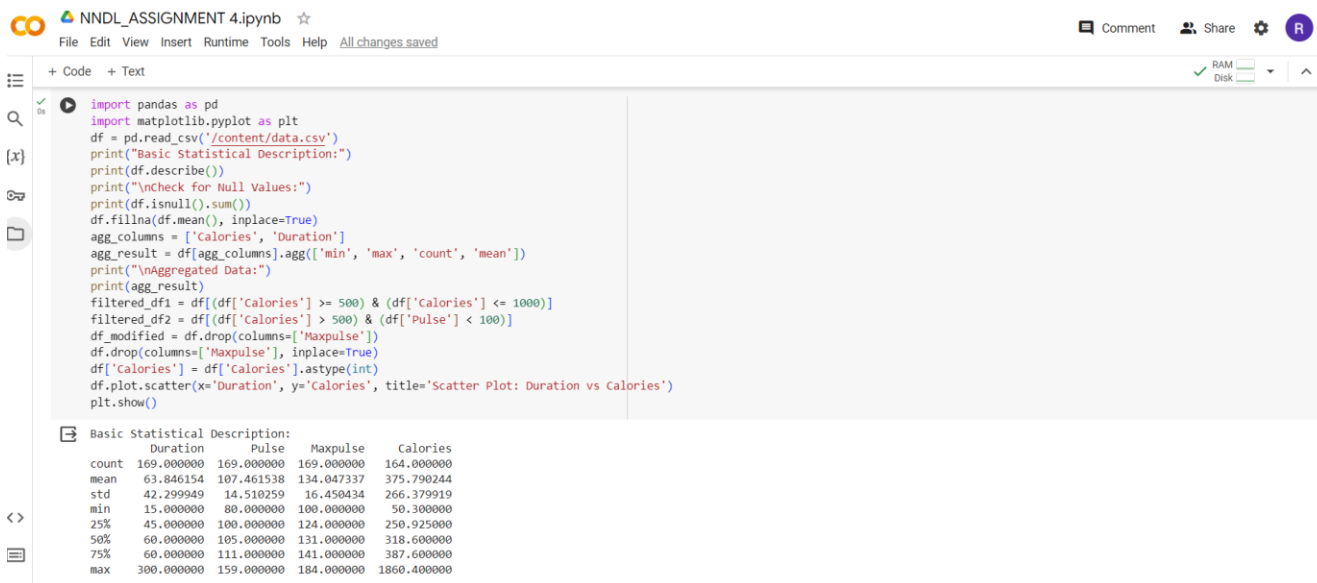
700#: 700745063

Github Link : https://github.com/raopranay1999/Neural-Network-Assignment-4/blob/main/NNDL_ASSIGNMENT_4.ipynb

1. Data Manipulation

- Read the provided CSV file 'data.csv'.
- <https://drive.google.com/drive/folders/1h8C3mLsso-R-sIOLsvoYwPLzy2fJ4IOF?usp=sharing>
- Show the basic statistical description about the data.
- Check if the data has null values.
 - Replace the null values with the mean
- Select at least two columns and aggregate the data using: min, max, count, mean.
- Filter the dataframe to select the rows with calories values between 500 and 1000.
- Filter the dataframe to select the rows with calories values > 500 and pulse < 100.
- Create a new "df_modified" dataframe that contains all the columns from df except for "Maxpulse".
- Delete the "Maxpulse" column from the main df dataframe
- Convert the datatype of Calories column to int datatype.
- Using pandas create a scatter plot for the two columns (Duration and Calories).Example





2. Linear Regression

- Import the given "Salary_Data.csv"
- Split the data in train_test partitions, such that 1/3 of the data is reserved as test subset.
- Train and predict the model.
- Calculate the mean_squared error
- Visualize both train and test data using scatter plot.

```
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

# a) Import the given "Salary_Data.csv"
data = pd.read_csv("/content/Salary_Data (2).csv")
print("First few rows of the data frame")
print(data.head())

# b) Split the data into train_test partitions
X = data[['YearsExperience']] # Assuming the independent variable is in the 'YearsExperience' column
y = data['Salary']           # Assuming the dependent variable is in the 'Salary' column

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)

# c) Train and predict the model
model = LinearRegression()
model.fit(X_train, y_train)
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# d) Calculate the mean_squared error
mse_train = mean_squared_error(y_train, y_train_pred)
mse_test = mean_squared_error(y_test, y_test_pred)

print(f"Mean Squared Error (Train): {mse_train}")
print(f"Mean Squared Error (Test): {mse_test}")
```

```
# e) Visualize both train and test data using scatter plot
plt.scatter(X_train, y_train, color='blue', label='Train Data')
plt.scatter(X_test, y_test, color='red', label='Test Data')
plt.plot(X_train, y_train_pred, color='green', linewidth=2, label='Regression Line')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.title('Salary Prediction Model')
plt.legend()
plt.show()
```

First few rows of the data frame

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

Mean Squared Error (Train): 29793161.082422983
Mean Squared Error (Test): 35301898.887134895

