

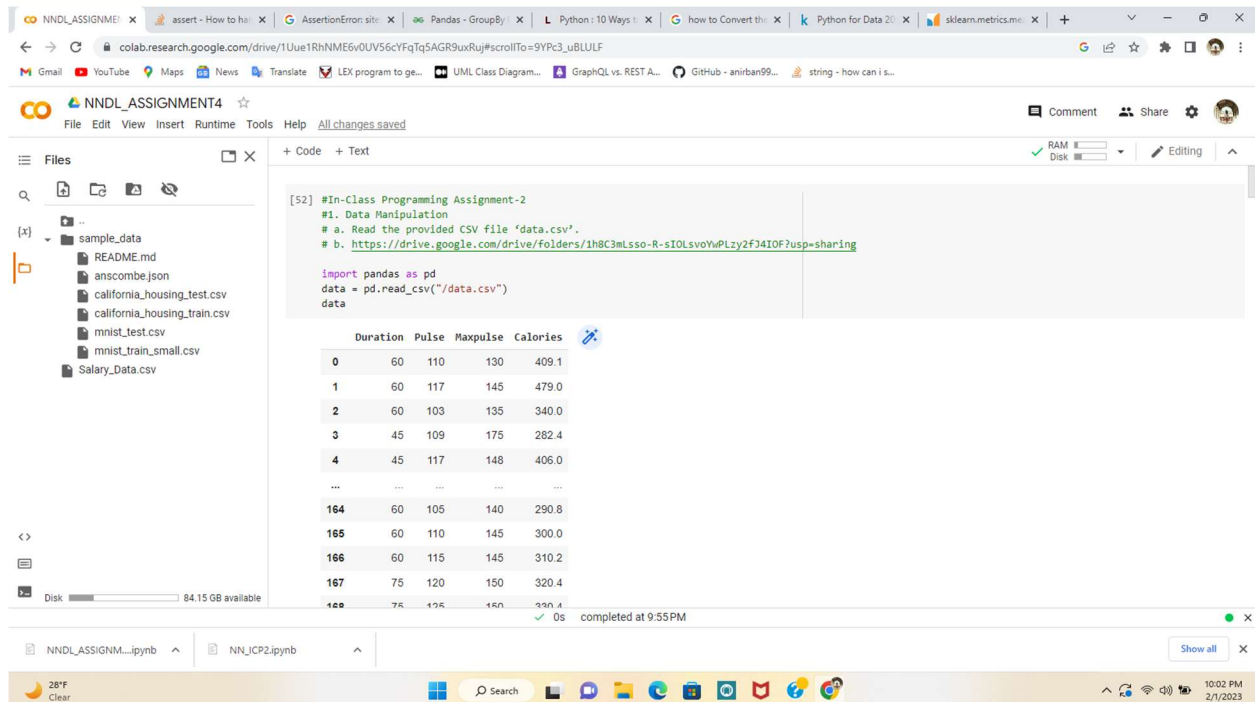
Spring 2023: CS5720 – NN &DL In-Class Programming Assignment-2

1. Data Manipulation

a. Read the provided CSV file 'data.csv'.

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

```
import pandas as pd
data = pd.read_csv("/data.csv")
data
```



The screenshot shows a Google Colab notebook titled "NNDL_ASSIGNMENT4". The code cell [52] contains the following code:

```
#In-Class Programming Assignment-2
#1. Data Manipulation
# a. Read the provided CSV file 'data.csv'.
# b. https://drive.google.com/drive/folders/1h8C3mLsso-R-sIOLsvoYwPLzy2fJ4IOF?usp=sharing

import pandas as pd
data = pd.read_csv("/data.csv")
data
```

The output of the code cell is a preview of the DataFrame:

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0
...
164	60	105	140	290.8
165	60	110	145	300.0
166	60	115	145	310.2
167	75	120	150	320.4
168	75	125	160	330.4

The notebook interface shows the file explorer on the left with a folder named "sample_data" containing several CSV files. The status bar at the bottom indicates that the code was completed at 9:55 PM.

Using pandas module, I have read the data.csv file and imported it and also uploaded in google collab notebook.

c. Show the basic statistical description about the data.

```
import pandas as pd
import numpy as np
```

```
# create a sample dataframe
#data = ("/data.csv")
df = pd.DataFrame(data)
```

```
mean = df.mean()
```

```

print("Mean:")
print(mean)

# calculate median
median = df.median()
print("\nMedian:")
print(median)

# calculate mode
mode = df.mode().iloc[0]
print("\nMode:")
print(mode)

# calculate variance
variance = df.var()
print("\nVariance:")
print(variance)

# calculate standard deviation
std_dev = df.std()
print("\nStandard Deviation:")
print(std_dev)

```

The screenshot shows a Google Colab notebook titled "NNDL_ASSIGNMENT4". The left sidebar displays a file explorer with a folder named "sample_data" containing several CSV files: "README.md", "anscombe.json", "california_housing_test.csv", "california_housing_train.csv", "mnist_test.csv", "mnist_train_small.csv", and "Salary_Data.csv". The main area shows a code cell with the following Python code:

```

print("\nVariance:")
print(variance)

# calculate standard deviation
std_dev = df.std()
print("\nStandard Deviation:")
print(std_dev)

```

The output of the code is displayed below the code cell, showing the statistical results for the "Duration" variable:

```

Mean:
Duration    63.846154
Pulse       107.461538
Maxpulse    134.047337
Calories     375.790244
dtype: float64

Median:
Duration    60.0
Pulse       105.0
Maxpulse    131.0
Calories     318.6
dtype: float64

Mode:
Duration    60.0
Pulse       100.0
Maxpulse    120.0
Calories     300.0
Name: 0, dtype: float64

Variance:

```

The output is completed at 9:55 PM. The bottom status bar shows the temperature as 28°F Clear and the time as 10:02 PM on 2/1/2023.

Using pandas and numpy I have shown the basic statistical description

```

#d.Check if the data has null values.
# i. Replace the null values with the mean
df.isnull().sum()
df = df.fillna(df.mean())

#e. Select at least two columns and aggregate the data using: min, max, count, mean
agg_df = df[['Duration', 'Calories']].agg({'Duration': ['min', 'max', 'count', 'mean'], 'Calories': ['min', 'max', 'count', 'mean']})

#f.Filter the dataframe to select the rows with calories values between 500 and 1000
result = df[df['Calories'].between(500,1000 )]
print(result)

```

In d,e,f I Have shown the required data using manipulations.

```

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

res=df[(df['Calories'] > 500) & (df['Pulse'] < 100)]
print(res)

#h. Create a new "df_modified" dataframe that contains all the columns from df except for
# "Maxpulse"
df_modified = df.drop('Maxpulse', axis=1)

```

colab.research.google.com/drive/1Uue1RhNME6v0UV56cYFqTq5AGR9uxRuj#scrollTo=9YPc3_uBLULF

NNDL_ASSIGNMENT4

File Edit View Insert Runtime Tools Help All changes saved

Files

- sample_data
 - README.md
 - anscombe.json
 - california_housing_test.csv
 - california_housing_train.csv
 - mnist_test.csv
 - mnist_train_small.csv
 - Salary_Data.csv

```
[58] #g.Filter the dataframe to select the rows with calories values > 500 and pulse < 100.

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

	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

```
[59] #h. Create a new "df_modified" dataframe that contains all the columns from df except for
# "Maxpulse"
df_modified = df.drop('Maxpulse', axis=1)
```

```
[60] df_modified
```

	Duration	Pulse	Calories
0	60	110	409.1
1	60	117	479.0

completed at 9:55 PM

28°F Clear 10:03 PM 2/1/2023

```
# i. Delete the "Maxpulse" column from the main df dataframe
df.drop('Maxpulse', axis=1, inplace=True)
```

```
#j. Convert the datatype of Calories column to int datatype
df['Calories'] = df['Calories'].astype(int)
```

```
df.dtypes
```

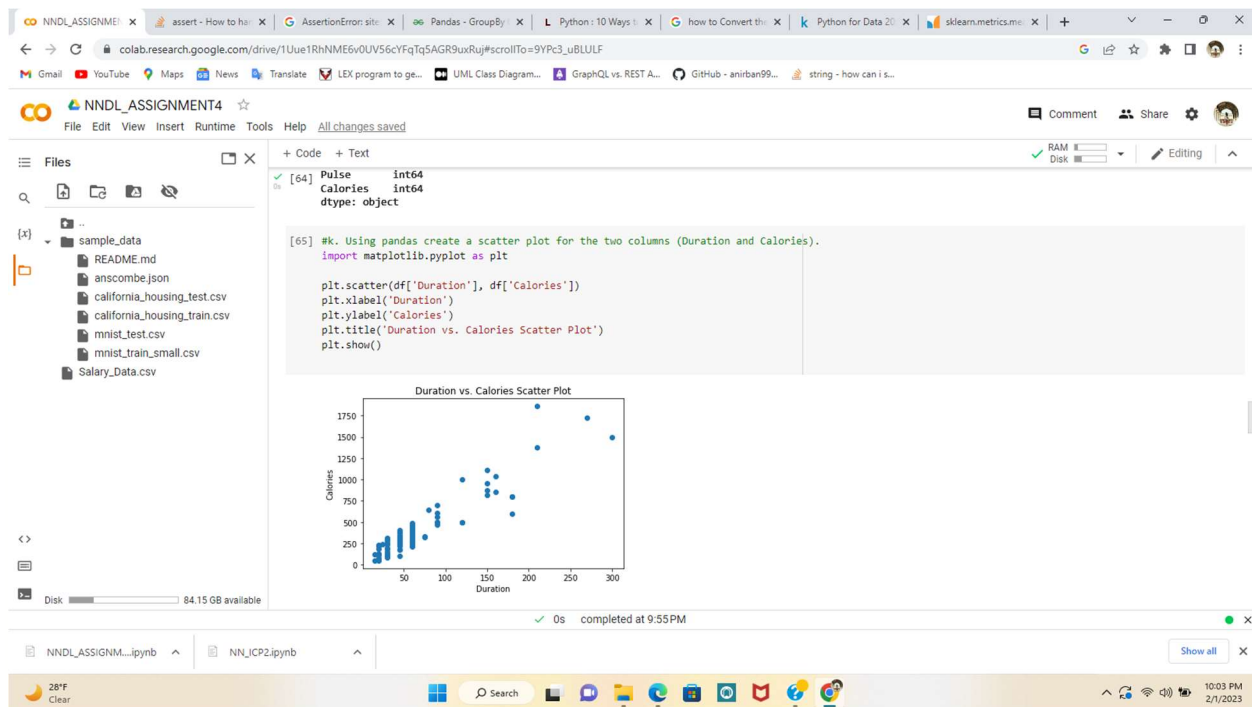
The screenshot shows a Google Colab notebook titled "NNDL_ASSIGNMENT4". The left sidebar displays a file explorer with a folder named "sample_data" containing files: "README.md", "anscombe.json", "california_housing_test.csv", "california_housing_train.csv", "mnist_test.csv", "mnist_train_small.csv", and "Salary_Data.csv". The main code area shows two cells. Cell [61] contains the code: `# i. Delete the "Maxpulse" column from the main df dataframe
df.drop('Maxpulse', axis=1, inplace=True)`. Cell [62] displays the resulting DataFrame `df` with columns "Duration", "Pulse", and "Calories". The DataFrame has 169 rows. The bottom status bar indicates "completed at 9:55 PM".

	Duration	Pulse	Calories
0	60	110	409.1
1	60	117	479.0
2	60	103	340.0
3	45	109	282.4
4	45	117	406.0
...
164	60	105	290.8
165	60	110	300.0
166	60	115	310.2
167	75	120	320.4
168	75	125	330.4

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

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

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



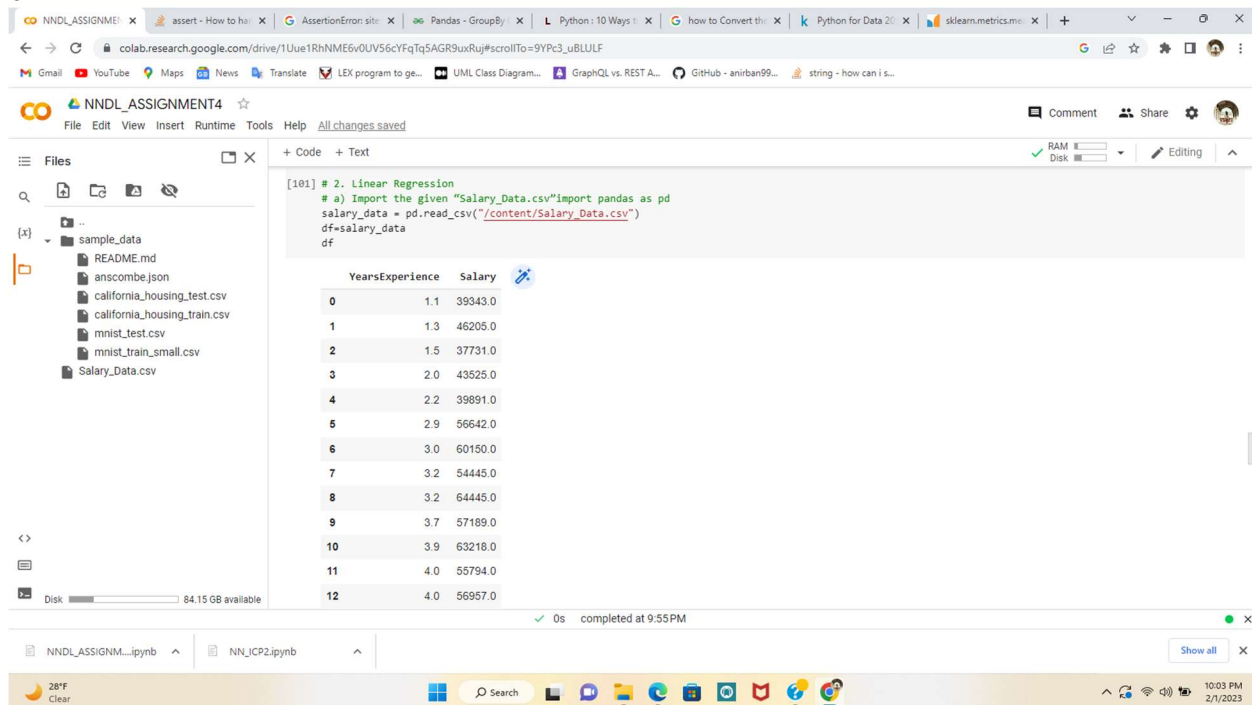
2. Linear Regression

a) Import the given "Salary_Data.csv" import pandas as pd

salary_data = pd.read_csv("/content/Salary_Data.csv")

df=salary_data

df



```

#b) Split the data in train_test partitions, such that 1/3 of the data is
reserved as test subset
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

# Split the data into features (predictors) and target variables
X = df.iloc[:, :-1]
y = df.iloc[:, -1]

# Split the data into training and testing subsets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=np.random.seed(123))

# c) Train and predict the model.

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Load the data

# Split the data into features (predictors) and target variables
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

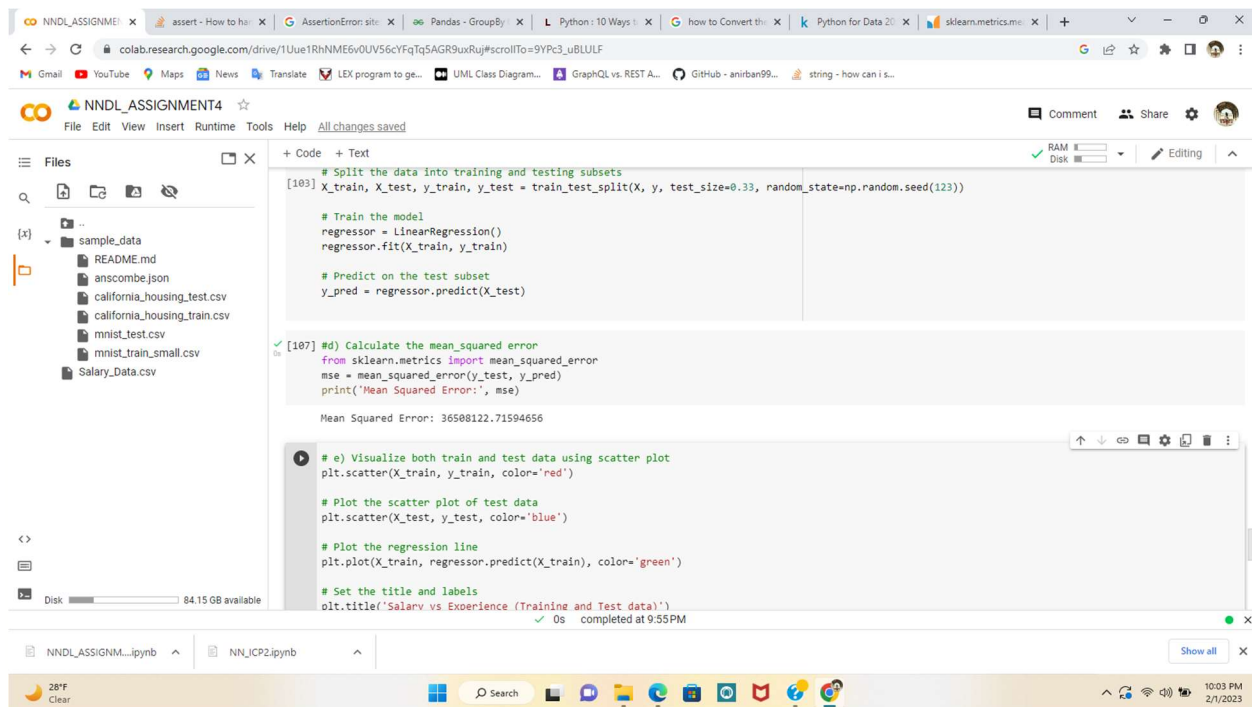
# Split the data into training and testing subsets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=np.random.seed(123))

# Train the model
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predict on the test subset
y_pred = regressor.predict(X_test)

#d) Calculate the mean_squared error
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
print('Mean Squared Error:', mse)

```



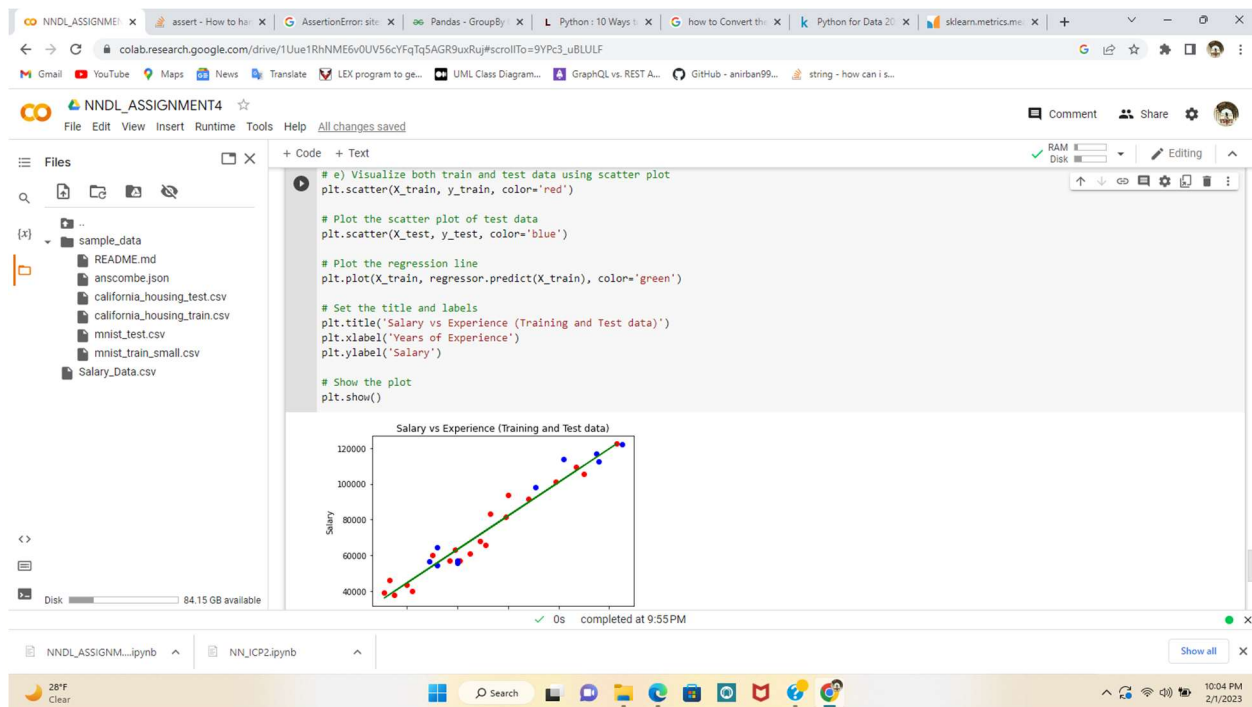
```
# e) Visualize both train and test data using scatter plot
plt.scatter(X_train, y_train, color='red')

# Plot the scatter plot of test data
plt.scatter(X_test, y_test, color='blue')

# Plot the regression line
plt.plot(X_train, regressor.predict(X_train), color='green')

# Set the title and labels
plt.title('Salary vs Experience (Training and Test data)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')

# Show the plot
plt.show()
```

REPO LINK: https://github.com/Goli18/NNDL_ASS4.git

VIDEO LINK: <https://www.veed.io/view/6689707c-8dd3-4833-b0a1-b274e45da358?source=compressor-sharing>