

**Name:** Muhammad Faique

**ID:** F2019065048

**Section:** W1

**Course:** Data Science Technologies

**Submitted To:** Sir, Mazhar Javed Awan

**Project Documentation**

**Topic: Pakistan Temperature**

**About Project:**

As we know that in the world of data science can be used to predict values and perform machine learning and deep learning algorithms. This project is done python programming language in Google Colab Jupiter Notebook. In my project, I have covered the area of Pakistan for Temperature or it is based on Pakistan temperature from the year 1901 to 2016. In this project, I have done the classification of data. And then first removing the missing values and any kind of noisy data. I have perform or applied numpy and pandas operations. Applied graphs and visualized the data through matplotlib and done plotting by using seaborn python libraries. I have applied machine learning models. By linear regression, multiple regression, decision tree classifier, random forest regressor and by using KNeighborsClassifier. I have built a confusion matrix and provided the heat map along with it for better a visualization. In the end, I have built classification reports.

**Implementation and working**

At first I have inserted important libraries that I have used to build the model and have done data science work.

import pandas as pd

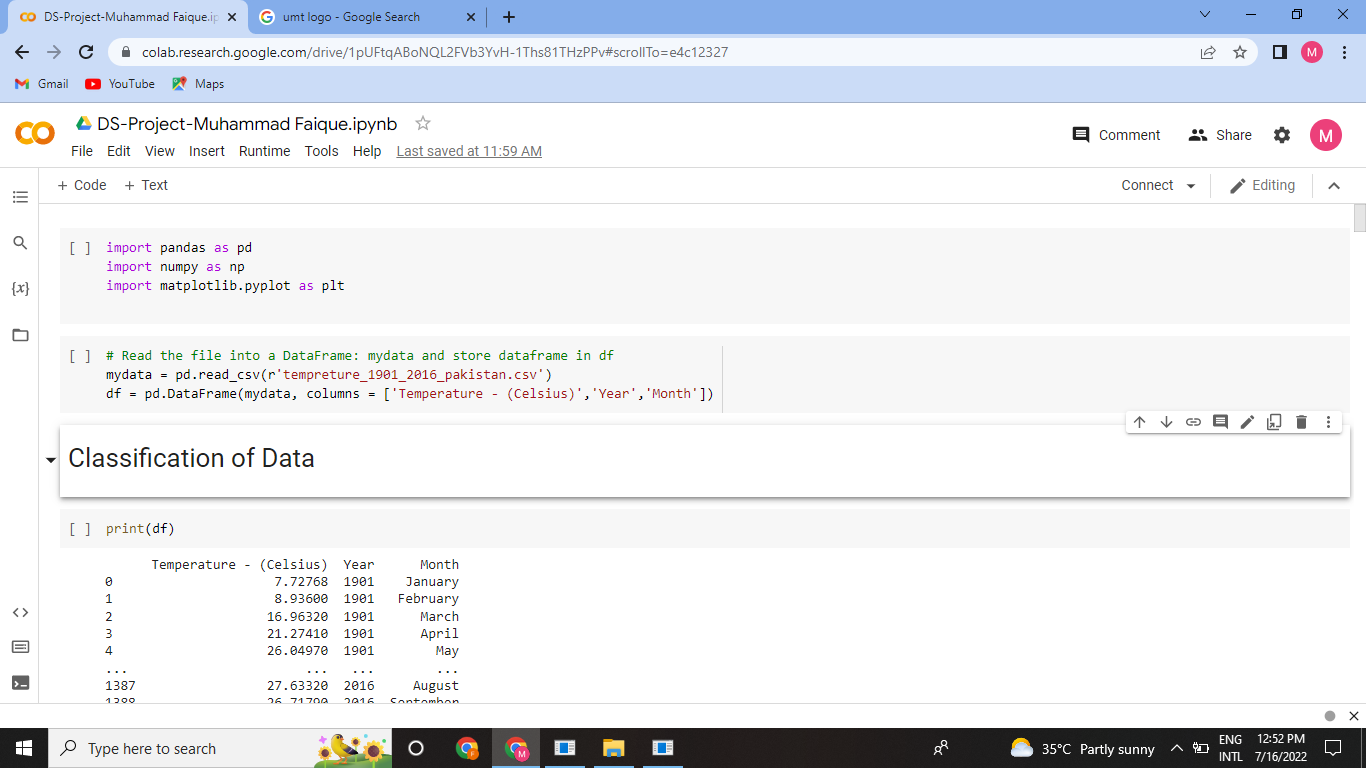
import numpy as np

import matplotlib.pyplot as plt

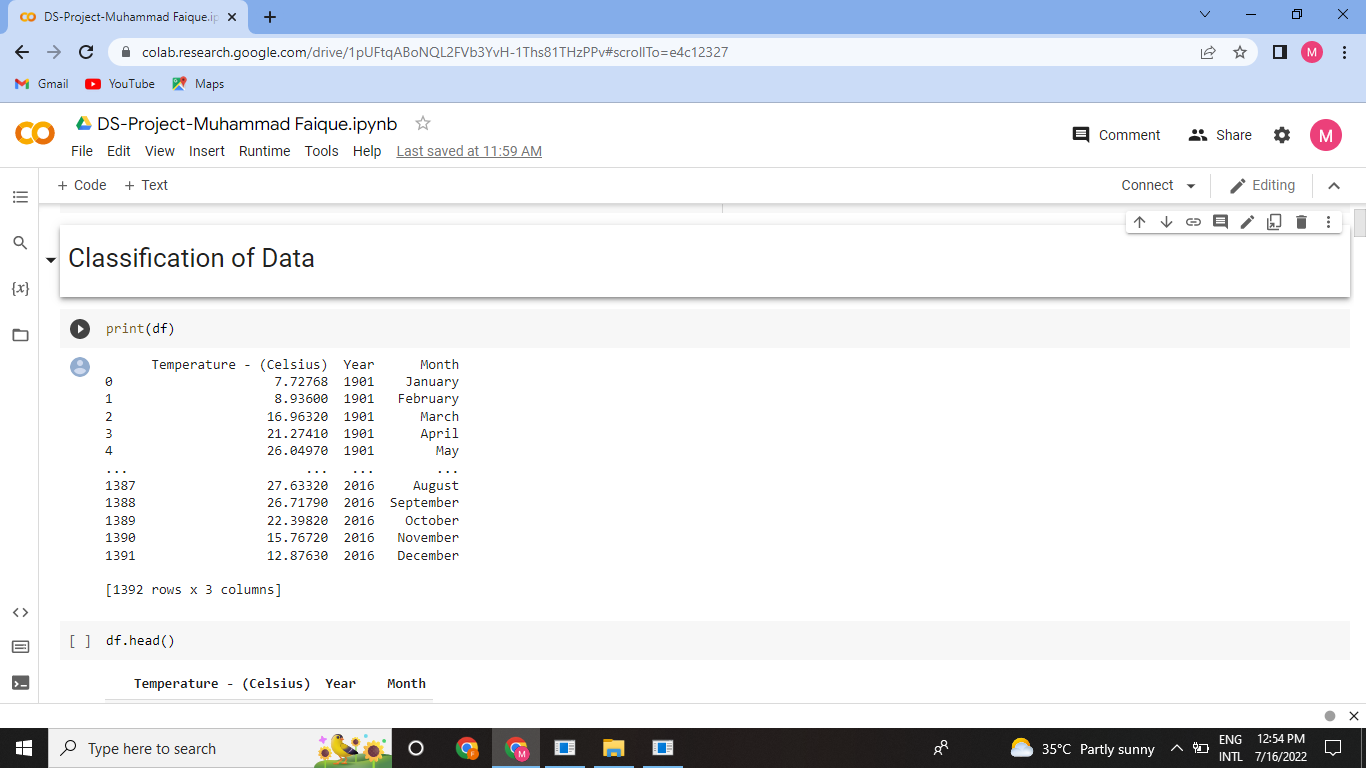
Then I am loading and reading the file of Pakistan temperature and creating dataframe.

mydata = pd.read\_csv(r'tempreture\_1901\_2016\_pakistan.csv')

df = pd.DataFrame(mydata, columns = ['Temperature - (Celsius)','Year','Month'])

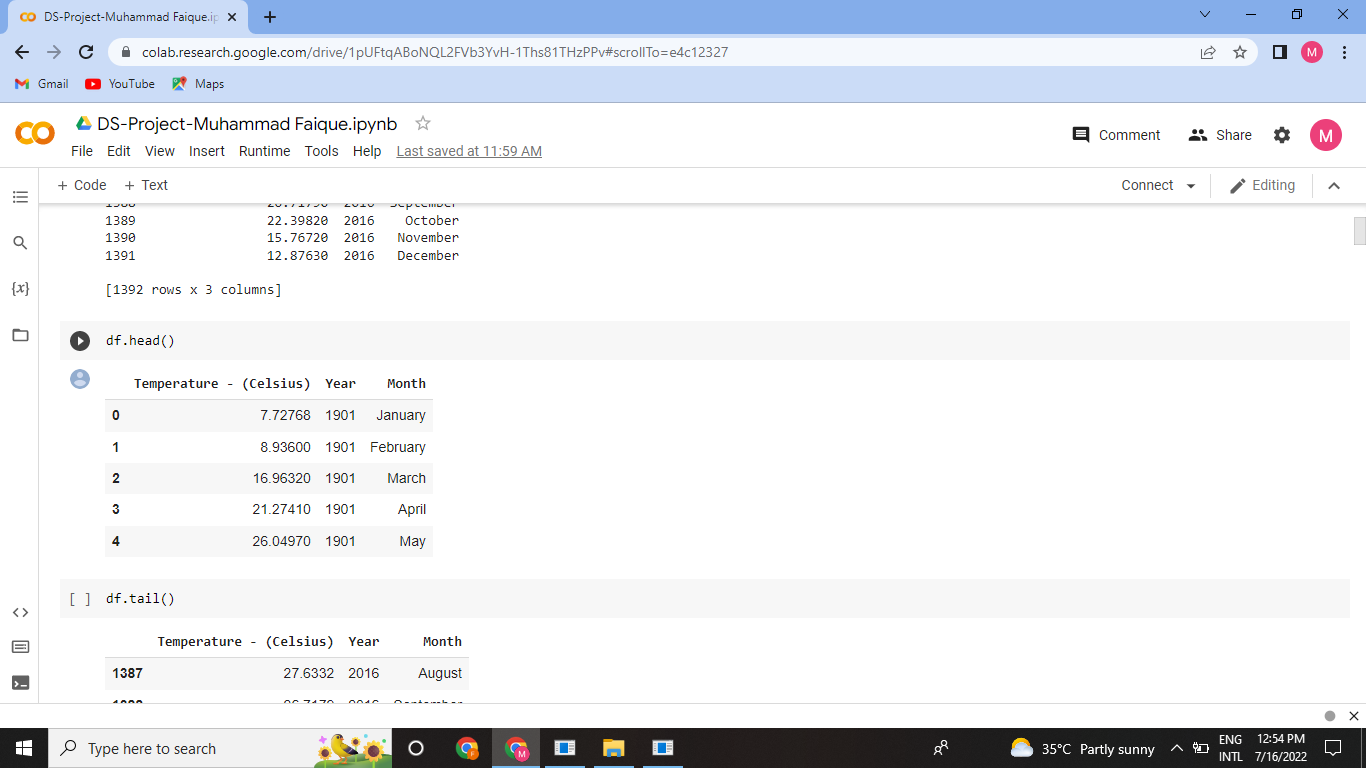


then printed and done data analysis work by checking first 5 rows and last 5 rows of Pakistan temperature dataset.

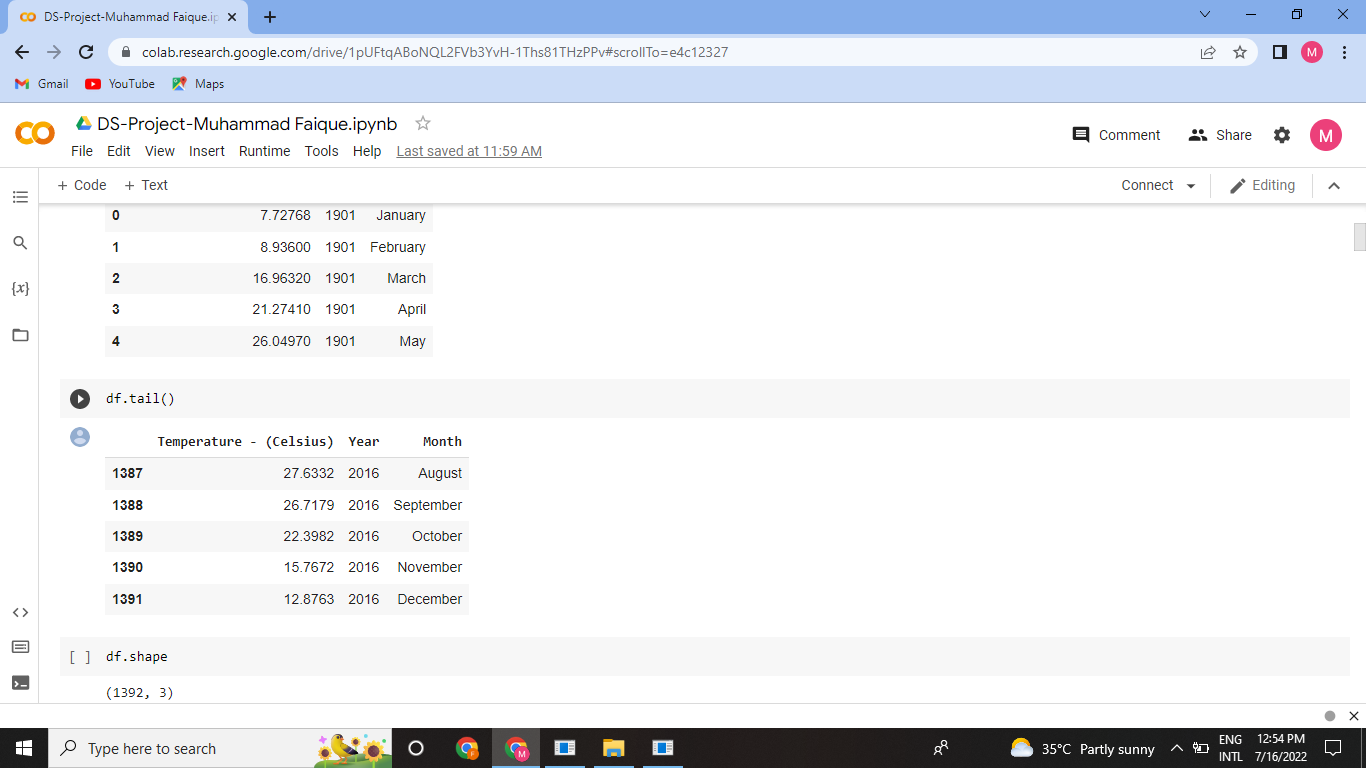


df.head()

Checking first five rows using head

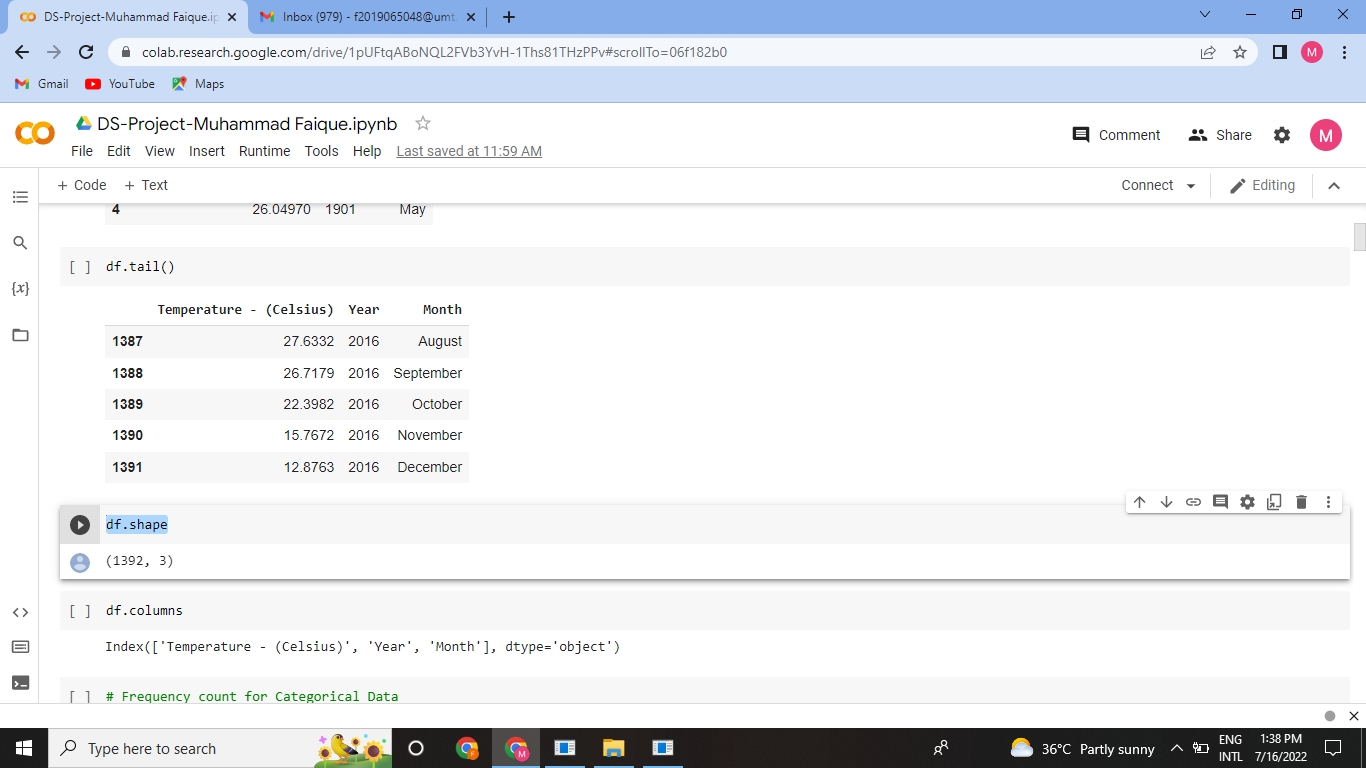


Checking the last five rows.

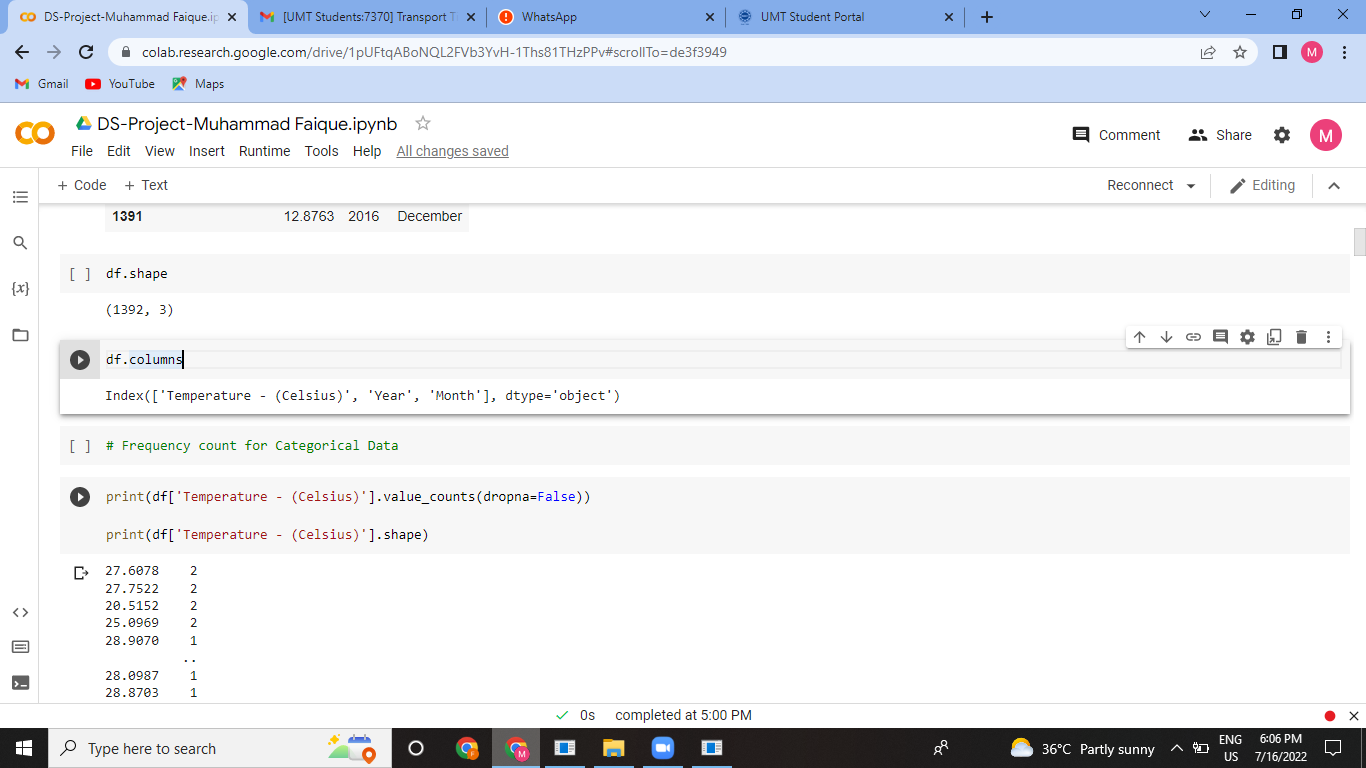


df.shape

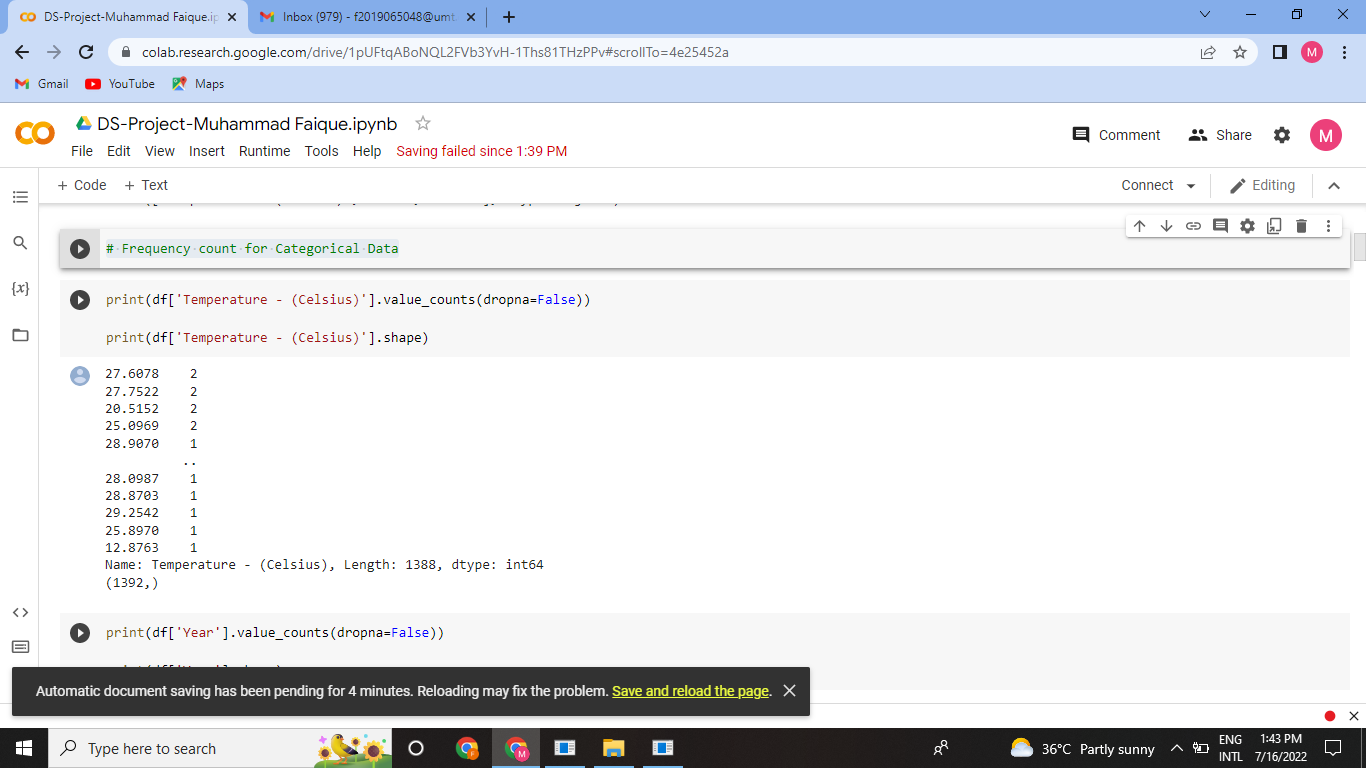
This shape command gives us the total number of rows and columns in our dataset.



df.columns

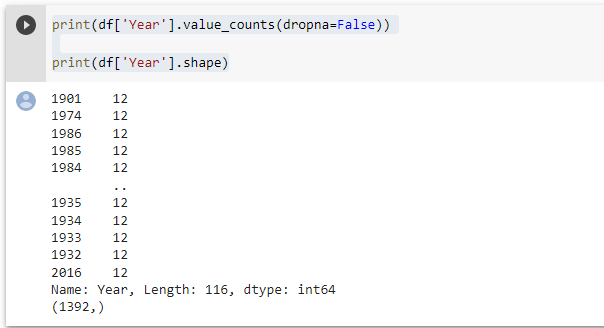


# Frequency count for Categorical Data



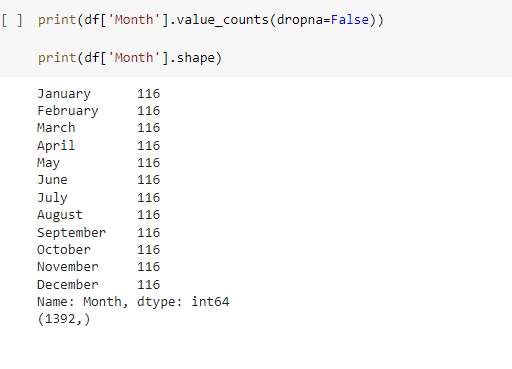
print(df['Year'].value\_counts(dropna=False))

print(df['Year'].shape)



print(df['Month'].value\_counts(dropna=False))

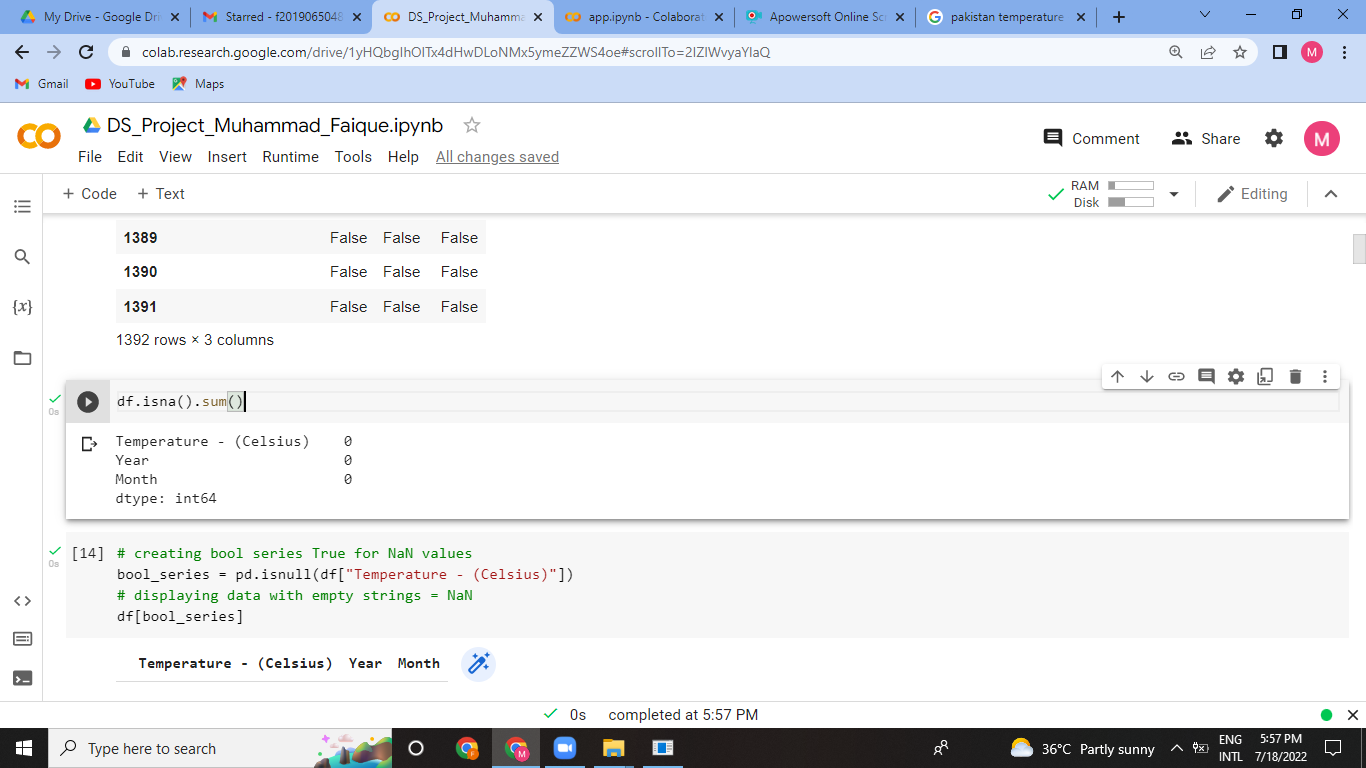
print(df['Month'].shape)



Analyzing missing values and noisy data

df.isna()



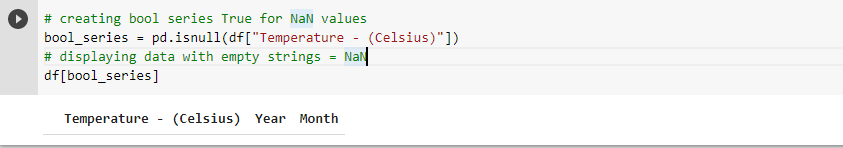


# creating bool series True for NaN values

bool\_series = pd.isnull(df["Temperature - (Celsius)"])

# displaying data with empty strings = NaN

df[bool\_series]



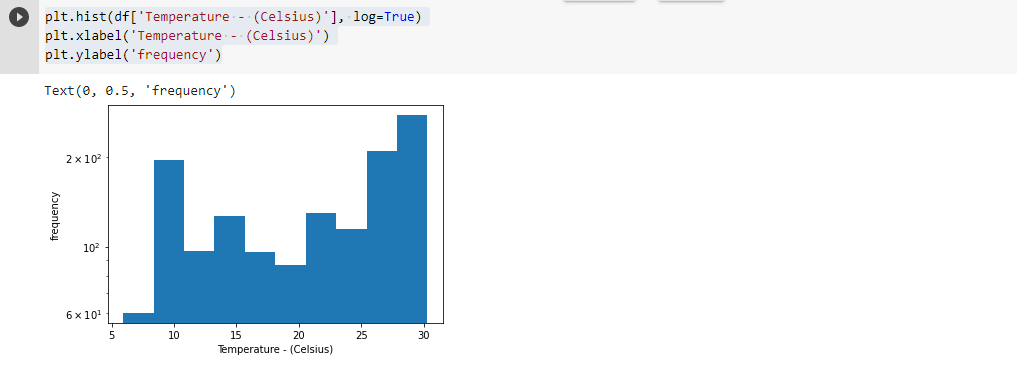
#As from above commands it can be seen that there are not any empty cells so no need to do interpolaion

# Create the Histogram plot

plt.hist(df['Temperature - (Celsius)'], log=True)

plt.xlabel('Temperature - (Celsius)')

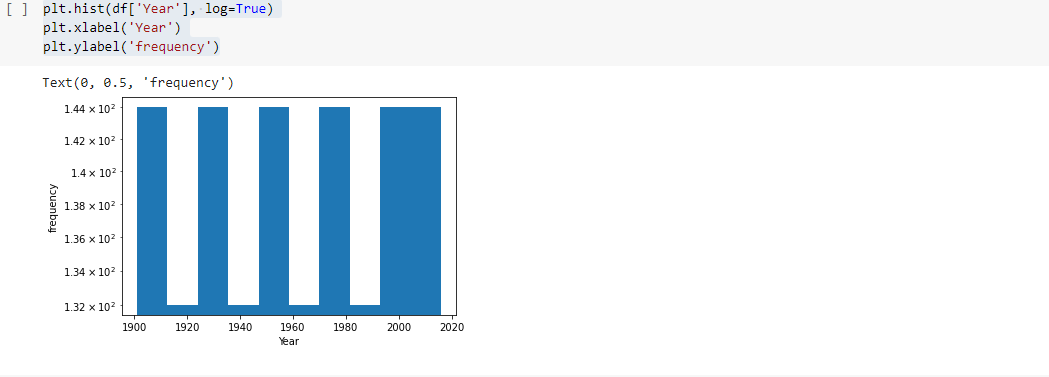
plt.ylabel('frequency')



plt.hist(df['Year'], log=True)

plt.xlabel('Year')

plt.ylabel('frequency')

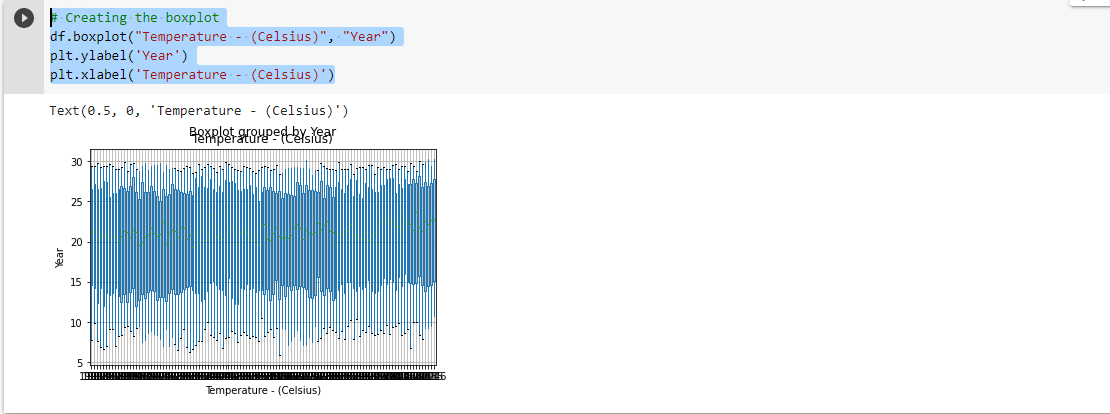


# Creating the boxplot

df.boxplot("Temperature - (Celsius)", "Year")

plt.ylabel('Year')

plt.xlabel('Temperature - (Celsius)')

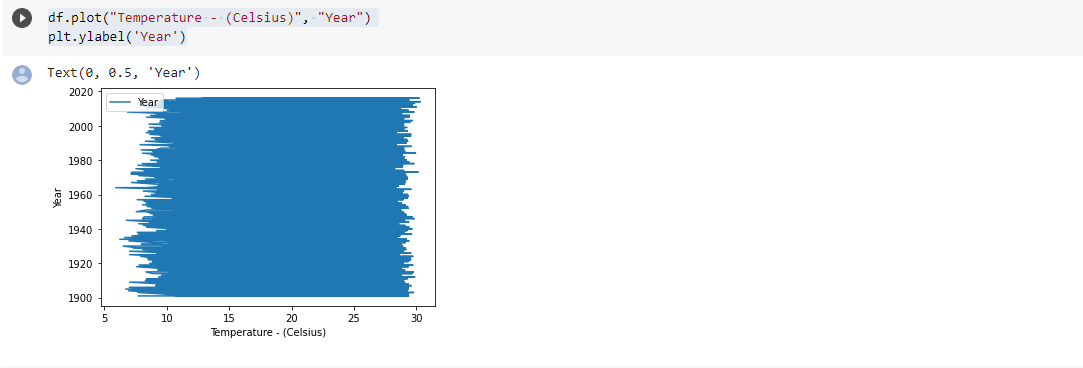


# Display the plot

plt.show()

df.plot("Temperature - (Celsius)", "Year")

plt.ylabel('Year')

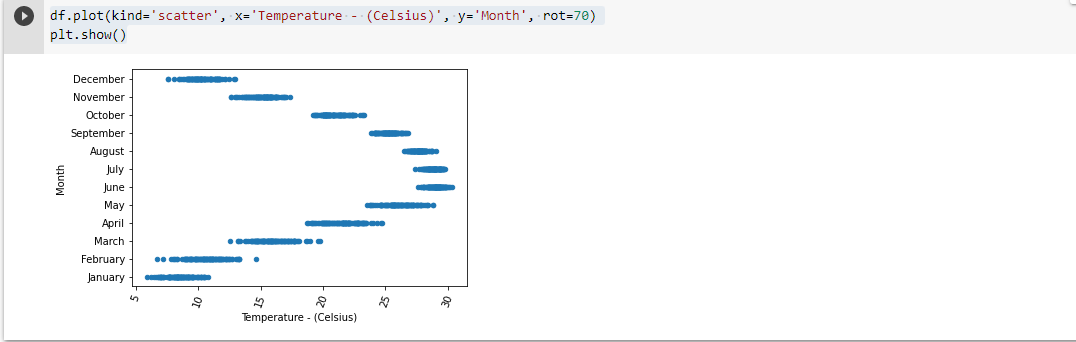


#As can be seen, the graph is difficult to understand. There are various lines making it difficult to see individual trends

#So we can go for Scatter Plot

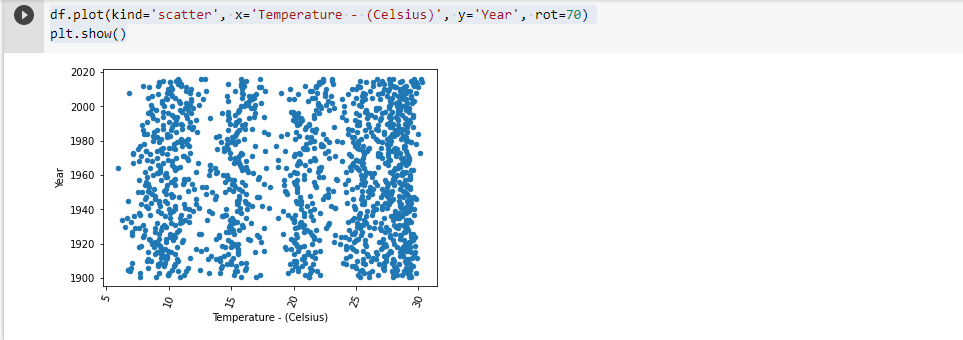
df.plot(kind='scatter', x='Temperature - (Celsius)', y='Month', rot=70)

plt.show()



df.plot(kind='scatter', x='Temperature - (Celsius)', y='Year', rot=70)

plt.show()



# Now we can apply different types of regressions to find the co-rrealtion

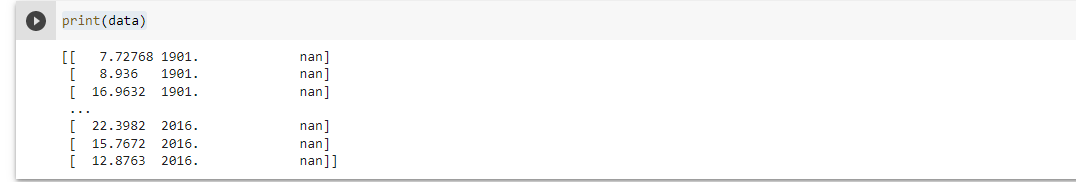
# Numpy operations

#basic examination using numpy

#Make sure to import numpy as np and loading the data

data = np.genfromtxt('tempreture\_1901\_2016\_pakistan.csv',delimiter=',', skip\_header= True,)

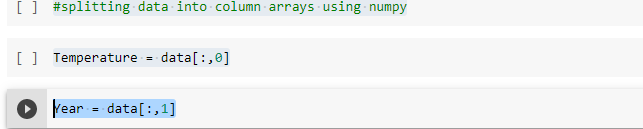
print(data)



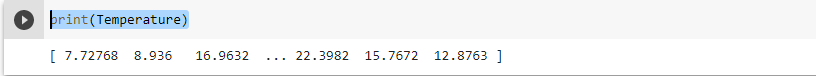
#splitting data into column arrays using numpy

Temperature = data[:,0]

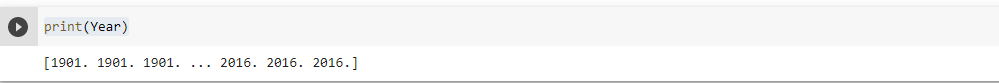
Year = data[:,1]



print(Temperature)

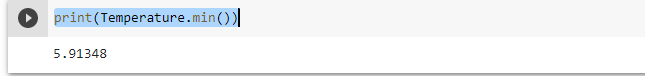


print(Year)

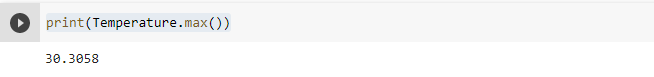


#some basic observations

print(Temperature.min())



print(Temperature.max())

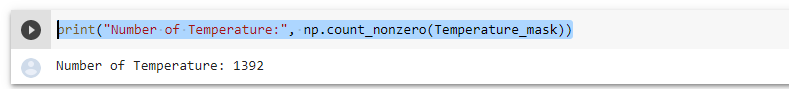


This gives us the minimum and maximum value of temperature.

#check for missing or bad data using numpy

Temperature\_mask = np.isfinite(Temperature)

print("Number of Temperature:", np.count\_nonzero(Temperature\_mask))



missing\_Temperature\_mask = ~np.isfinite(Temperature)

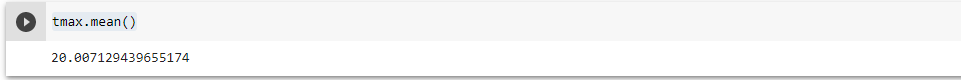
print("Number of missing Temperatures:", np.count\_nonzero(missing\_Temperature\_mask))



#calculating average Annual temperature

tmax = Temperature[(Year >= 1901) & (Year <= 2016)]

tmax.mean()

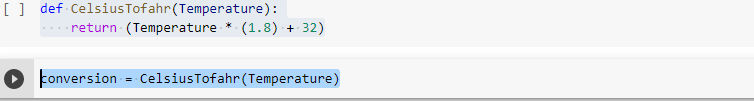


#conversion using numpy

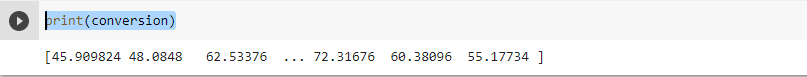
def CelsiusTofahr(Temperature):

    return (Temperature \* (1.8) + 32)

conversion = CelsiusTofahr(Temperature)



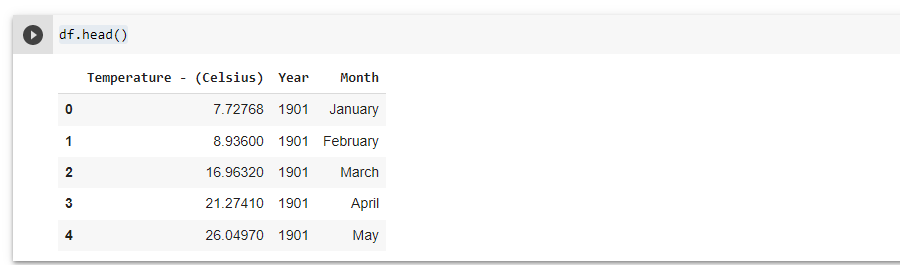
print(conversion)



Here, I am converting the temperature in Celsius to temperature in Fahrenheit using the formula “F = (Value of Temperature in Celsius \* 1.8) +32” for conversion using function (‘ def ’) and numpy operations.

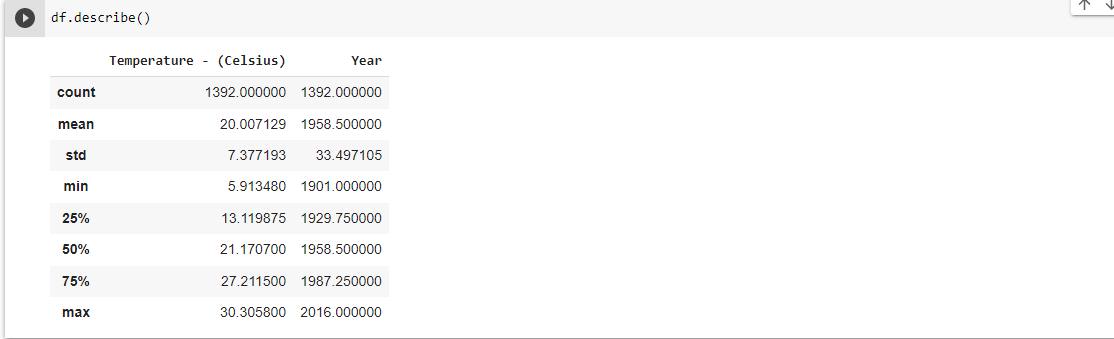
**Pandas operations**

df.head()



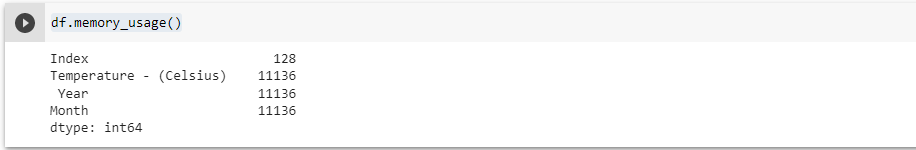
It Gives us overall Summary of our dataset Pakistan Temperature.

df.describe()



It provides us memory usage with respect to ram.

df.memory\_usage()



It gives the frequency count of values of certain specific columns. Or how many times a value is repeated in dataset columns.

df.value\_counts()



df.drop\_duplicates()

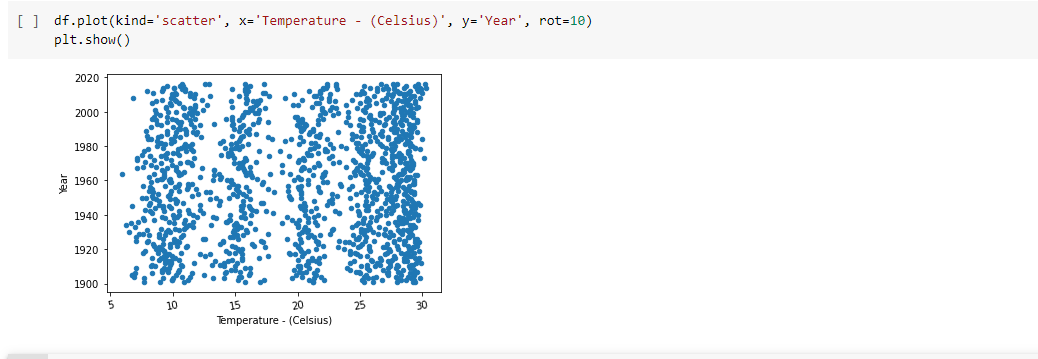


# plotting through matplotlib

import matplotlib.pyplot as plt

df.plot(kind='scatter', x='Temperature - (Celsius)', y='Year', rot=10)

plt.show()



x = df['Temperature - (Celsius)']

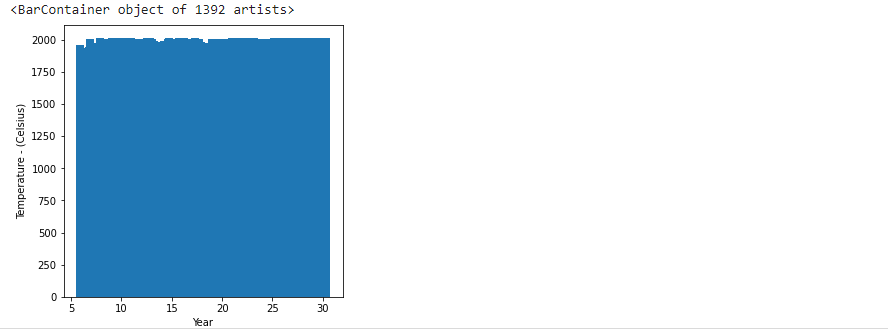
y = df['Year']

fig, ax = plt.subplots(figsize = (5,5))

plt.ylabel('Temperature - (Celsius)')

plt.xlabel('Year')

plt.bar(x,y)

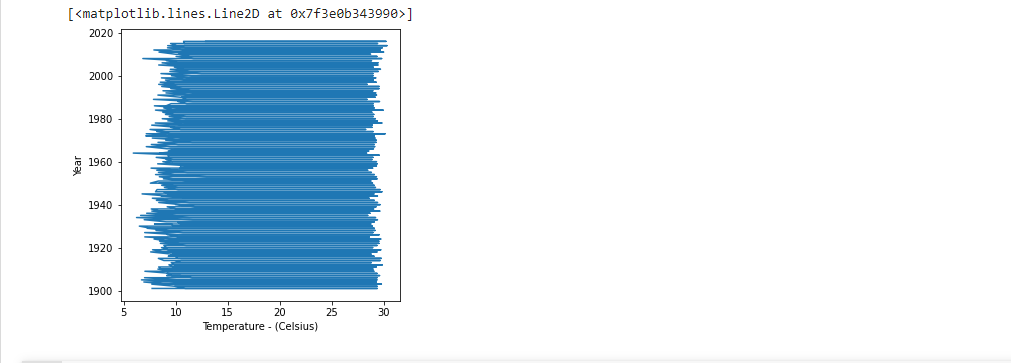


fig, ax = plt.subplots(figsize = (5,5))

plt.xlabel('Temperature - (Celsius)')

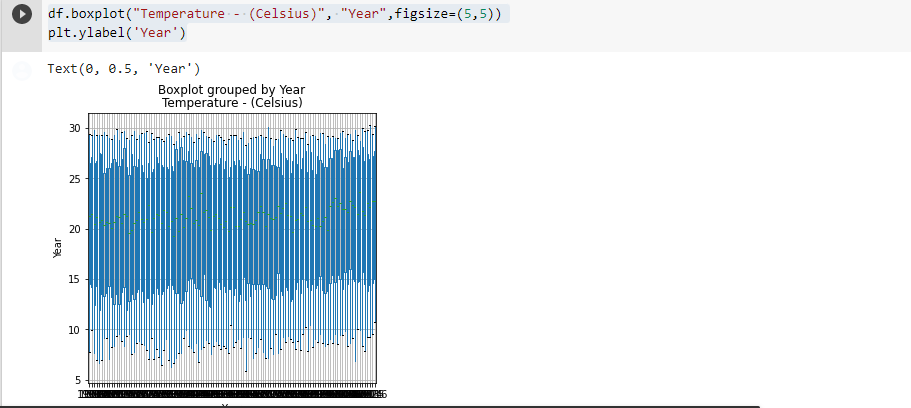
plt.ylabel('Year')

ax.plot(x,y)



df.boxplot("Temperature - (Celsius)", "Year",figsize=(5,5))

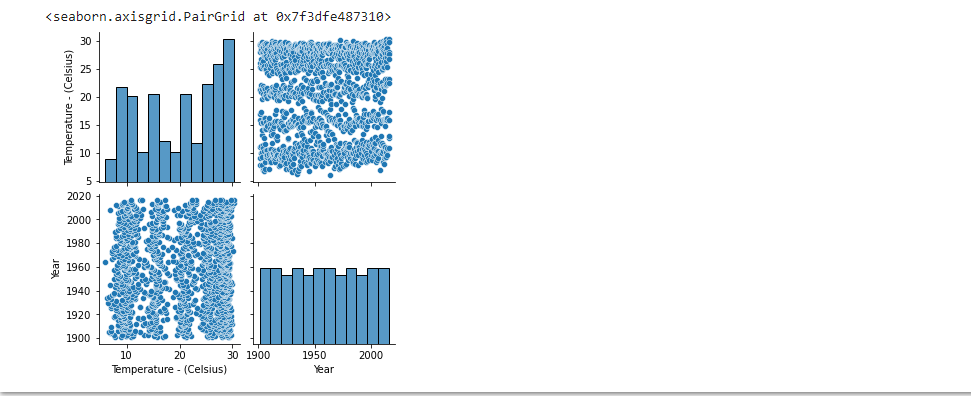
plt.ylabel('Year')



# plotting through seaborn

import seaborn as sns

sns.pairplot(df)



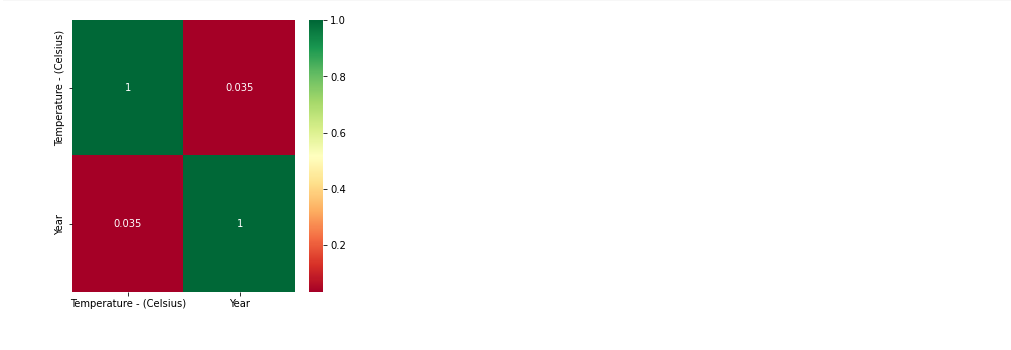
# plot heat map to get correlations of each features in dataset

corrmat = df.corr()

top\_corr\_features = corrmat.index

plt.figure(figsize=(5,5))

g=sns.heatmap(df.corr(),annot=True,cmap="RdYlGn")



# Machine learning models:

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

#linear regression

# 

# Print the slope and intercept

print(a,b)

# 

# Generate theoretical x and y data: x\_theor, y\_theor

# 

# Plot the Anscombe data and theoretical line

\_ = plt.plot(x, y, marker='.', linestyle='none')

\_ = plt.plot(x\_theor, y\_theor)

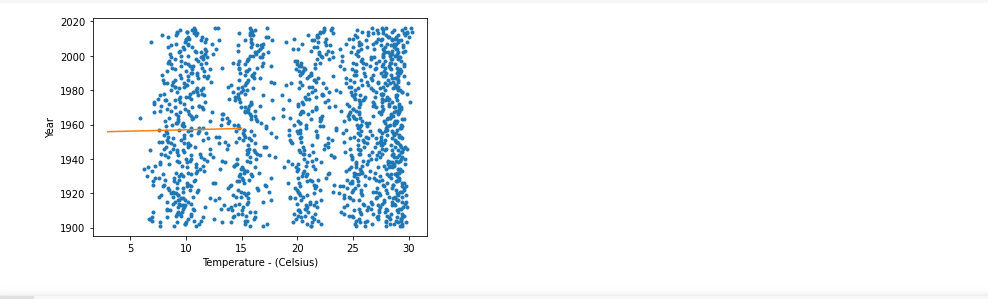
# Label the axes

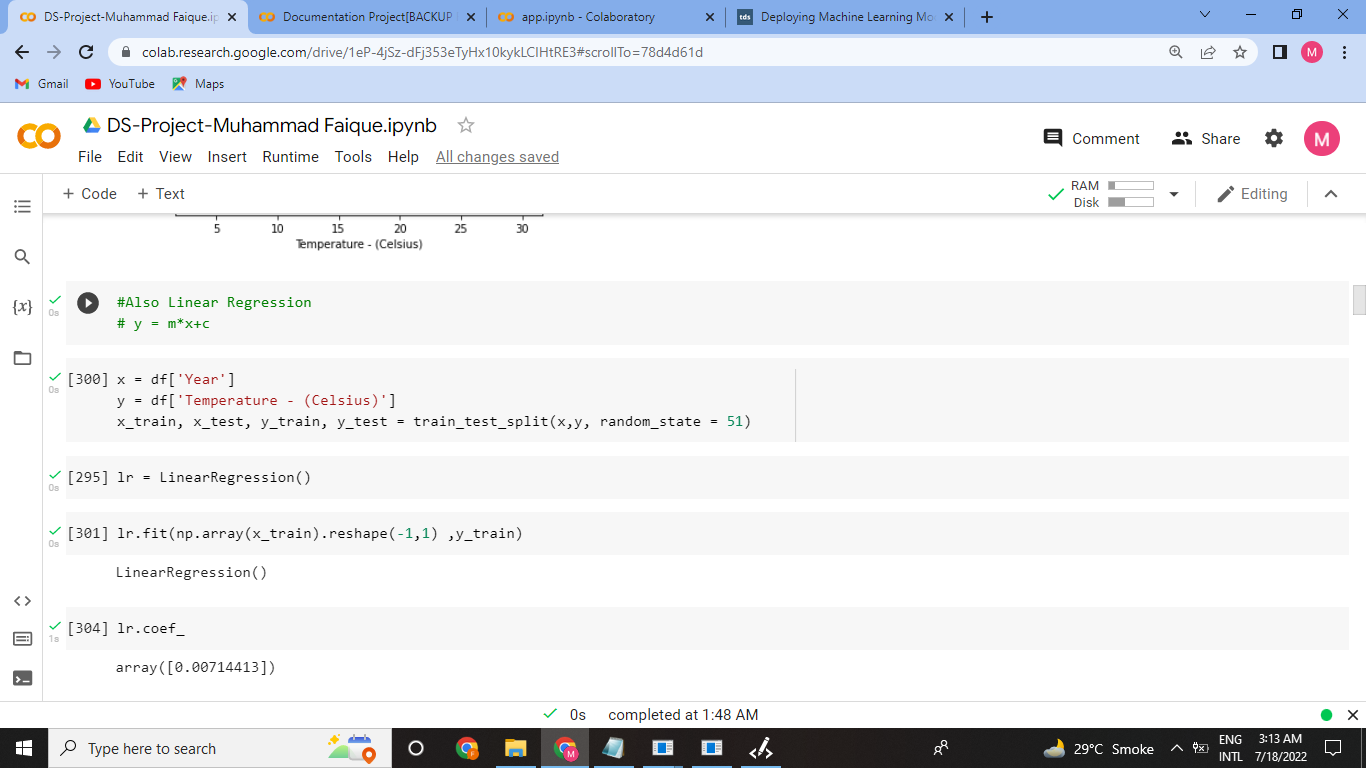
plt.xlabel('Temperature - (Celsius)')

plt.ylabel('Year')

# Show the plot

plt.show()

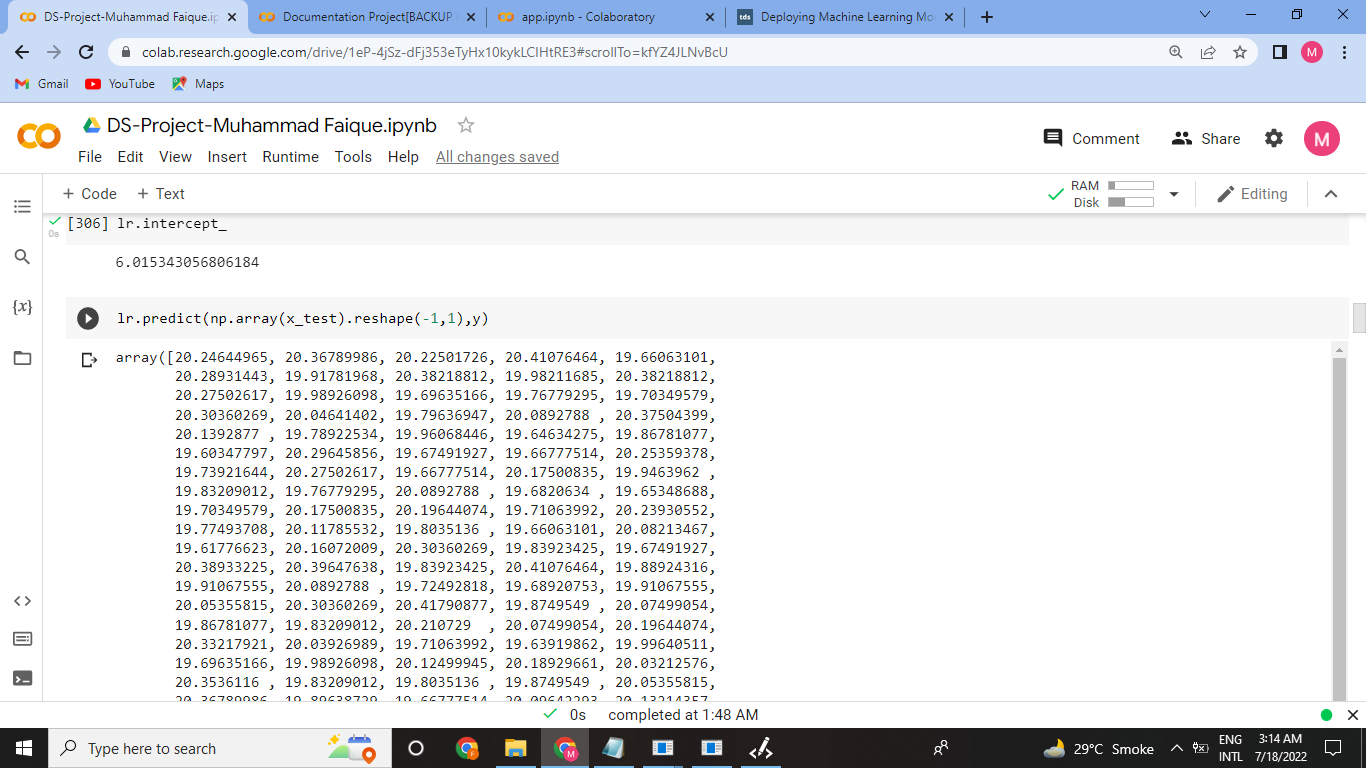




I am using linear regression model here. In which, we train data of year and temperature (in Celsius) and we predicate the temperature in Celsius based on the provided year by the user. Since, we cannot covert Month values in float otherwise it will give an error that it cannot covert string into float. So, first we have must import the library for linear regression and for training and test data.

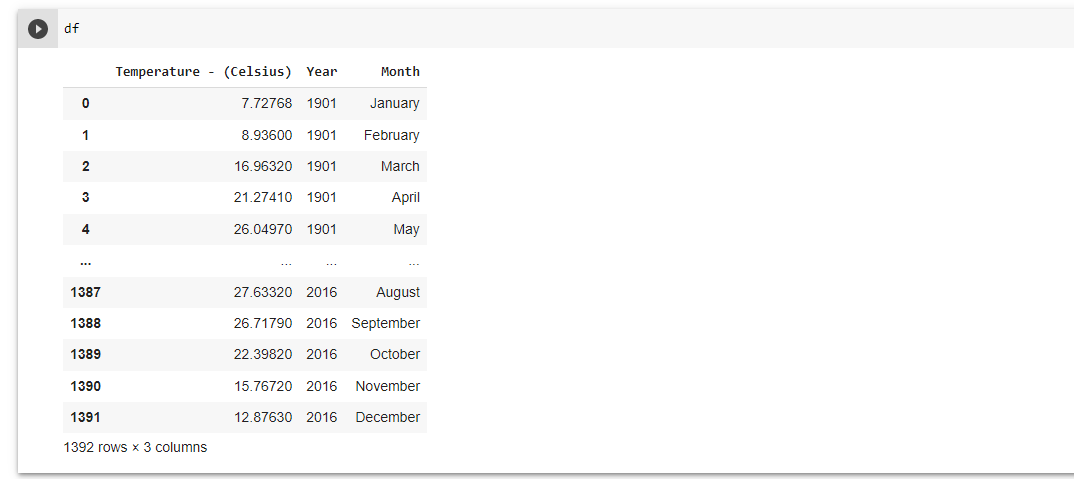
from sklearn.model\_selection import train\_test\_split

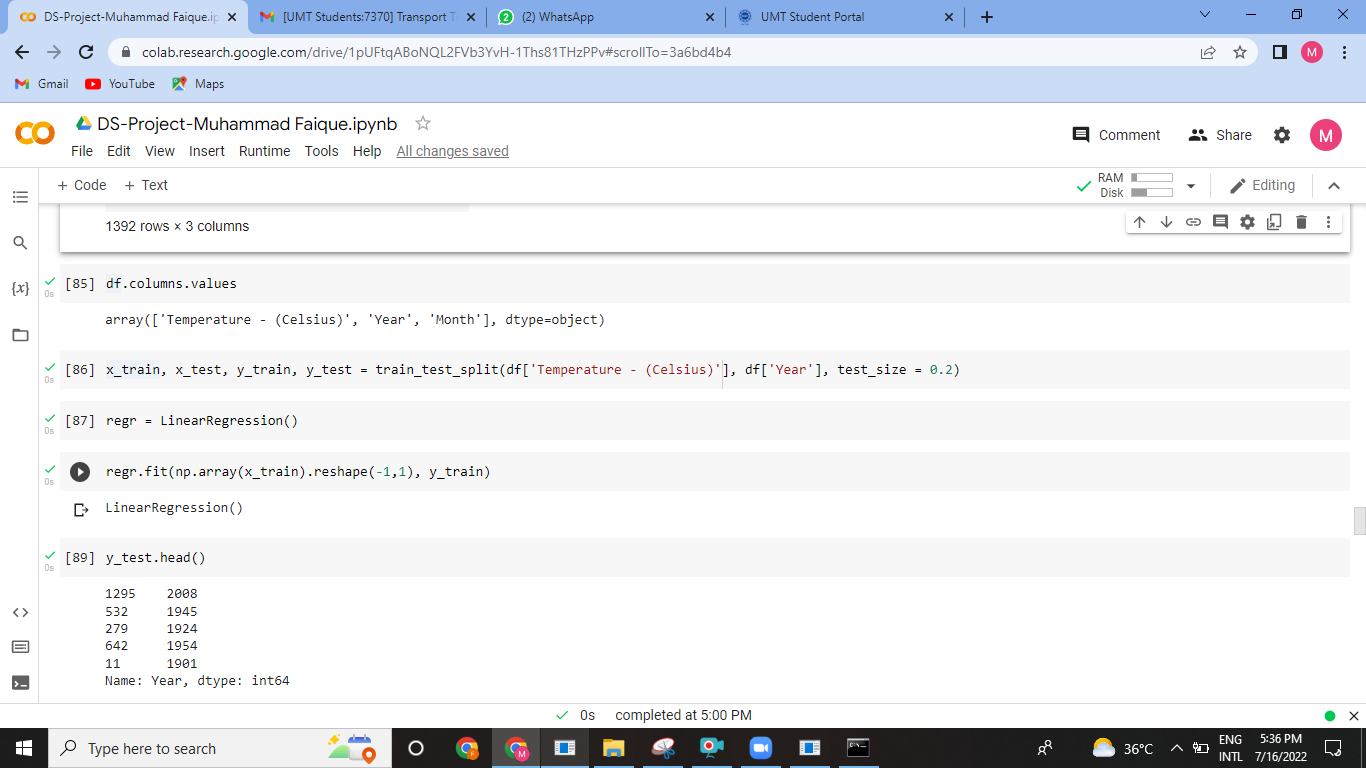
from sklearn.linear\_model import LinearRegression

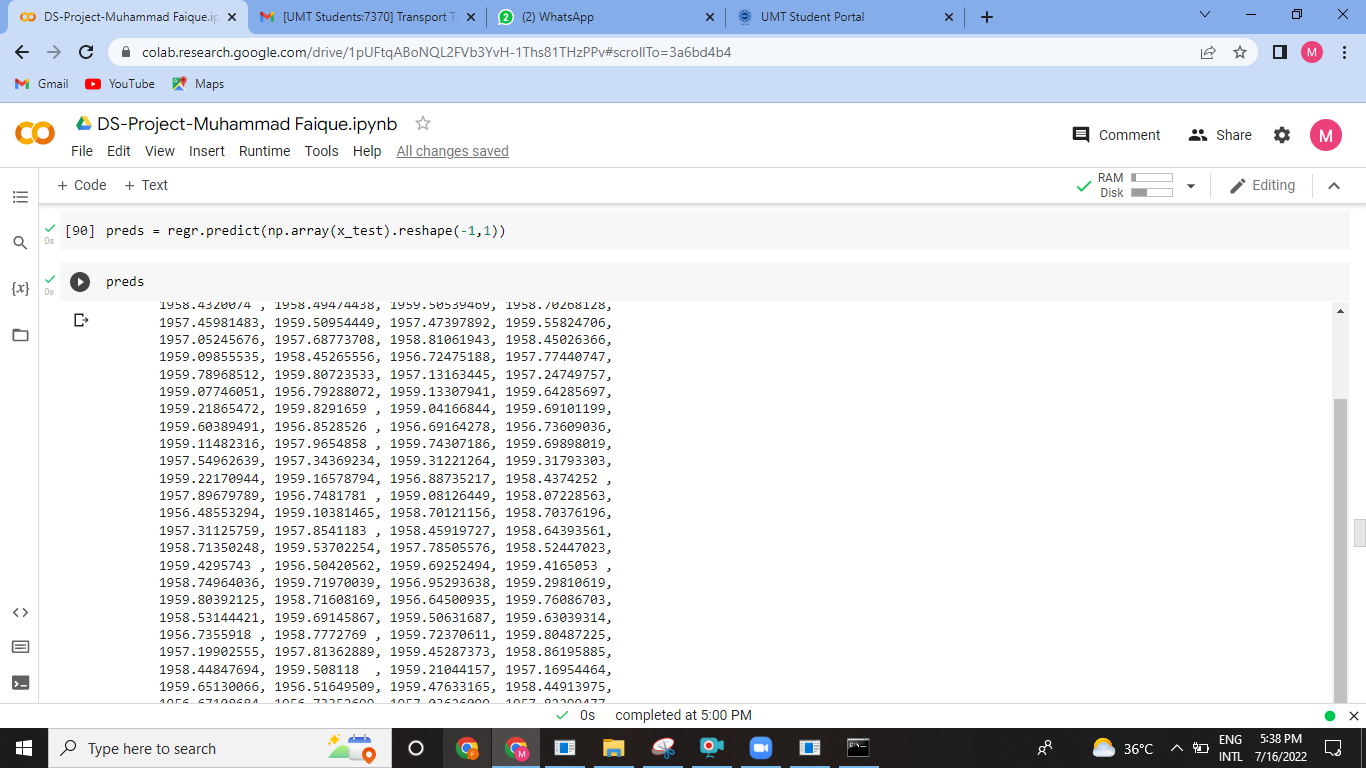


#Multiple regression

df

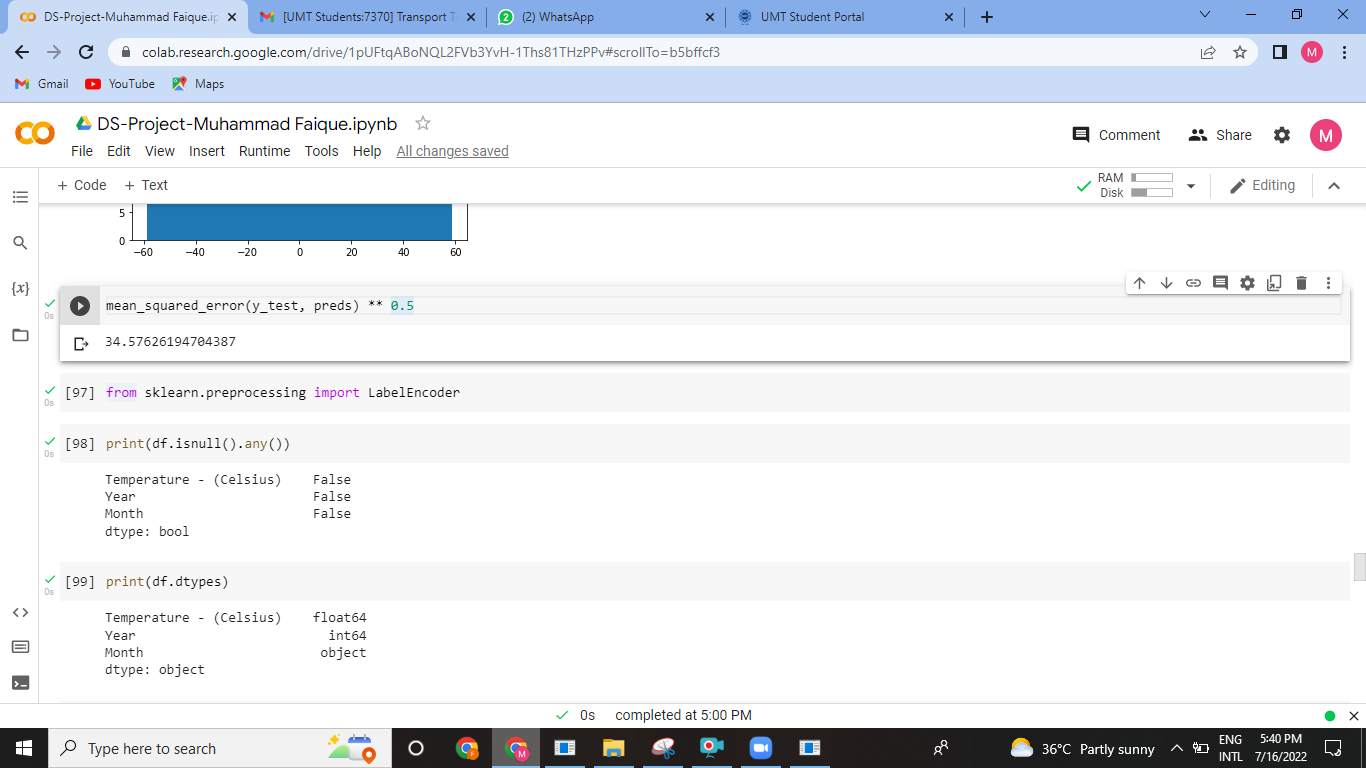


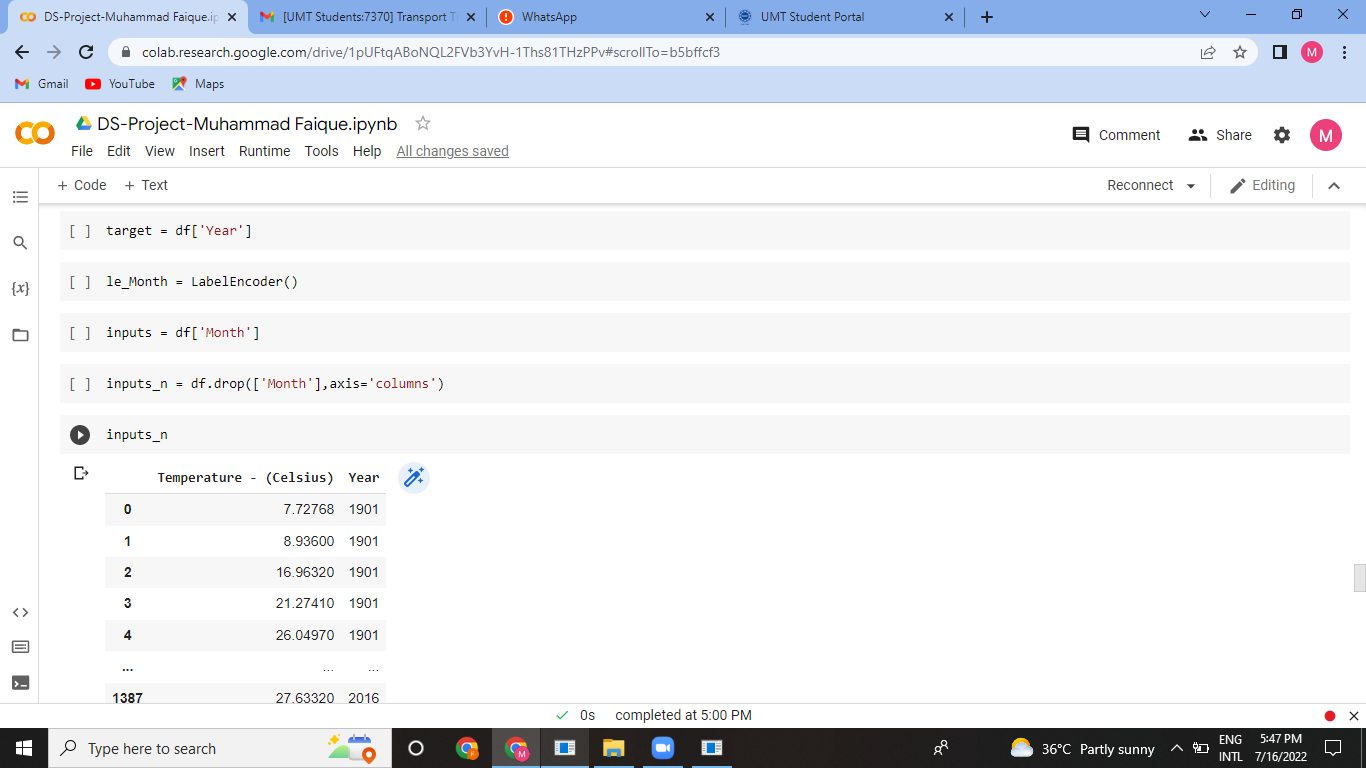


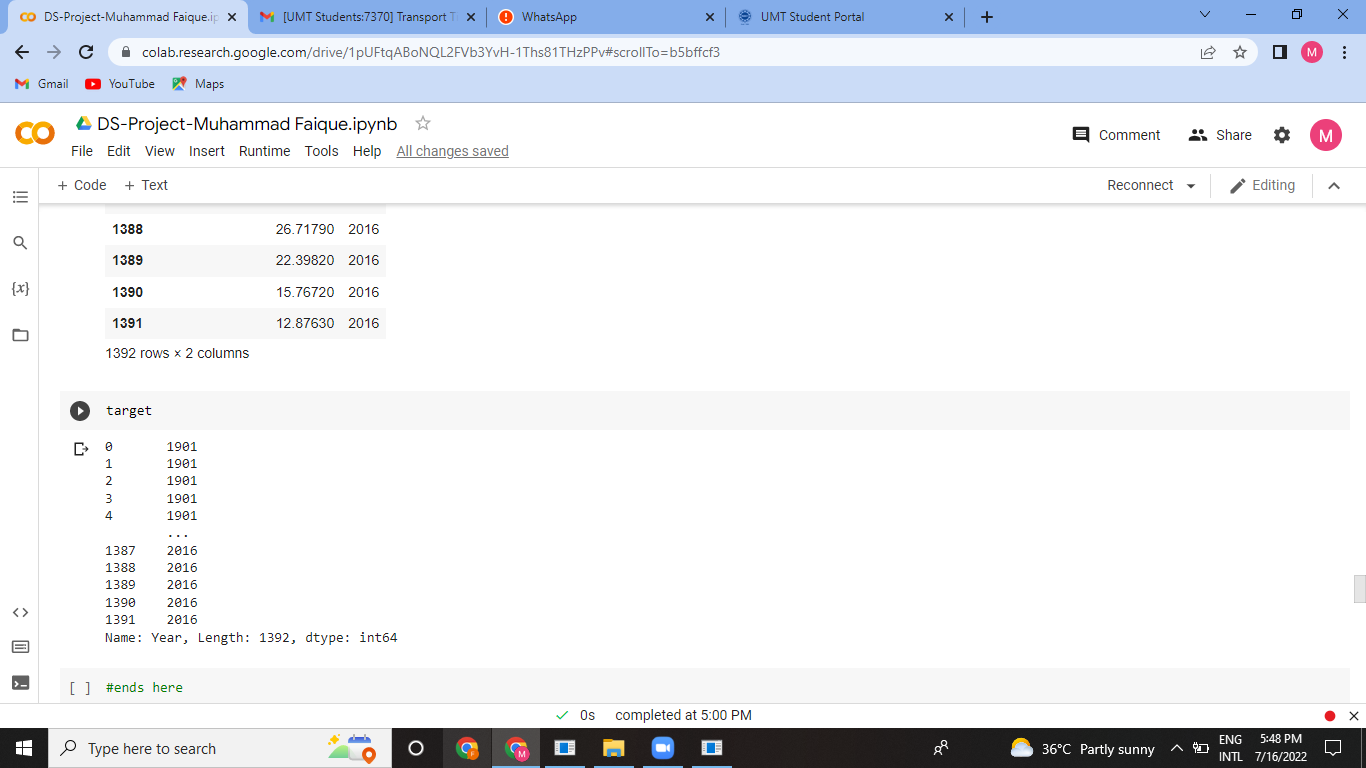


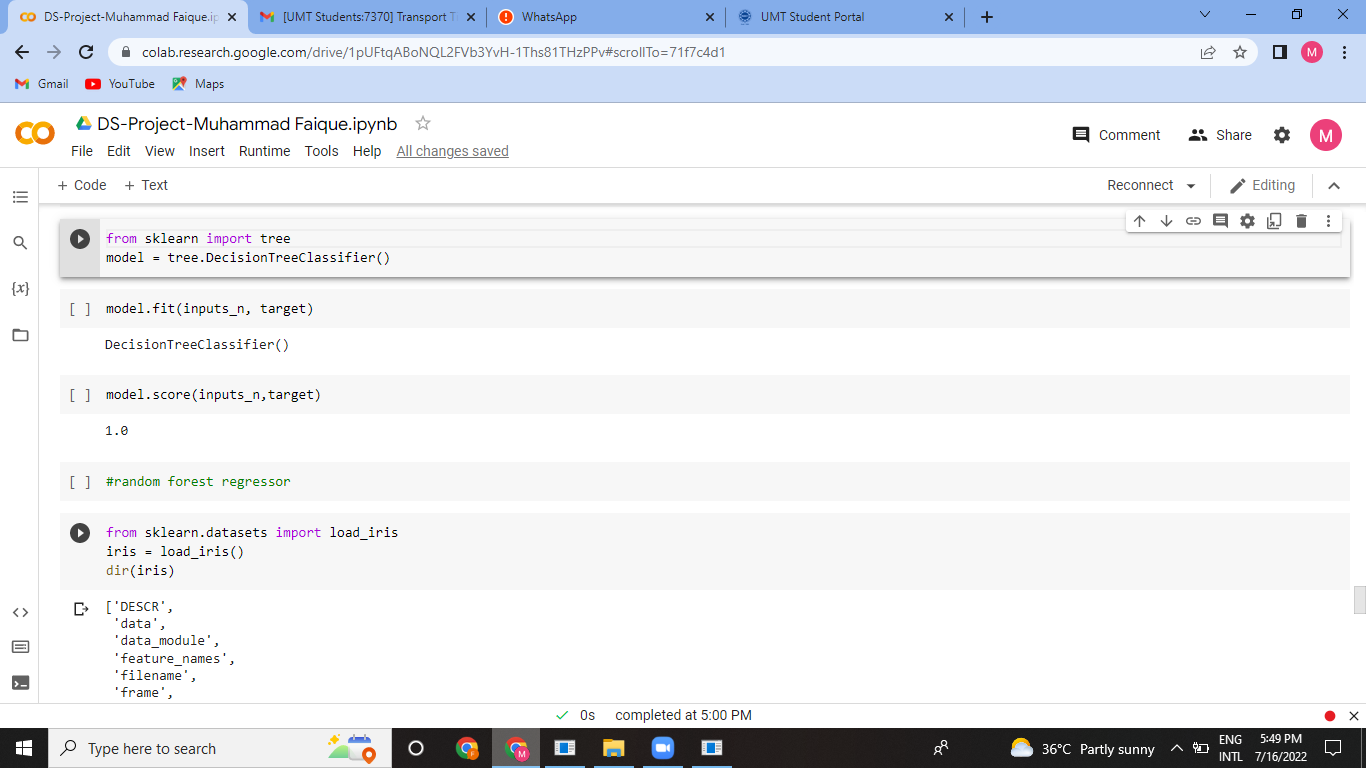
**Plotting histogram**

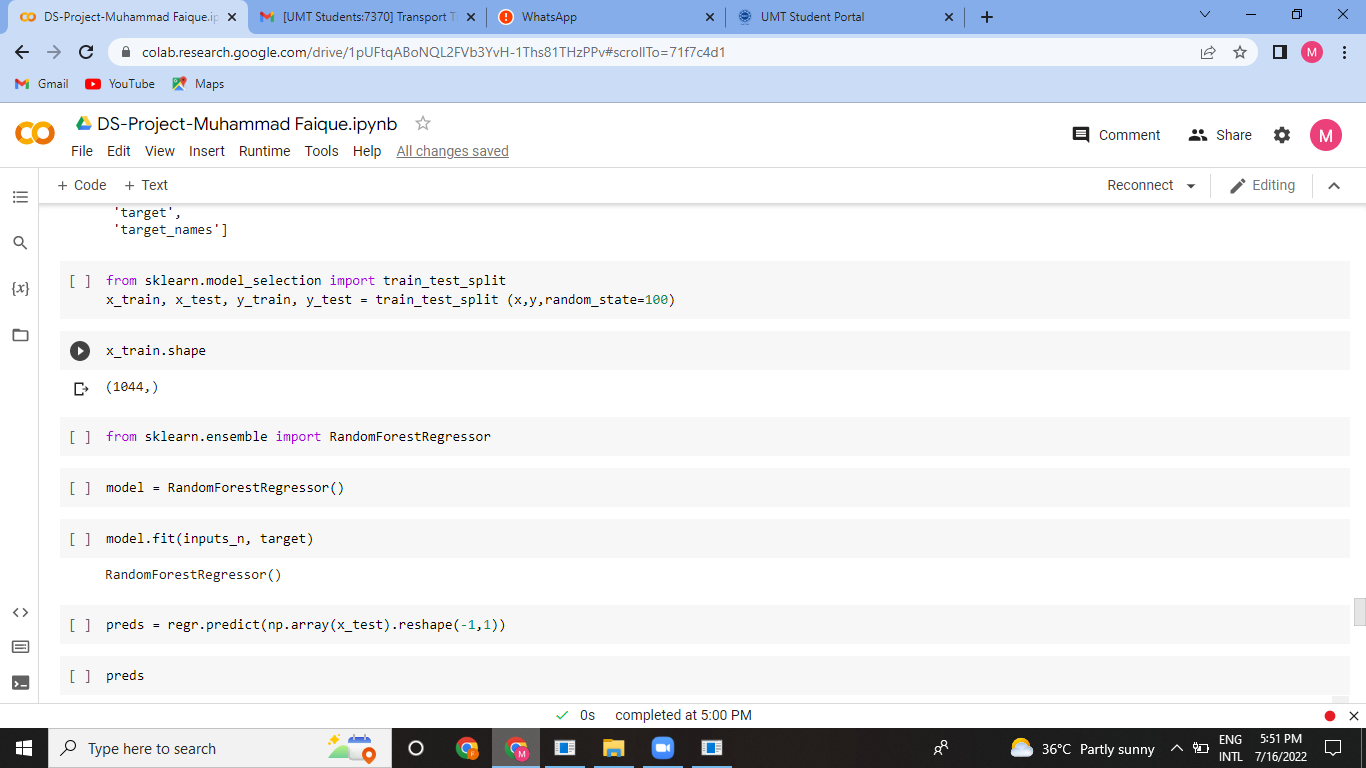


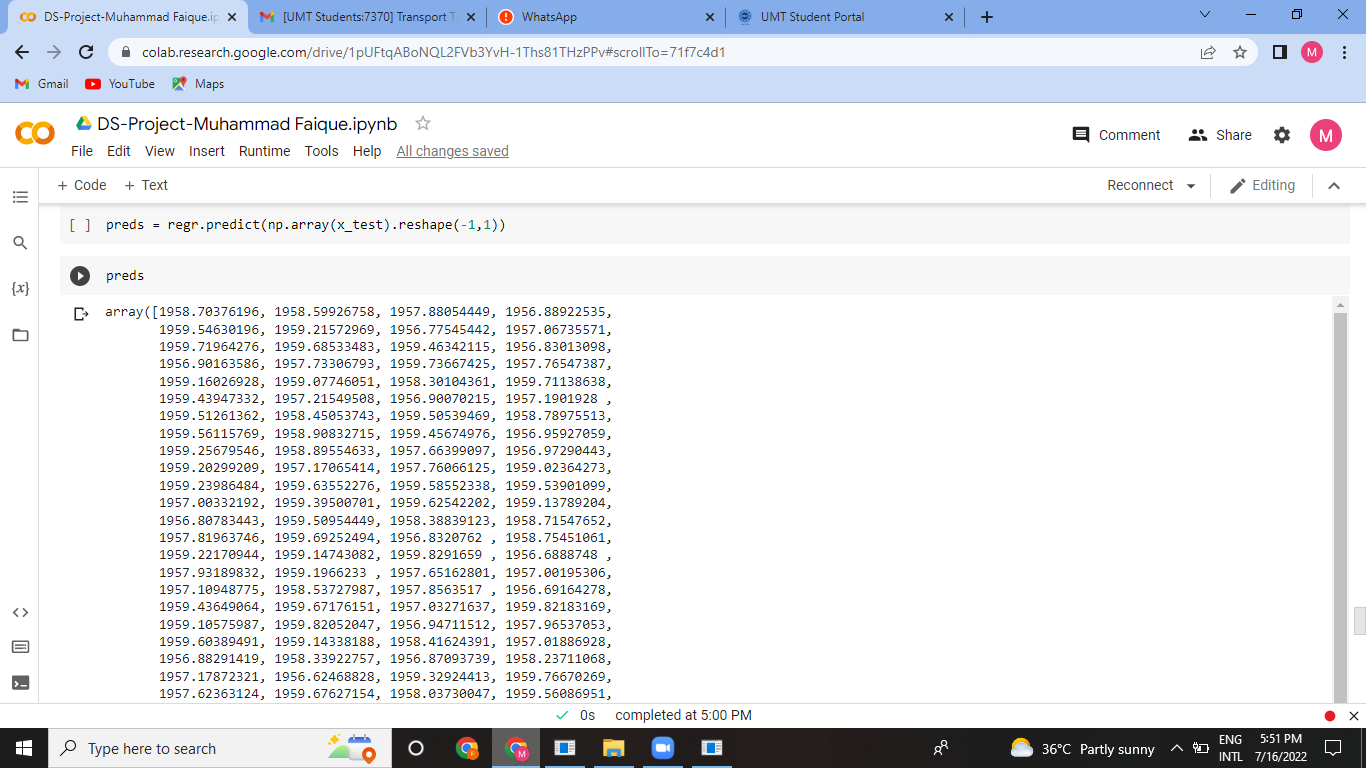


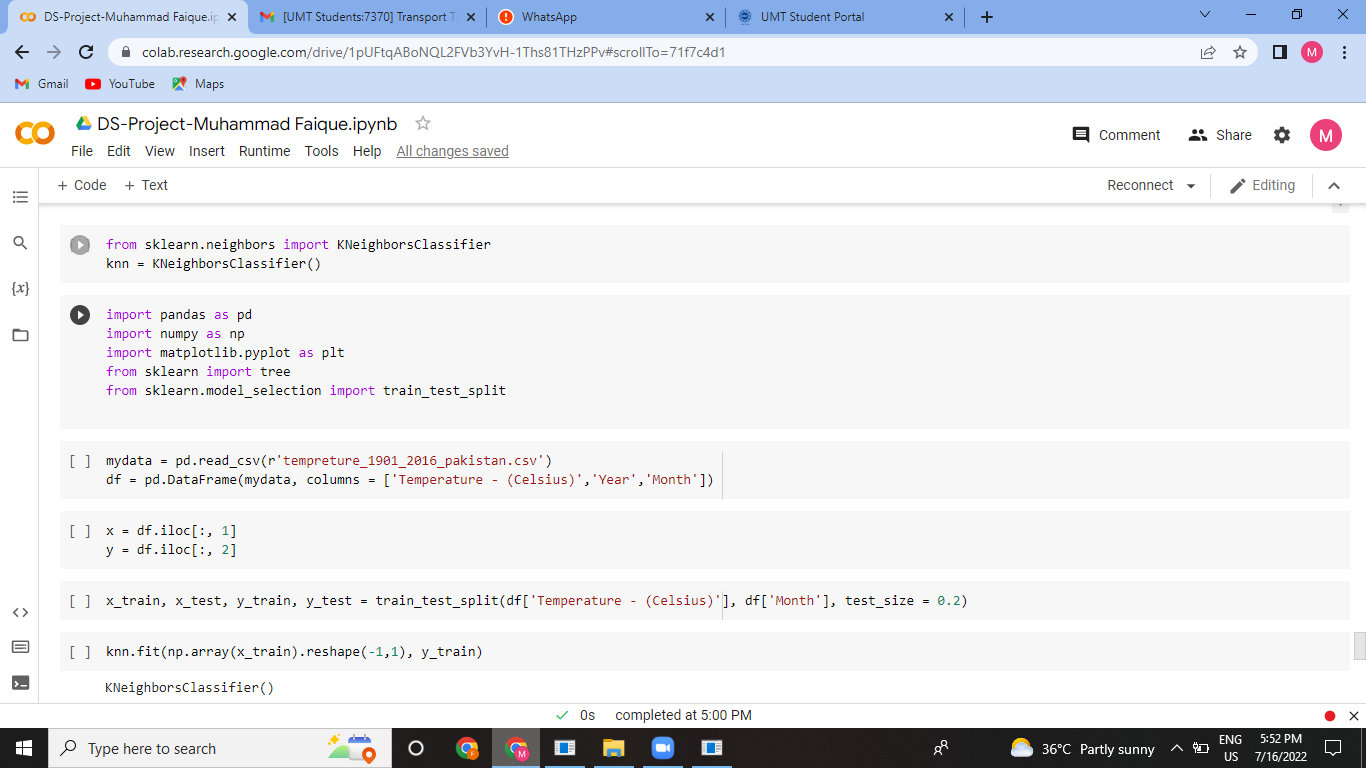


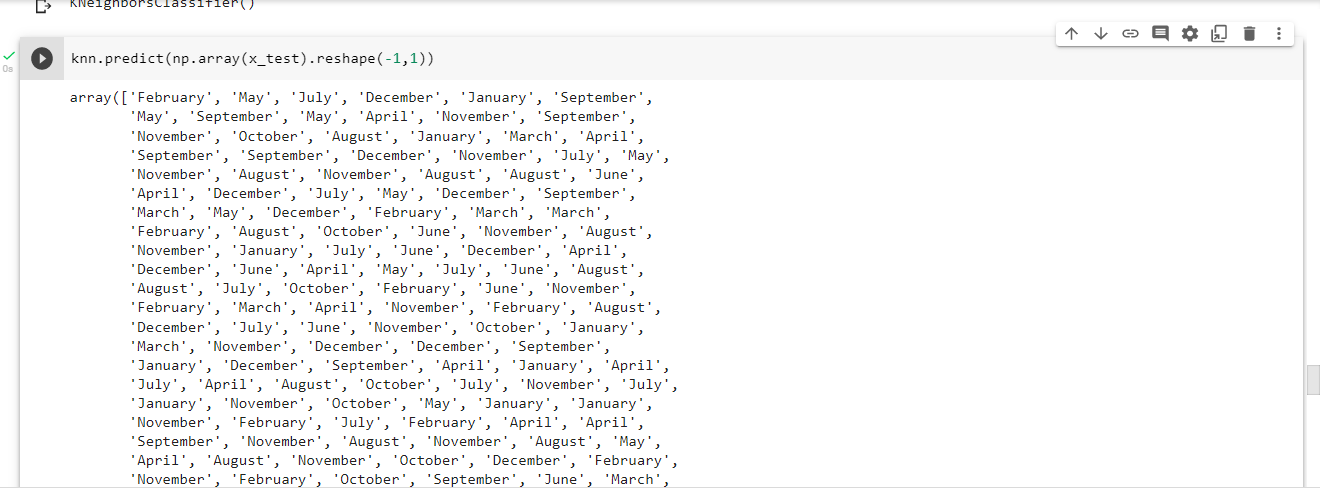


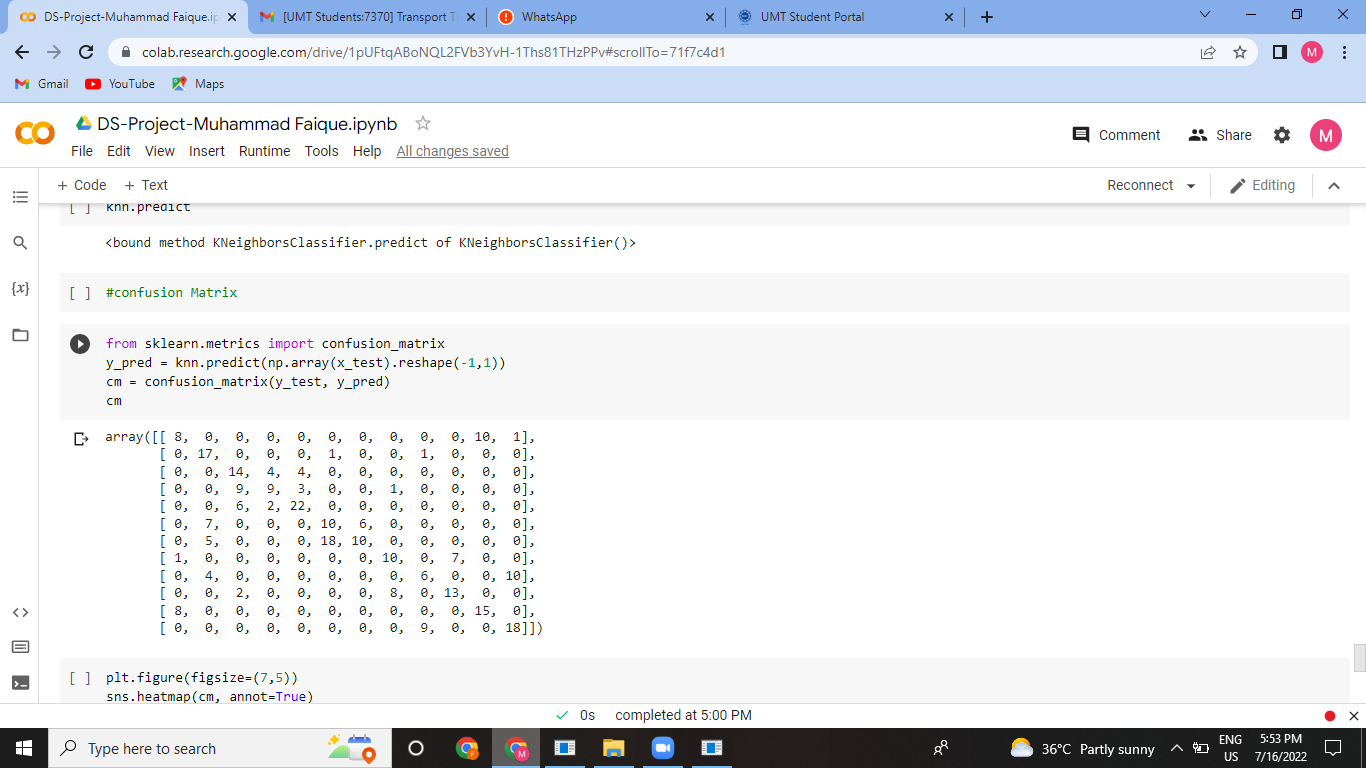




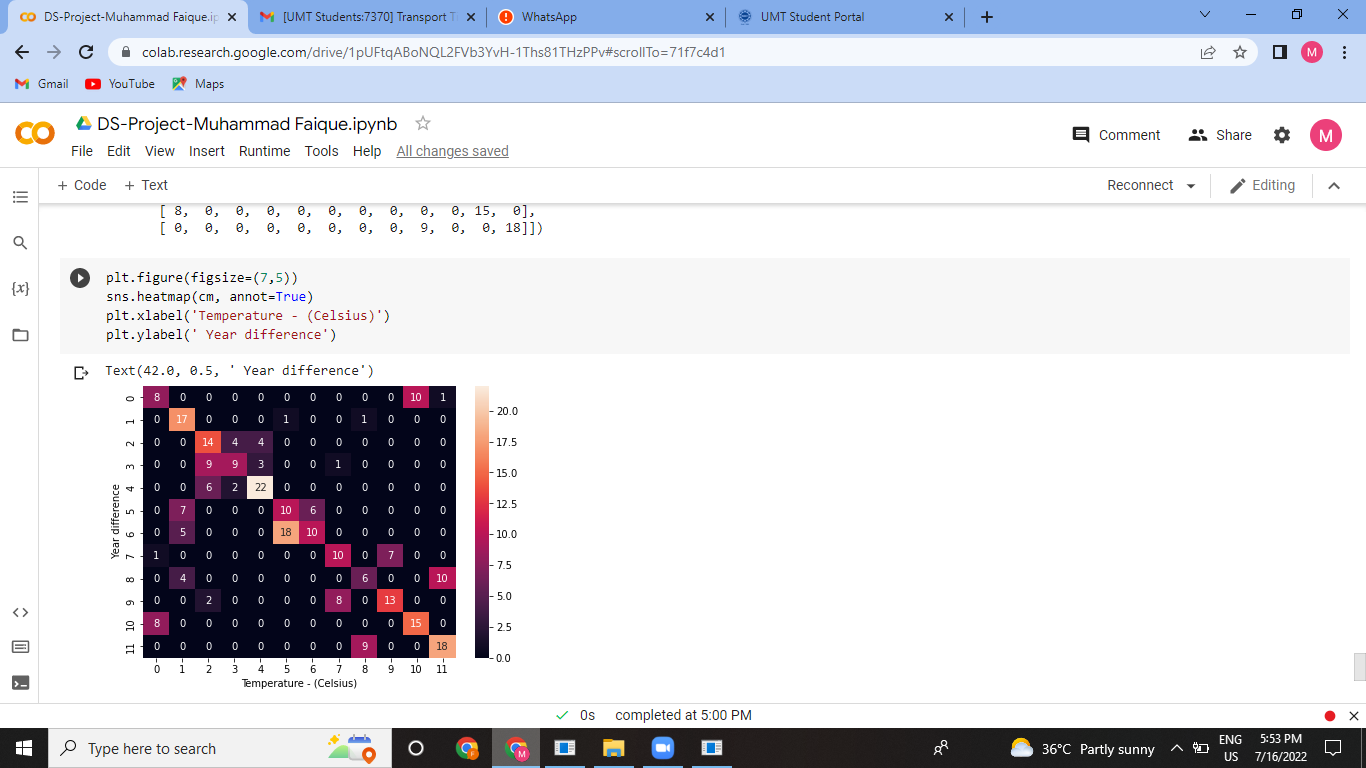








**Heat Map Visualization:**

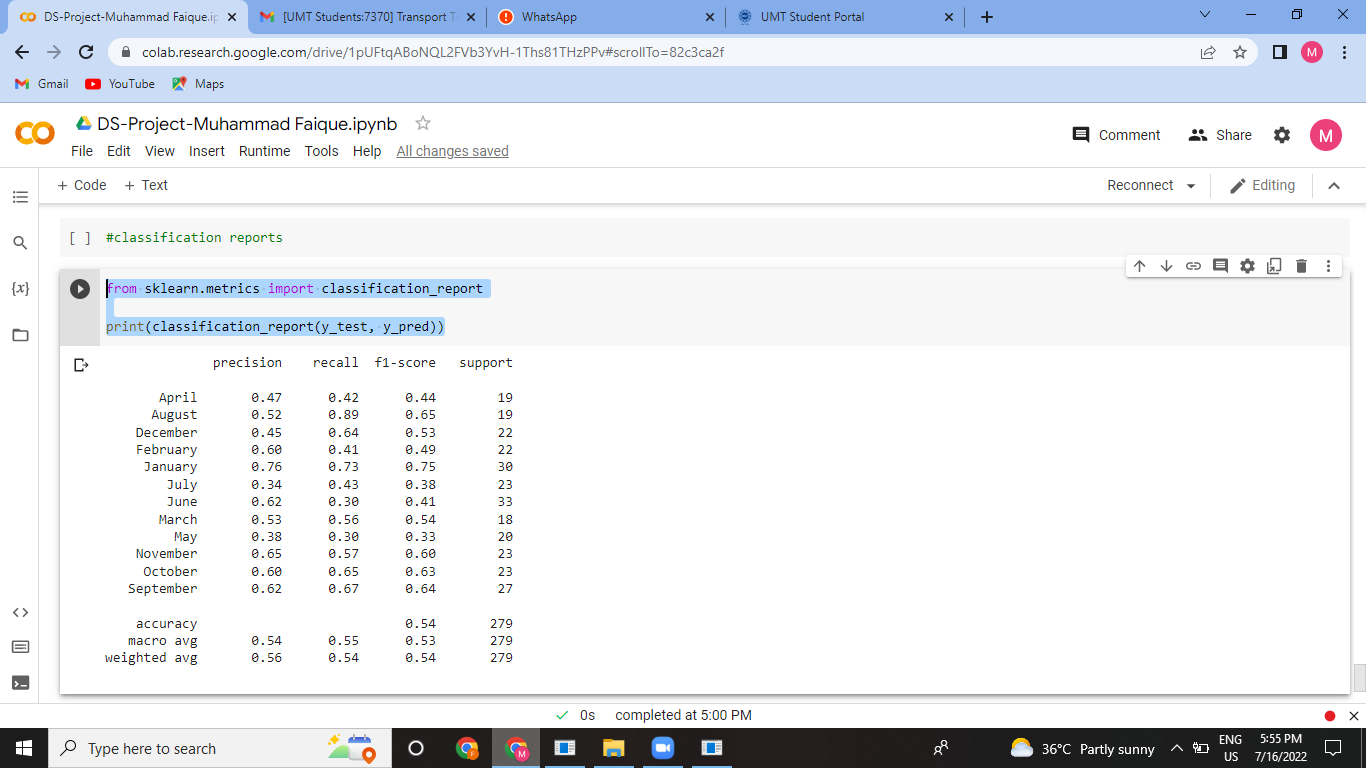


I have plotted Heat Map for better visualization and also finding or checking if there is any missing values.

In the end, I have prepared classification reports.

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_pred)



This shows that months have their following precisions;

April: 52% 0.52 0.52 21

August: 61% 0.83 0.70 24

December: 32% 0.26 0.29 23

February: 36% 0.35 0.36 23

January: 70% 0.91 0.79 23

July: 58% 0.48 0.52 23

June: 33% 0.29 0.31 17

March: 63% 0.77 0.69 22

May: 42% 0.47 0.44 30

November: 81% 0.52 0.63 25

October: 60% 0.60 0.60 20

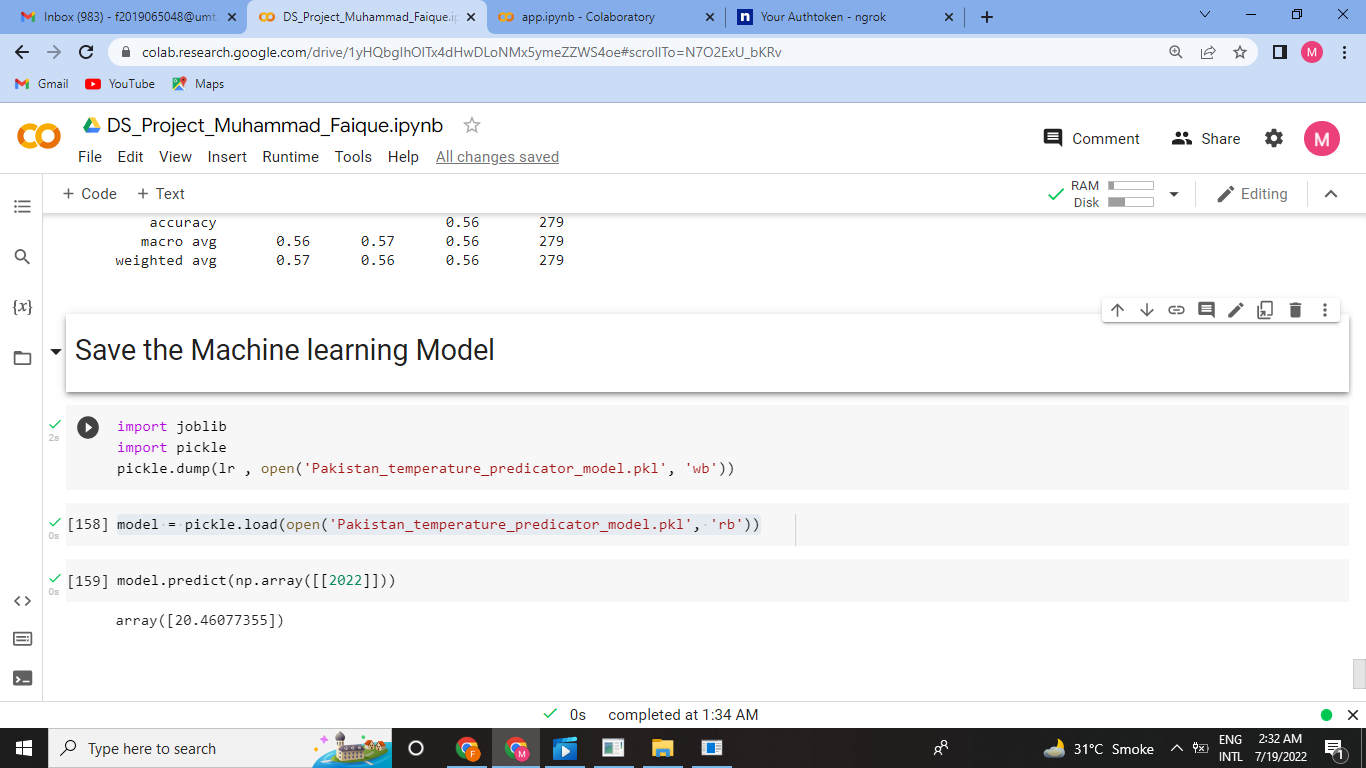
September: 50% 0.43 0.46 28

accuracy 0.54 279

macro avg 0.53 0.54 0.53 279

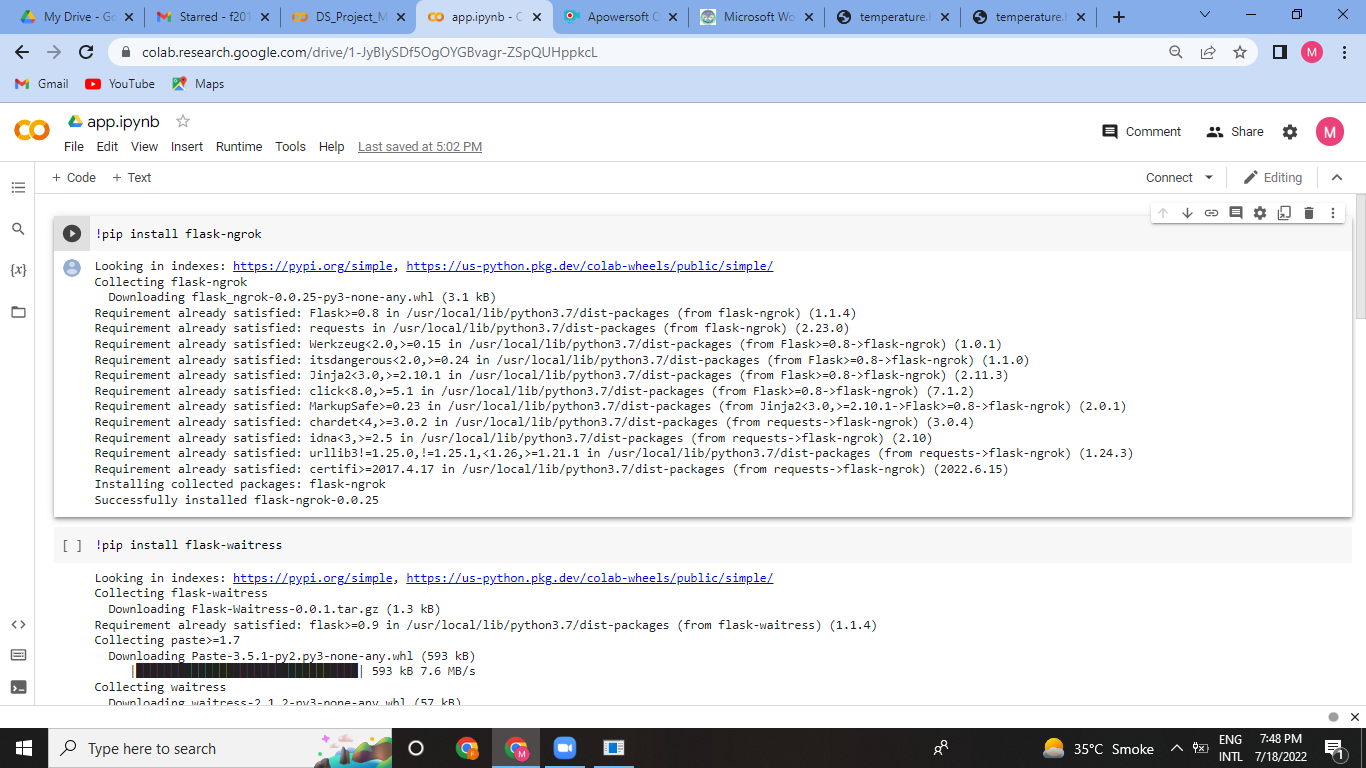
weighted avg 0.53 0.54 0.53 279

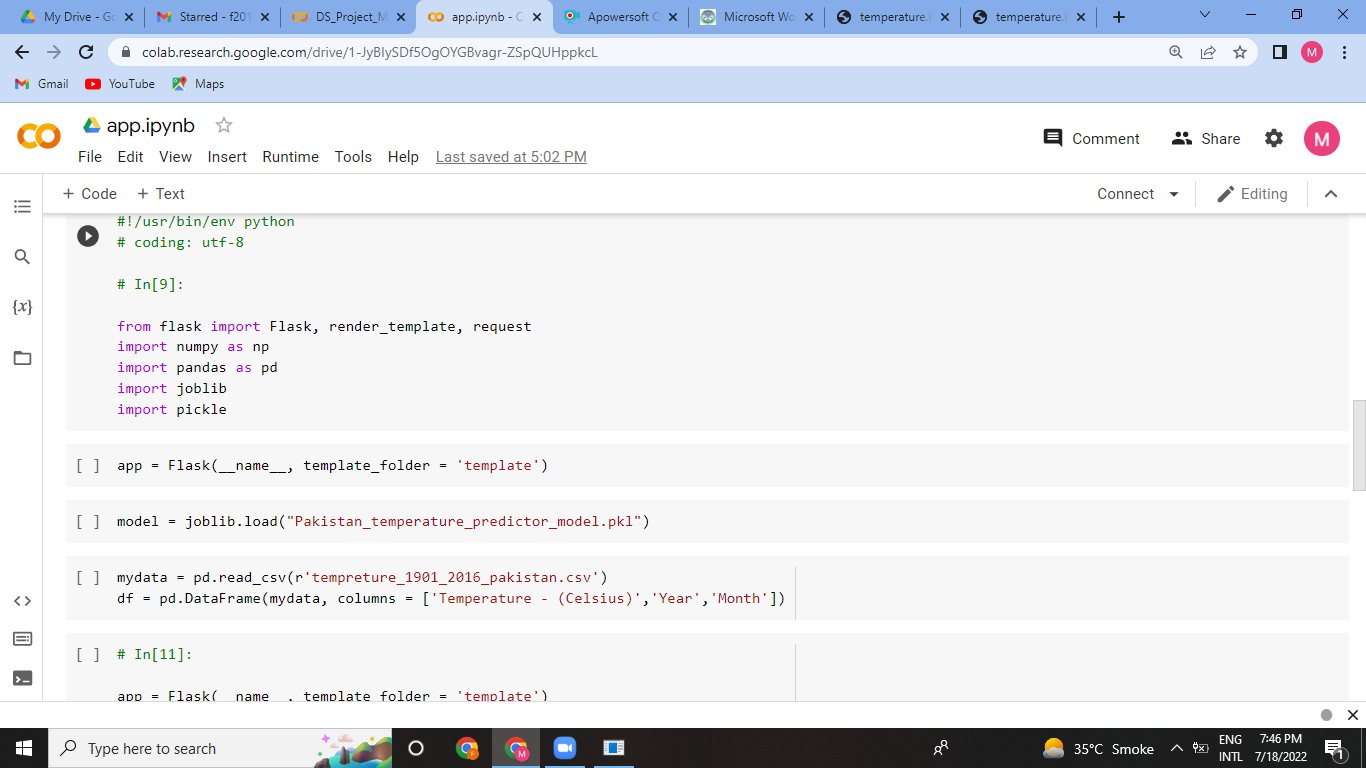
This is the result of our data imbalance.

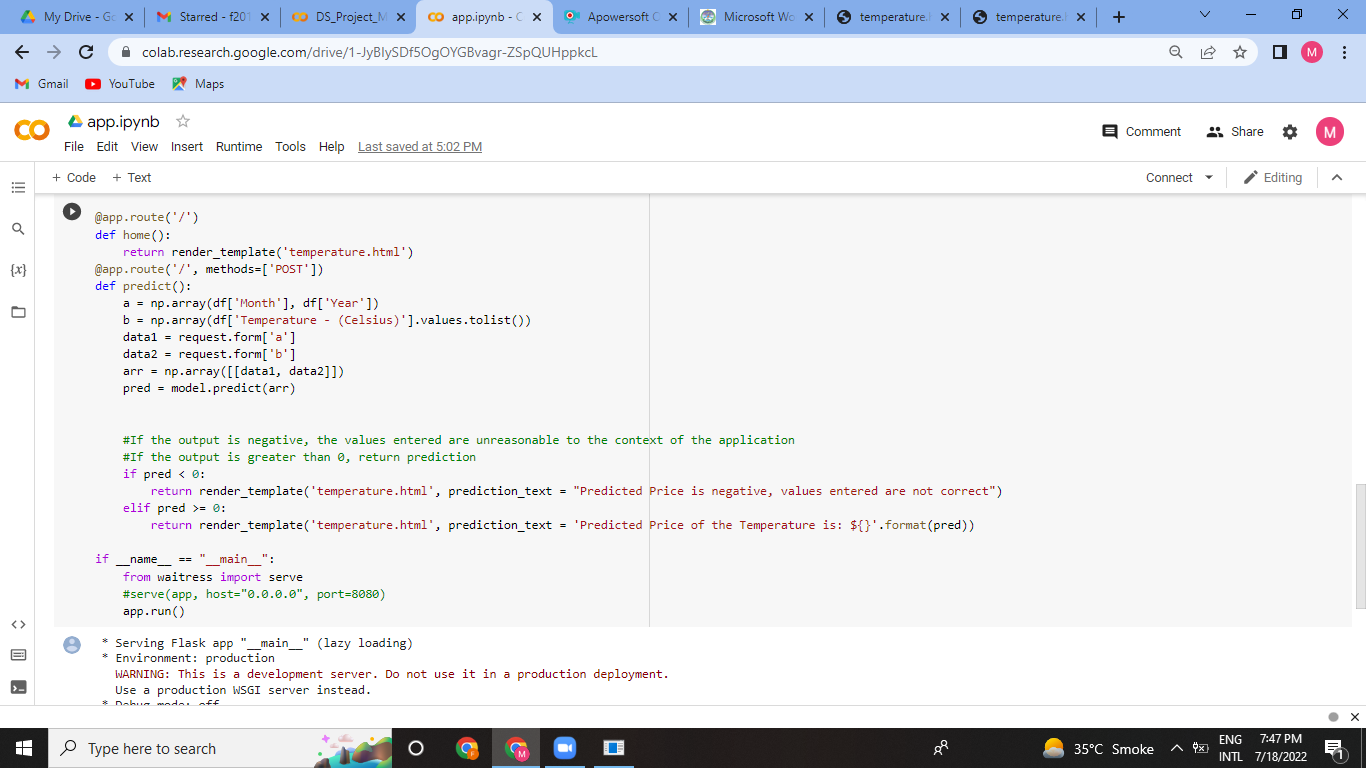


We save our machine learning model using the library joblib. It can be done by using either pickle or joblib library. Once dump is done then .pkl file is created and can be downloaded and I save that in variable model and then predict using linear regression (y=mx+c) model and give the predicated temperature of Pakistan of year 2022.

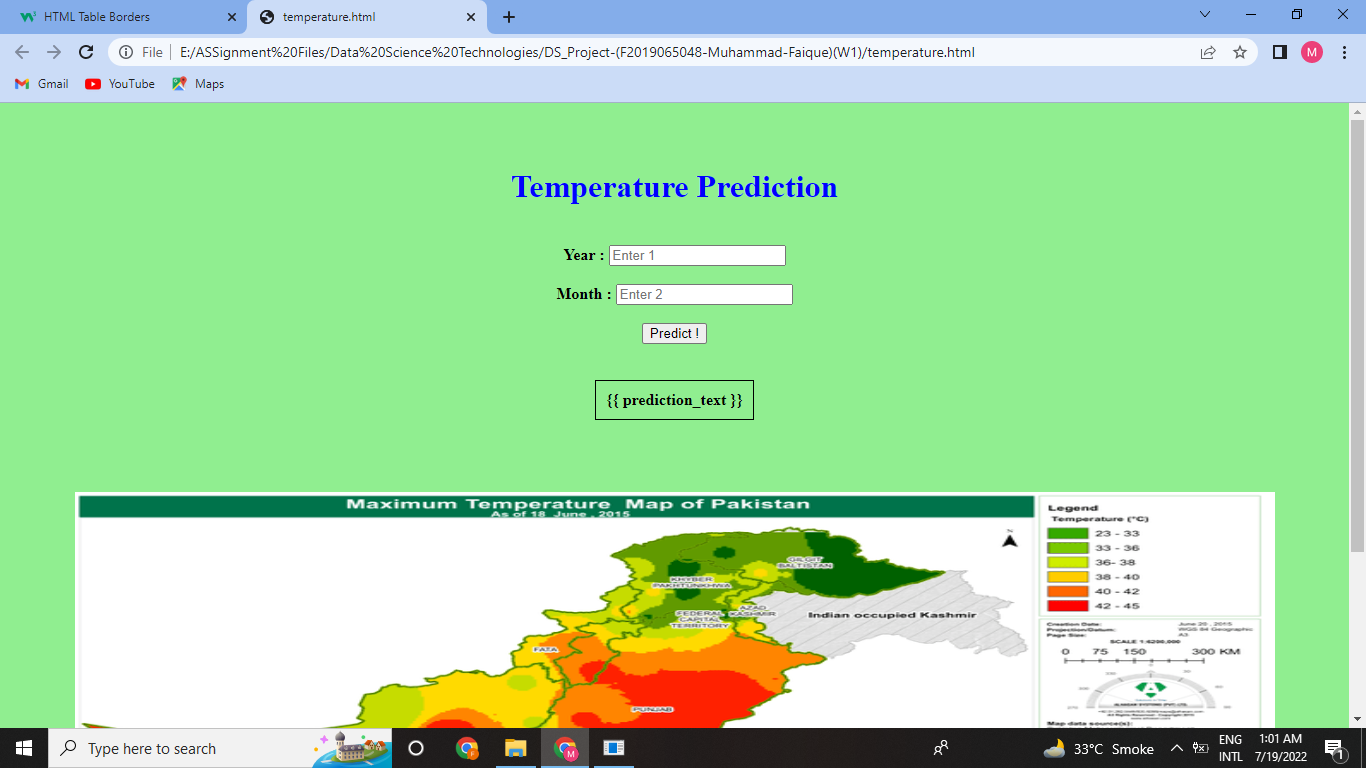
**Deployment through Flask:**

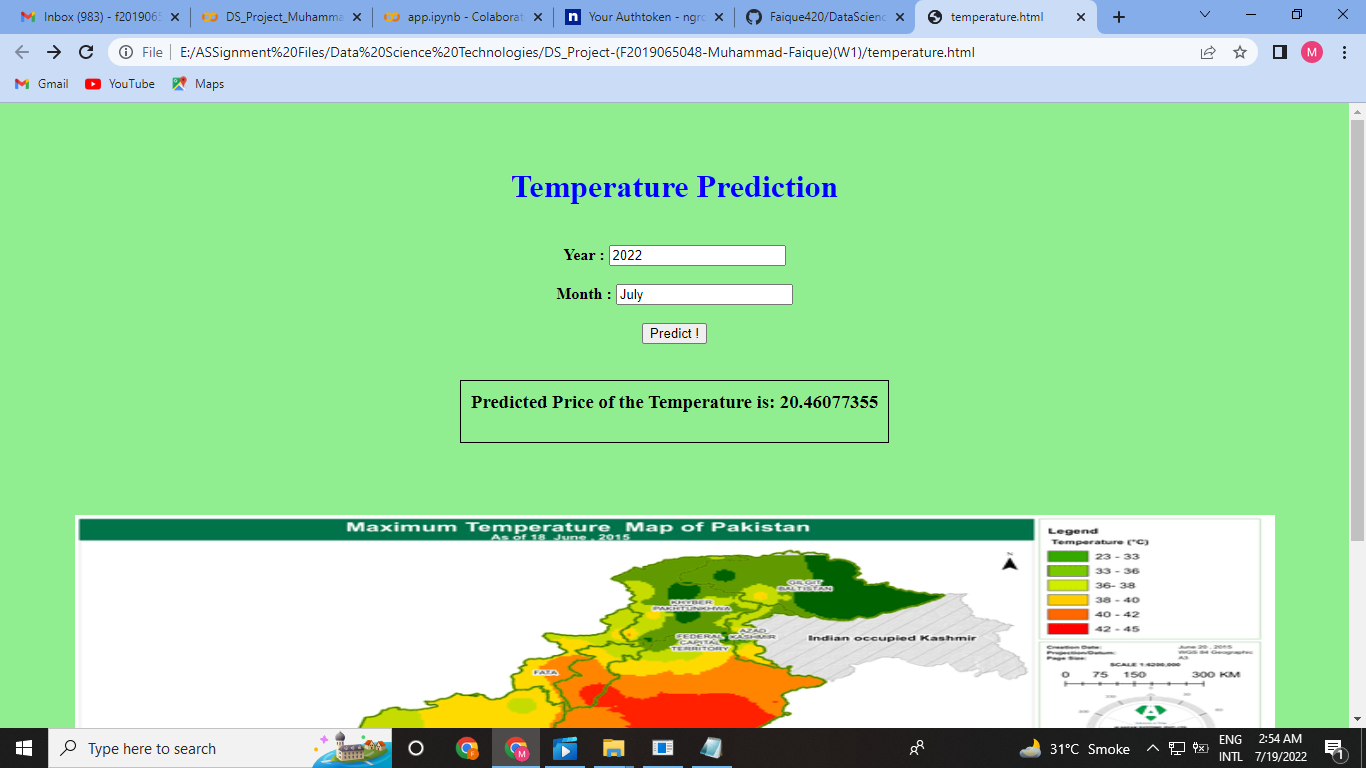






On the web,





**<END>**