**Experiment - 1**

**Student Name: Vivek Kumar UID: 21BCS8129**

**Branch: BE-CSE(LEET) Section/Group: WM-20BCS-616/A**

**Semester: 5th Date of Performance: 16/08/2022**

**Subject Name: Machine Learning Lab Subject Code: 20CSP-317**

**1. Aim/Overview of the practical:**

Implement exploratory data analysis on any data set.

**2. Task to be done/ Which logistics used:**

We have performed the Data analysis on Student\_data.csv and the housing.csv file and perform treatment ofMissing Values and Outliers (Preparation of the dataset for analysis by the treatment of the irregularities.

**3. Algorithm/Flowchart (For programming-based labs):**

**4. Steps for experiment/practical/Code:**

from google.colab import drive

drive.mount('/content/drive')

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sb

data = pd.read\_csv('/content/drive/MyDrive/Data/Students\_data.csv')

data.head()

data.tail()

data["gender"].unique()

data.shape

data.describe()

data.isnull().sum()

data.iloc[:5, 0]

data.iloc[:5, 1]

data['GPA']

data.groupby('race').agg({'GPA': 'count'})

data.groupby('race').agg({'GPA': 'median'})

data.columns

new\_data = data.drop('Calculus1', axis=1)

new\_data.head()

"""\*\*Relationship Analysis\*\*"""

corelation = new\_data.corr()

sb.heatmap(corelation, xticklabels=corelation.columns, yticklabels=corelation.columns, annot=True)

sb.pairplot(new\_data)

sb.relplot(x='GPA',y='Algebra',hue='race',data=data)

sb.distplot(data['Algebra'])

sb.catplot(x='GPA',kind='box',data=data)

df = pd.read\_csv("/content/drive/MyDrive/Data/housing.csv")

df.head()

df.info()

df.head(10)

df.describe()

df.shape

# obtain the missing values present in the given raw Housing Data

df.isnull().sum()

# Six variables among them, having 20 missing values in each case

# Approximately, 4 percent observations in each variable has missing values

(20/506)\*100

# getting the column names of the dataset

df.columns

# Detection of outliers among all variable

# %matplotlib inline

plt.subplots(figsize=(17,10))

df.boxplot(patch\_artist=True, sym="k.")

plt.xticks(rotation=90)

# For first category: "cat\_mv\_out"

cat\_mv = pd.concat([df["CHAS"]],axis=1)

cat\_mv

cat\_mv.isnull().sum()

cat\_mv.mode()

# Replacing the missing values with mode(value 0) to this categorical variable

# replace nan value to zero(mode = 0)

cat\_mv.replace(np.nan, 0, inplace=True)

# After replacing with mode(Value = 0), now there is no missing values in this categorical variable

cat\_mv.isnull().sum()

# dimension (506 Observations and 1 column)

cat\_mv.shape

# For the second category: "num\_mv\_out" means Numerical variables containing missing values and outliers too

num\_mv\_out = pd.concat([df["CRIM"], df["ZN"], df["LSTAT"]],axis=1)

num\_mv\_out.isnull().sum()

"""Each variable has missing values equal to 20 obs"""

num\_mv\_out.describe()

# Replacing the missing values with median of its variables ("num\_mv\_out")

num\_mv\_out = num\_mv\_out.fillna(num\_mv\_out.median())

# Now, "num\_mv\_out" has no missing values

num\_mv\_out.isnull().sum()

num\_mv\_out.shape

# For the third category: "num\_mv\_noOut" means Numerical variables containing missing values but "no outliers"

num\_mv\_noOut = pd.concat([df["INDUS"], df["AGE"]],axis=1)

num\_mv\_noOut

num\_mv\_noOut.isnull().sum()

"""Each variable has missing values equal to 20 obs"""

# Replacing the missing values with mean of its variable ("num\_mv\_noOut")

# this category doesn't have outliers but having missing values in the two variables

num\_mv\_noOut = num\_mv\_noOut.fillna(num\_mv\_noOut.mean())

# Now, this cateory ("num\_mv\_noOut") has no missing values

num\_mv\_noOut.isnull().sum()

"""\*\*\*PHASE 2: TREATMENT OF OUTLIERS\*\*\*

\*\*\*After treatment of missing values, the dataset will have only outliers problems. So, the next treatment will be for outliers. Now, assign a dataset that will contain all 14 variables including the above three category ("Treated Missing Values" Variables). Finally, split this dataset into three categories. But the thing is, Only the first category will be focussed here because the first category contains outliers. The second and third categories have no outliers.\*\*\*

1. num\_out = Numerical variables containing outliers (Missing values will be treated with \*\*\*median\*\*\*)--- "CRIM", "ZN", "RM", "DIS", "PTRATIO", "B", "LSTAT", "MEDV"

2. num\_noOut = Numerical variables containing "no outliers" (Missing values will be treated with \*\*\*mean\*\*\*)--- "INDUS", "NOX", "AGE", "RAD", "TAX"

3. cat\_out = Categorical variable conatining no outliers --- "CHAS"---- In this variable, the observation is either 1 or 0

"""

# For assigning or concatenating all the variables including with six treated missing values variables into a dataset

df1 = pd.concat([cat\_mv,num\_mv\_out, num\_mv\_noOut, df["RM"], df["DIS"], df["PTRATIO"], df["B"], df["MEDV"], df["NOX"], df["RAD"], df["TAX"]],axis=1)

df1

# No missing values after merging all variables

df1.isnull().sum()

# Boxplot for all variables

plt.subplots(figsize=(17,10))

df1.boxplot(patch\_artist=True, sym="k.")

plt.xticks(rotation=90)

"""Nine variables containing outliers and remain doesn't have outliers

\*\*\*Now, It's time for treatment of outliers\*\*\*

1. num\_out = Numerical variables containing outliers (Missing values will be treated with \*\*\*median\*\*\*)--- "CRIM", "ZN", "RM", "DIS", "PTRATIO", "B", "LSTAT", "MEDV"

"""

num\_out = pd.concat([df1["CRIM"], df1["ZN"], df1["RM"], df1["DIS"], df1["PTRATIO"], df1["B"], df1["LSTAT"], df1["MEDV"]],axis=1)

num\_out

# Detecting outliers in "cat\_out"

plt.subplots(figsize=(17,10))

num\_out.boxplot(patch\_artist=True, sym="k.")

plt.xticks(rotation=90)

# Getting the basic statistical summary of those variables containing outliers

num\_out.describe()

# Detecting and Removing Outliers

# Inter Quartile Range (IQR) is the difference between the 3rd Quartile and the first Quartile

# The data points which fall below Q1 – 1.5 IQR or above Q3 + 1.5 IQR are outliers.

def detect\_outlier(feature):

    Q1 = np.percentile(feature, 25)

    Q3 = np.percentile(feature, 75)

    IQR = Q3 - Q1

    IQR \*= 1.5

    minimum = Q1 - IQR

    maximum = Q3 + IQR

    flag = False

    if(minimum > np.min(feature)):

        flag = True

    if(maximum < np.max(feature)):

        flag = True

    return flag

"""Using tukey method to remove outliers. Whiskers are set at 1.5 times Interquartile Range (IQR). Any value beyond the acceptance range are considered as outliers.

\*\*\*Replacing the outliers with the median value of that feature\*\*\*

\*\*\*Why replacing with median value?\*\*\*

As the mean value is highly influenced by the outliers, it is advised to replace the outliers with the median value.

"""

def  remove\_outlier(feature):

    Q1 = np.percentile(num\_out[feature], 25)

    Q3 = np.percentile(num\_out[feature], 75)

    IQR = Q3 - Q1

    IQR \*= 1.5

    minimum = Q1 - IQR # the acceptable minimum value

    maximum = Q3 + IQR # the acceptable maximum value

    median = num\_out[feature].median()

    num\_out.loc[num\_out[feature] < minimum, feature] = median

    num\_out.loc[num\_out[feature] > maximum, feature] = median

# taking all the column

num\_out = num\_out.iloc[:, : ]

for i in range(len(num\_out.columns)):

        remove\_outlier(num\_out.columns[i])

# In "num\_out" matrix, it contains all varibles

num\_out = num\_out.iloc[:, : ]

num\_out

# This shows that these are the variables from "num\_out" which contain Outliers

for i in range(len(num\_out.columns)):

    if(detect\_outlier(num\_out[num\_out.columns[i]])):

        print(num\_out.columns[i], "Contains Outlier")

# Removing the outliers

for i in range (3):

    for i in range(len(num\_out.columns)):

        remove\_outlier(num\_out.columns[i])

# After removing outliers, the following boxplots of each variable from "num\_out" show, they have no more outliers

plt.subplots(figsize=(17,10))

num\_out.boxplot(patch\_artist=True, sym="k.")

plt.xticks(rotation=90)

# Finally, concatenating all variables after treatment of outliers with those varibales that have no outliers into a dataset

final\_df = pd.concat([num\_out, df1["CHAS"], df1["INDUS"], df1["NOX"], df1["AGE"], df1["RAD"], df1["TAX"]],axis=1)

"""# After treatment of missing values as well as outliers

# The dataset is now ready for further analysis

"""

final\_df

# Boxplot for the final dataset

plt.subplots(figsize=(17,10))

final\_df.boxplot(patch\_artist=True, sym="k.")

plt.xticks(rotation=90)

# Here, Correlation matrix shows:

# the relationship among explanatory variables as well as,

# the relationship between the dependent varibale with each of the explanatory variables

sb.pairplot(final\_df)

# the heatmap also shows the same things and interpretations which earlier correlation matrix has been shown

fig, ax = plt.subplots(figsize=(17,10))

correlation\_matrix = final\_df.corr().round(2)

# annot = True to print the values inside the square

sb.heatmap(data=correlation\_matrix, annot=True)

print("PEARSON CORRELATION")

print(final\_df.corr(method="pearson"))

sb.heatmap(final\_df.corr(method="pearson"))

plt.savefig("heatmap\_pearson\_final.png")

plt.clf()

plt.close()

#scatter plot to see how these features RAD, RM ,DIS, LSTAT, NOX, AGE, TAX, INDUS vary with Target variable (MEDV)

plt.figure(figsize=(17,5))

features = ['LSTAT','NOX','AGE','TAX','RM','DIS','INDUS']

target = final\_df['MEDV']

for i, col in enumerate(features):

    plt.subplot(1, len(features) , i+1)

    x = final\_df[col]

    y = target

    plt.scatter(x, y, marker='o')

    plt.title(col)

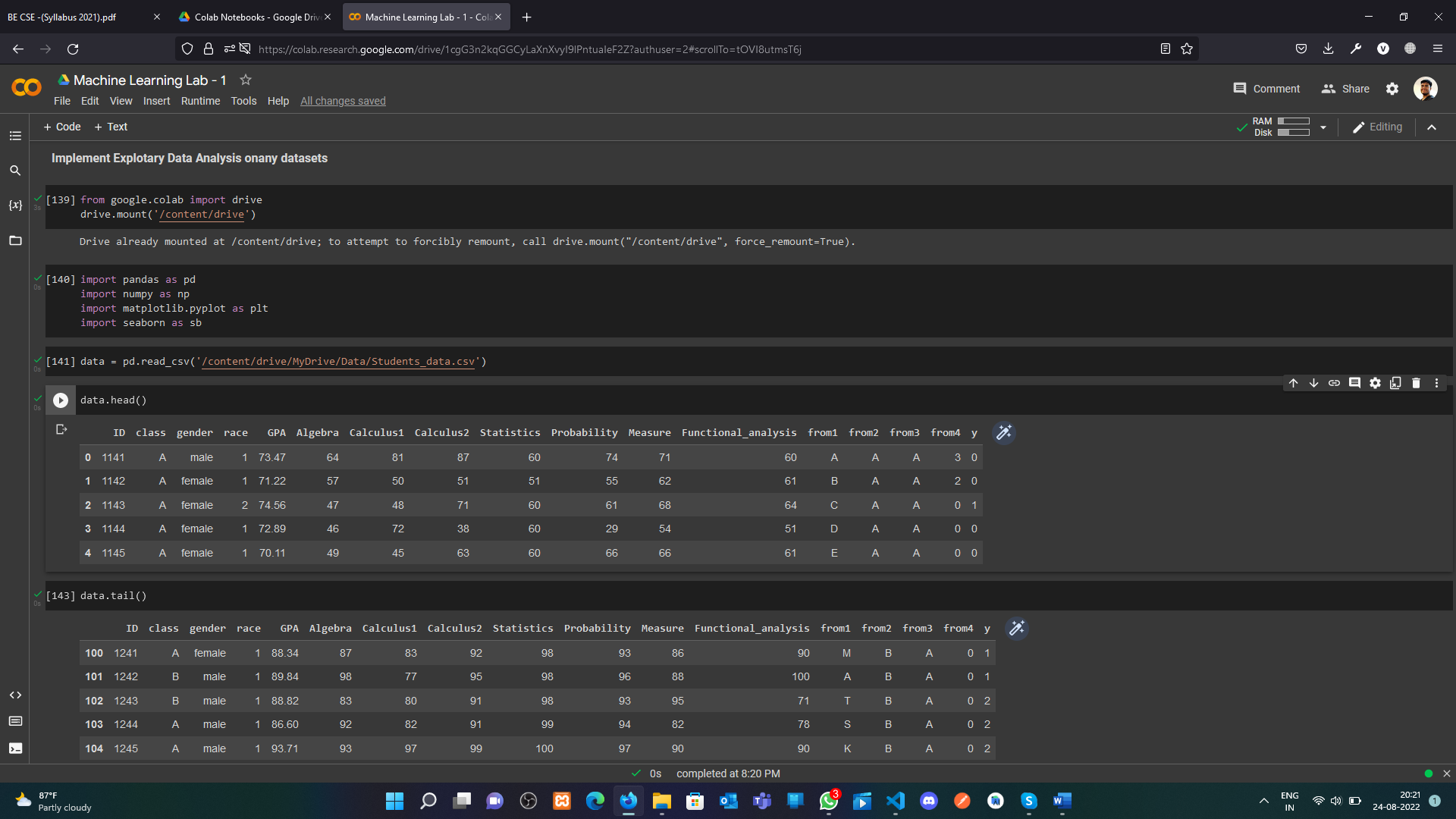
    plt.xlabel(col)

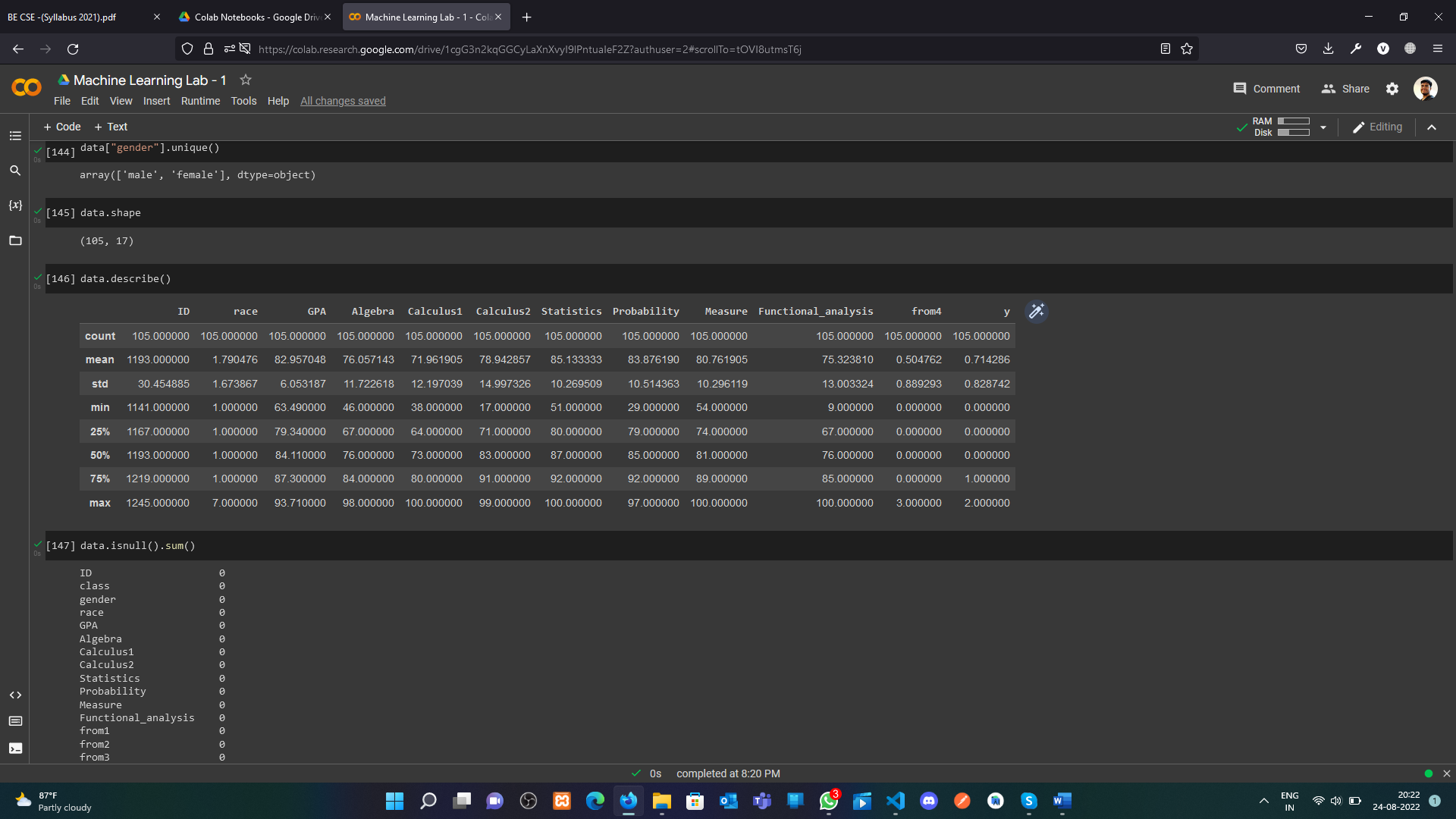
    plt.ylabel('MEDV')

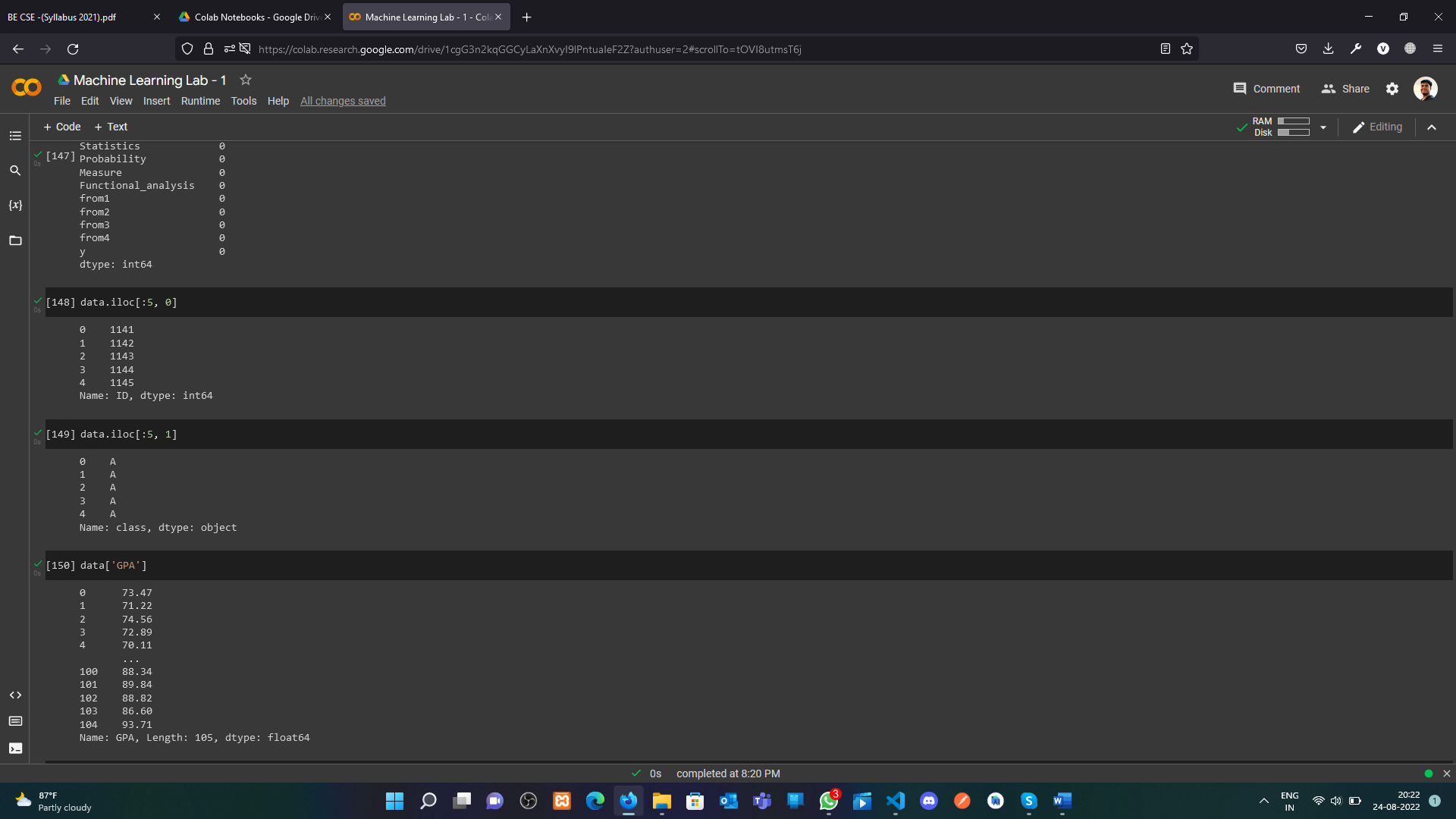
**5. Observations/Discussions/ Complexity Analysis:**

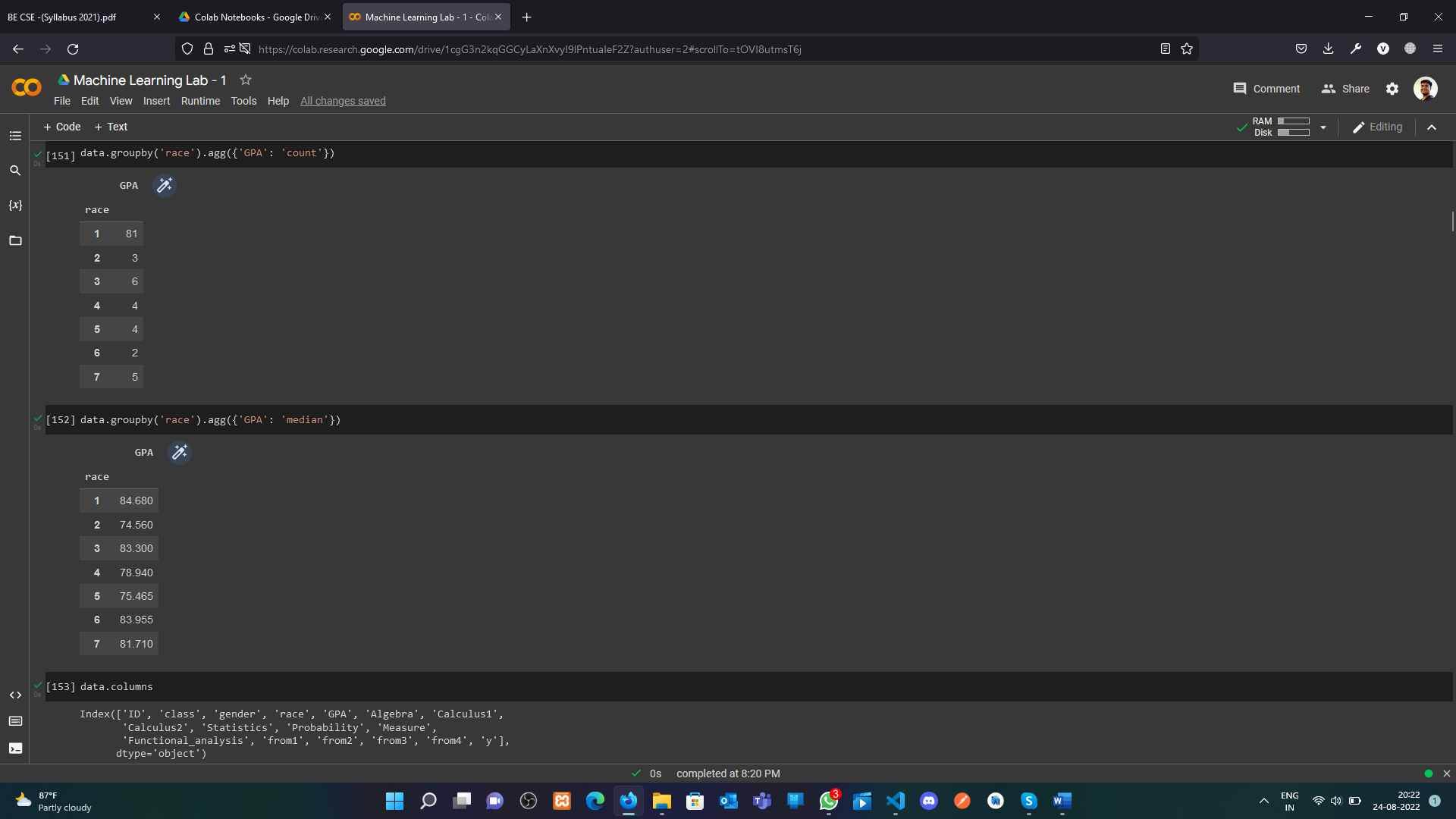
In this Experiment we have done the Data manipulation in such way where we have found the Unique, count of data, head, Tail, shape, and Descriptive data analysis, iloc, groupby, column, and core relation of the data. Moreover, we have Plotted the graph using seaborn library such as heatmap, pairplot, relplot, distplot and catplot.

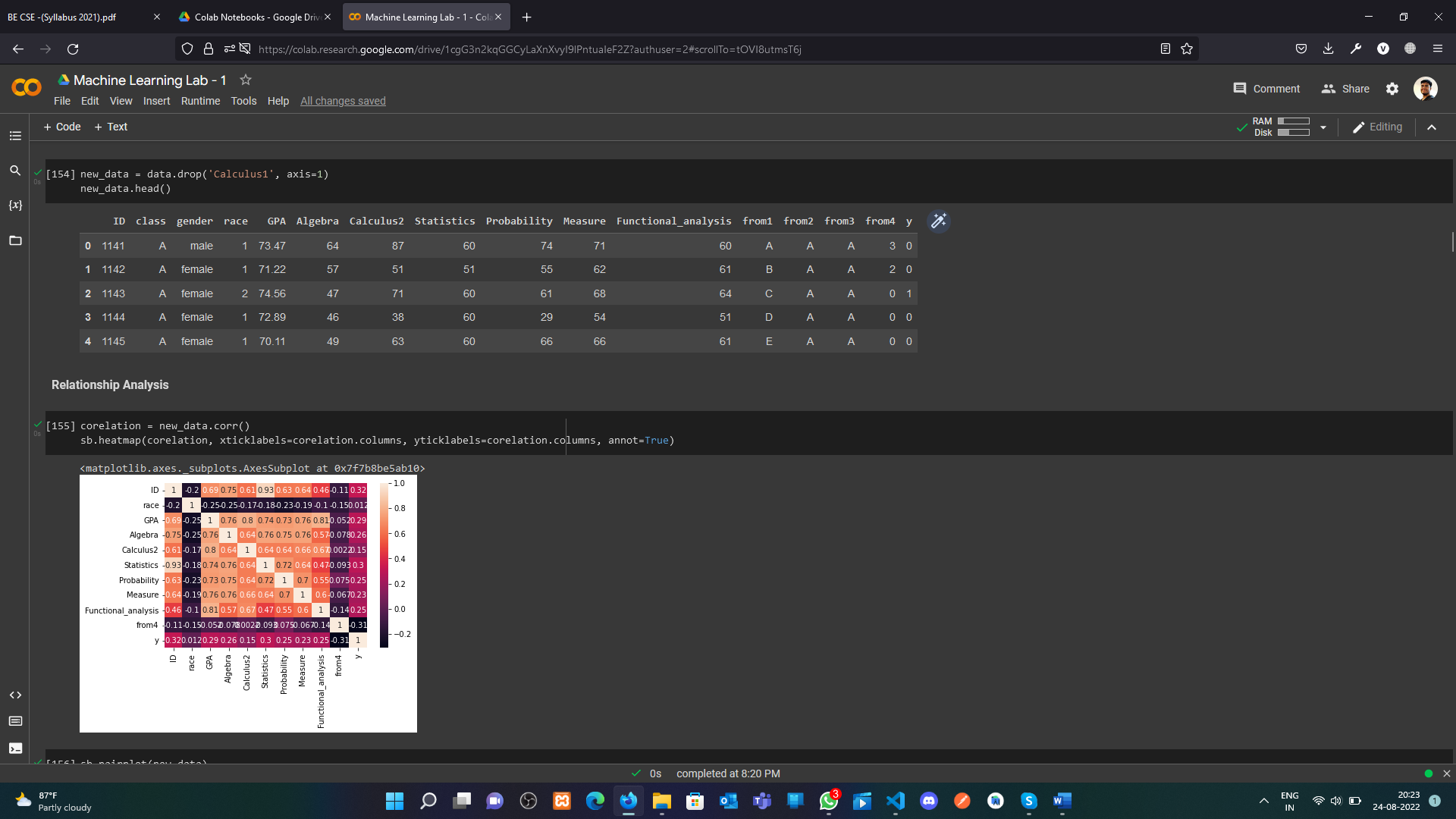
**6. Result/Output/Writing Summary:**

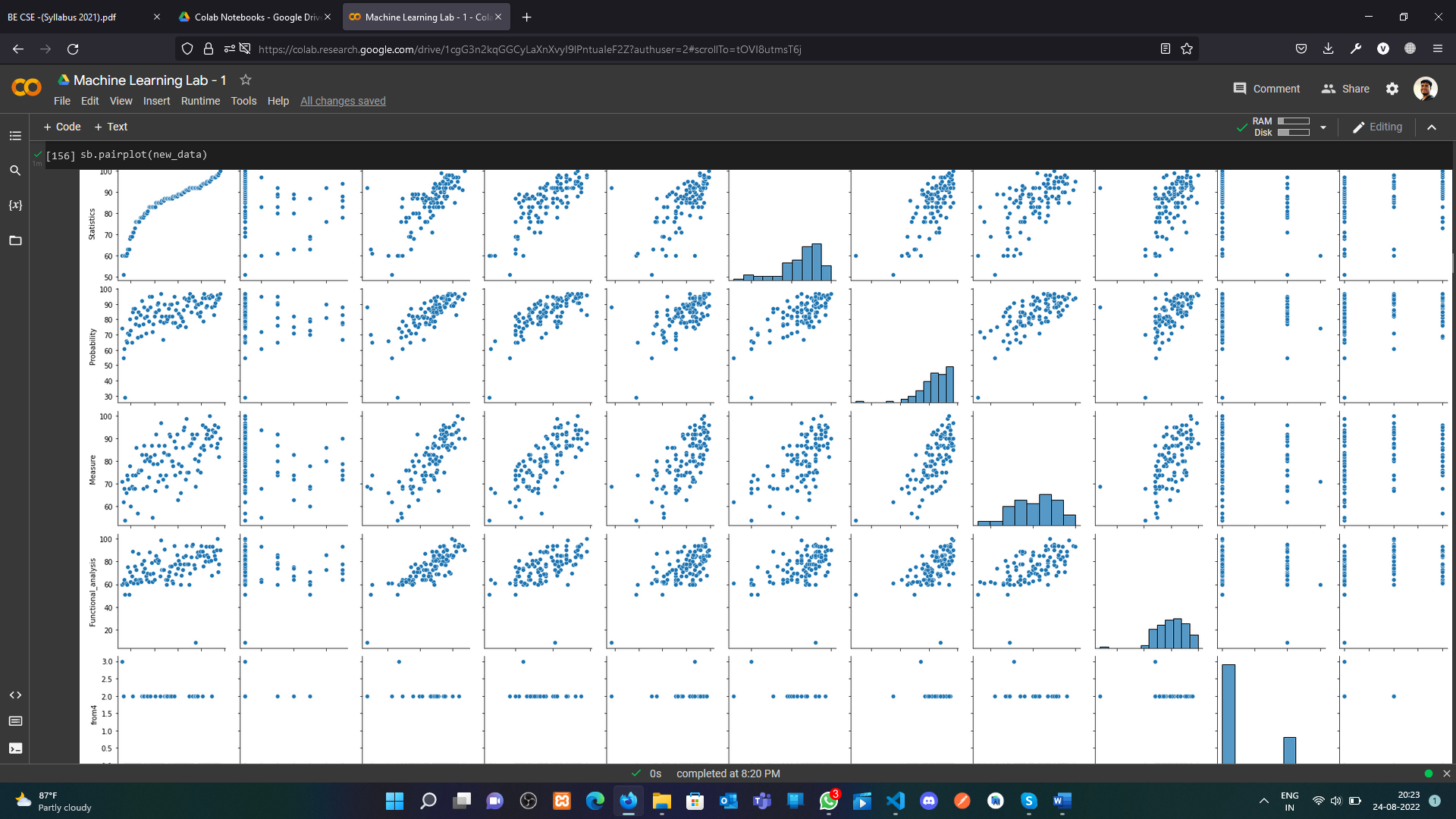


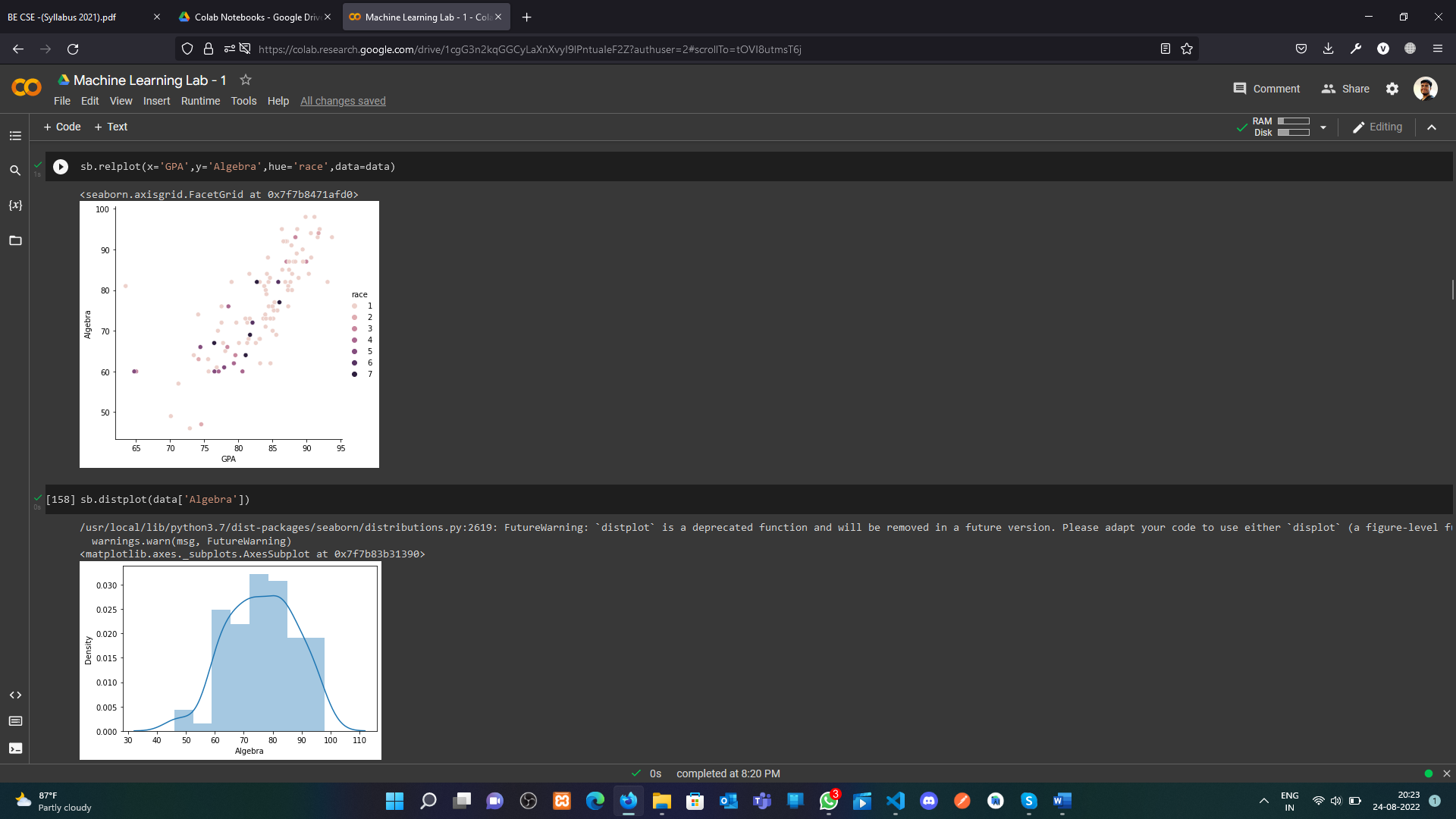


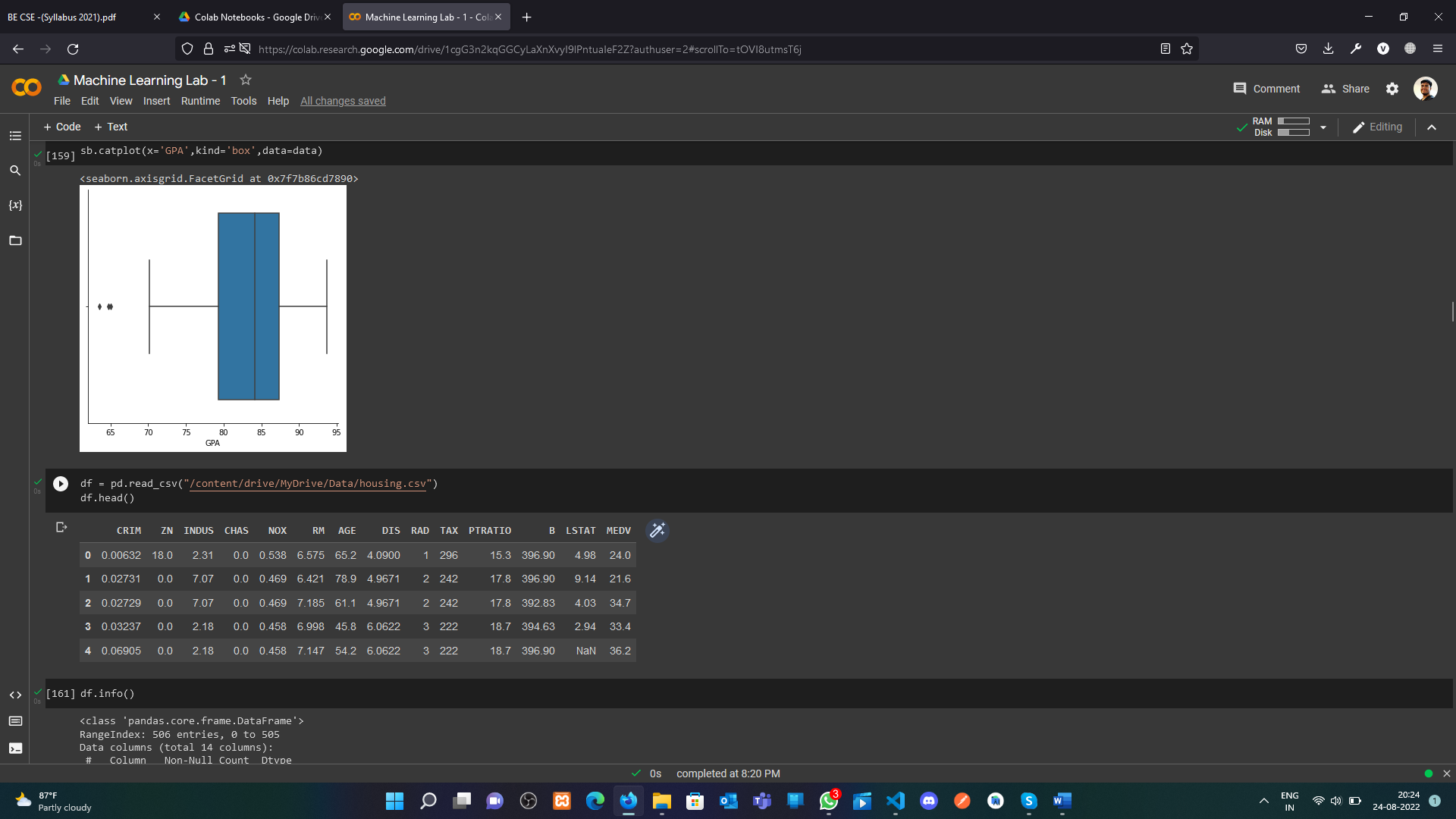




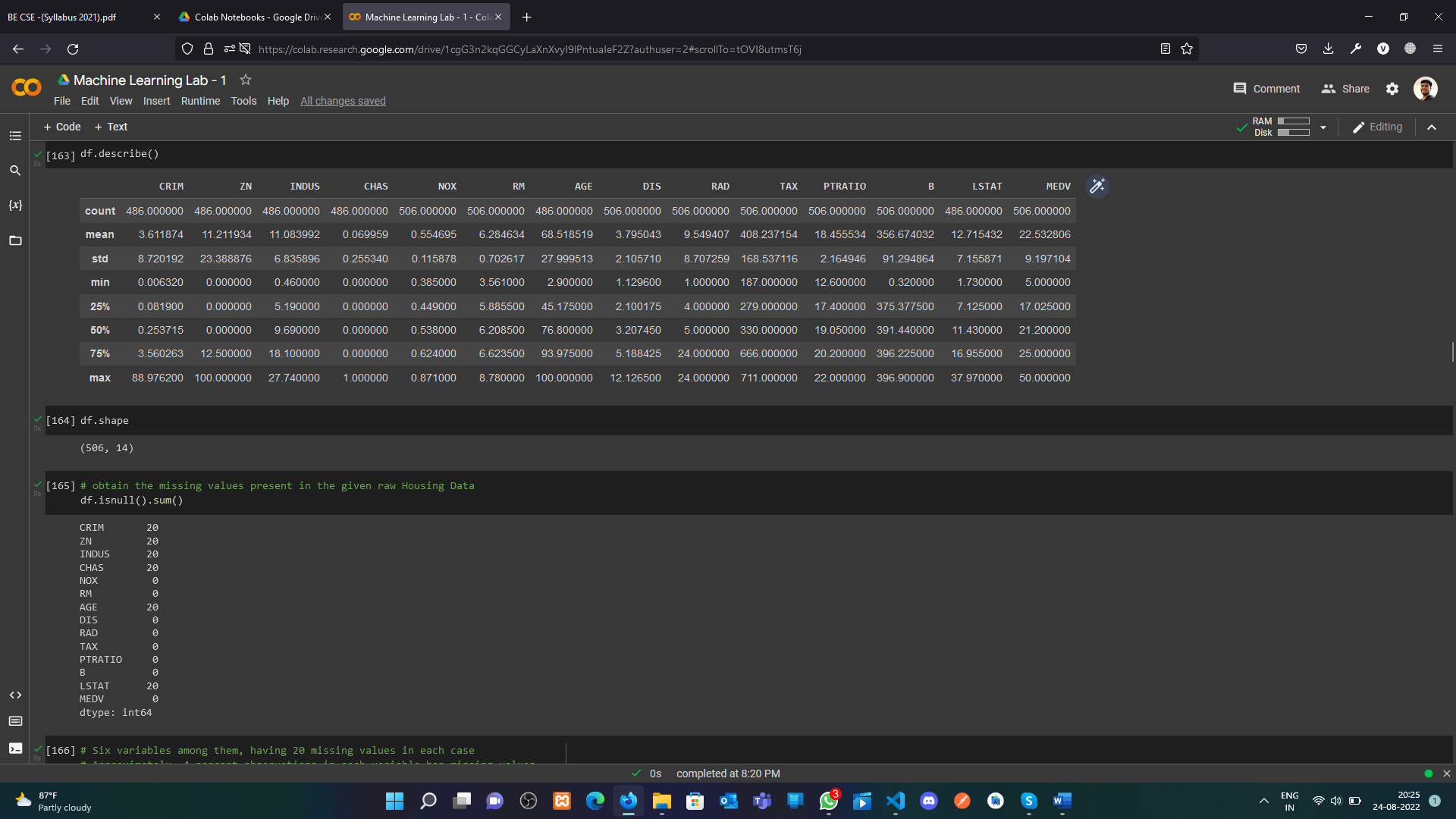


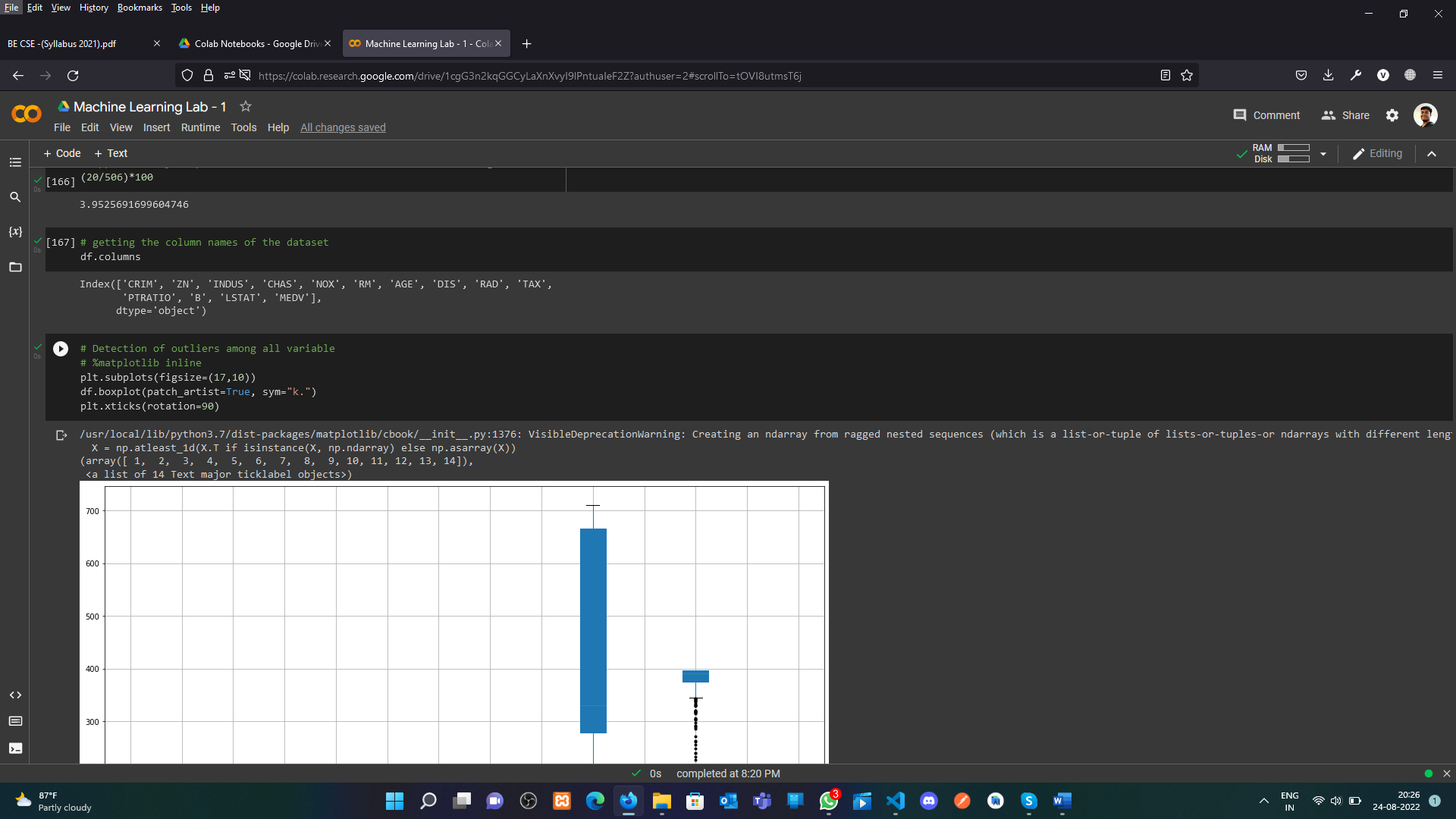


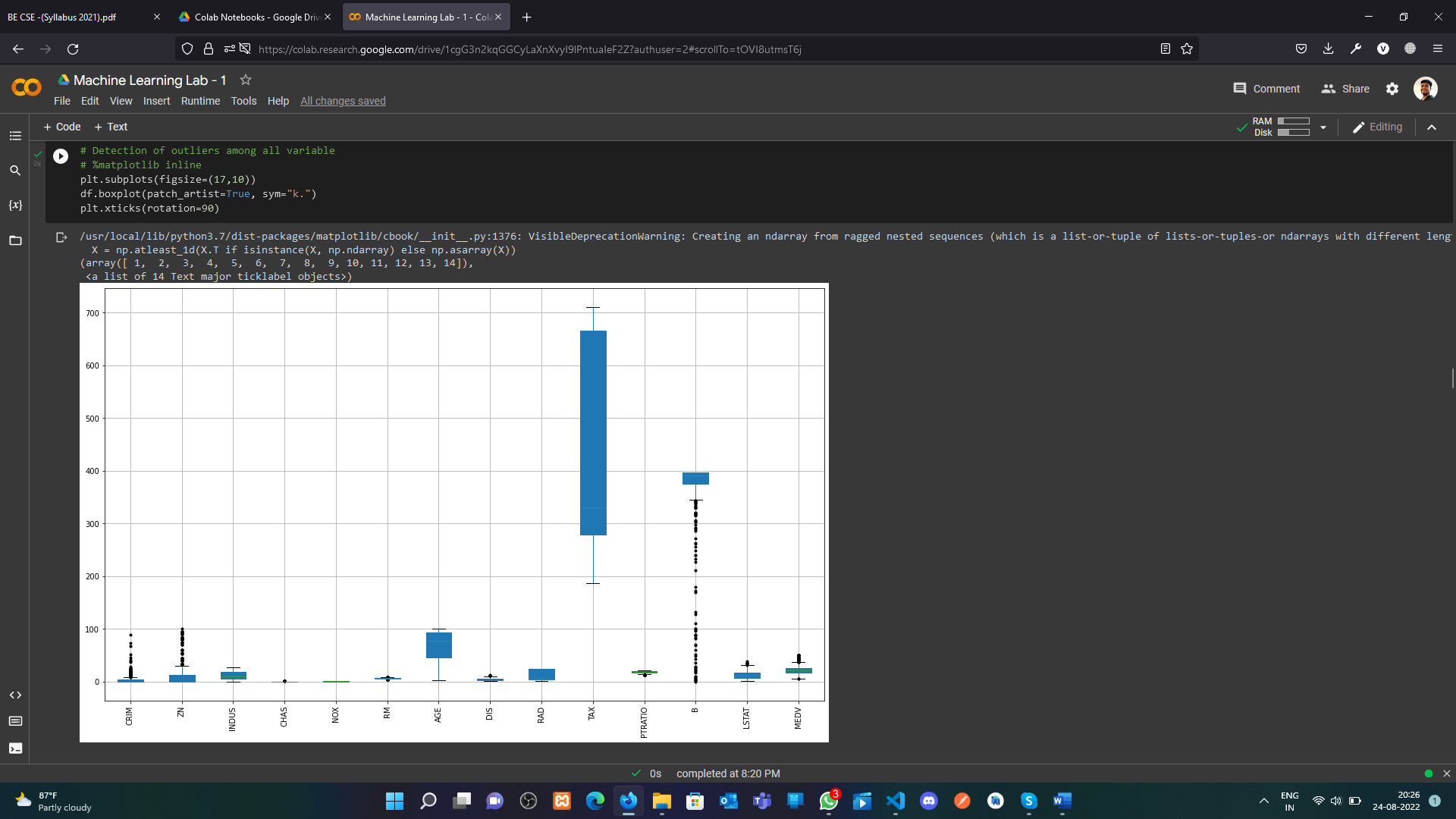


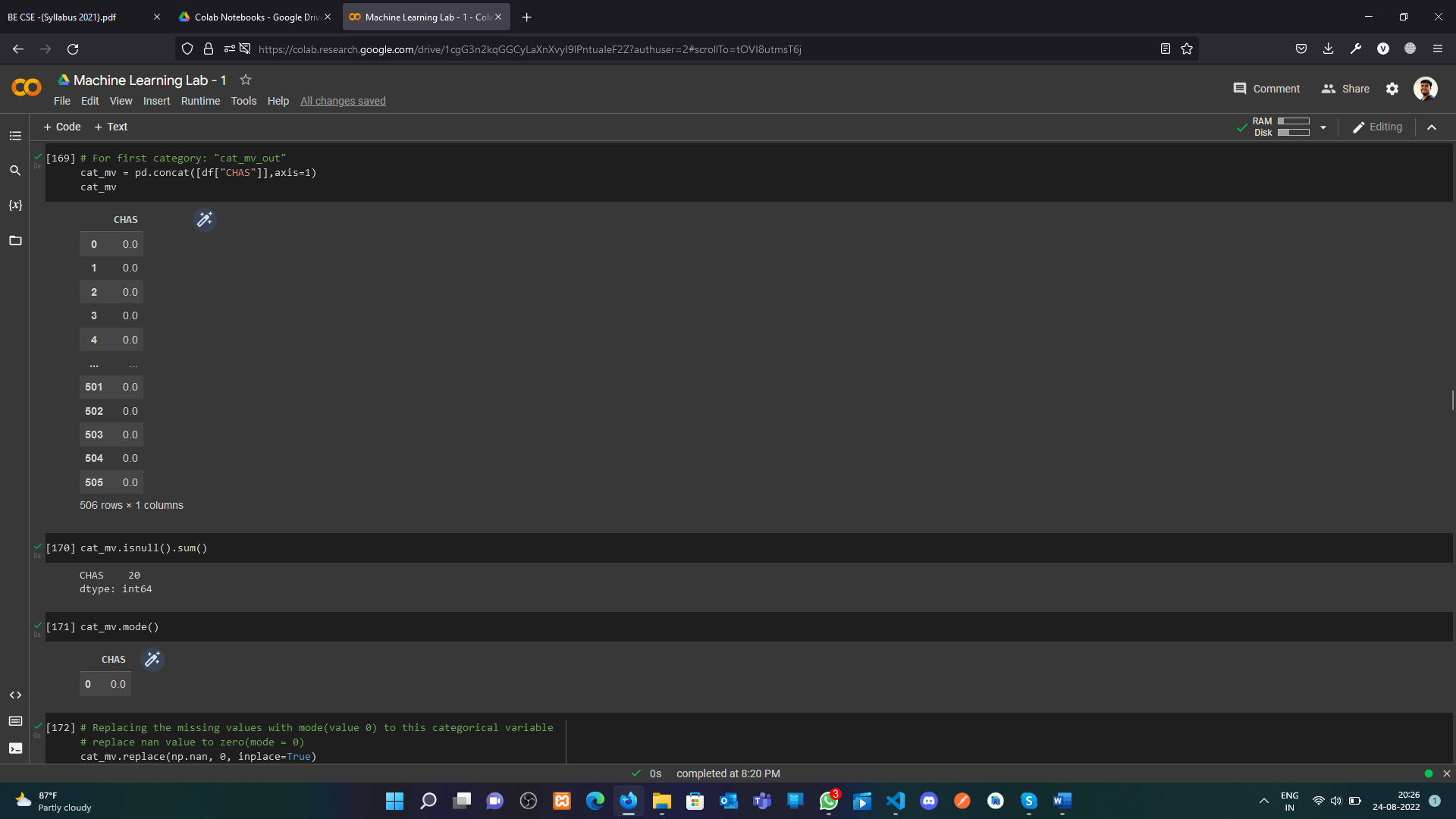


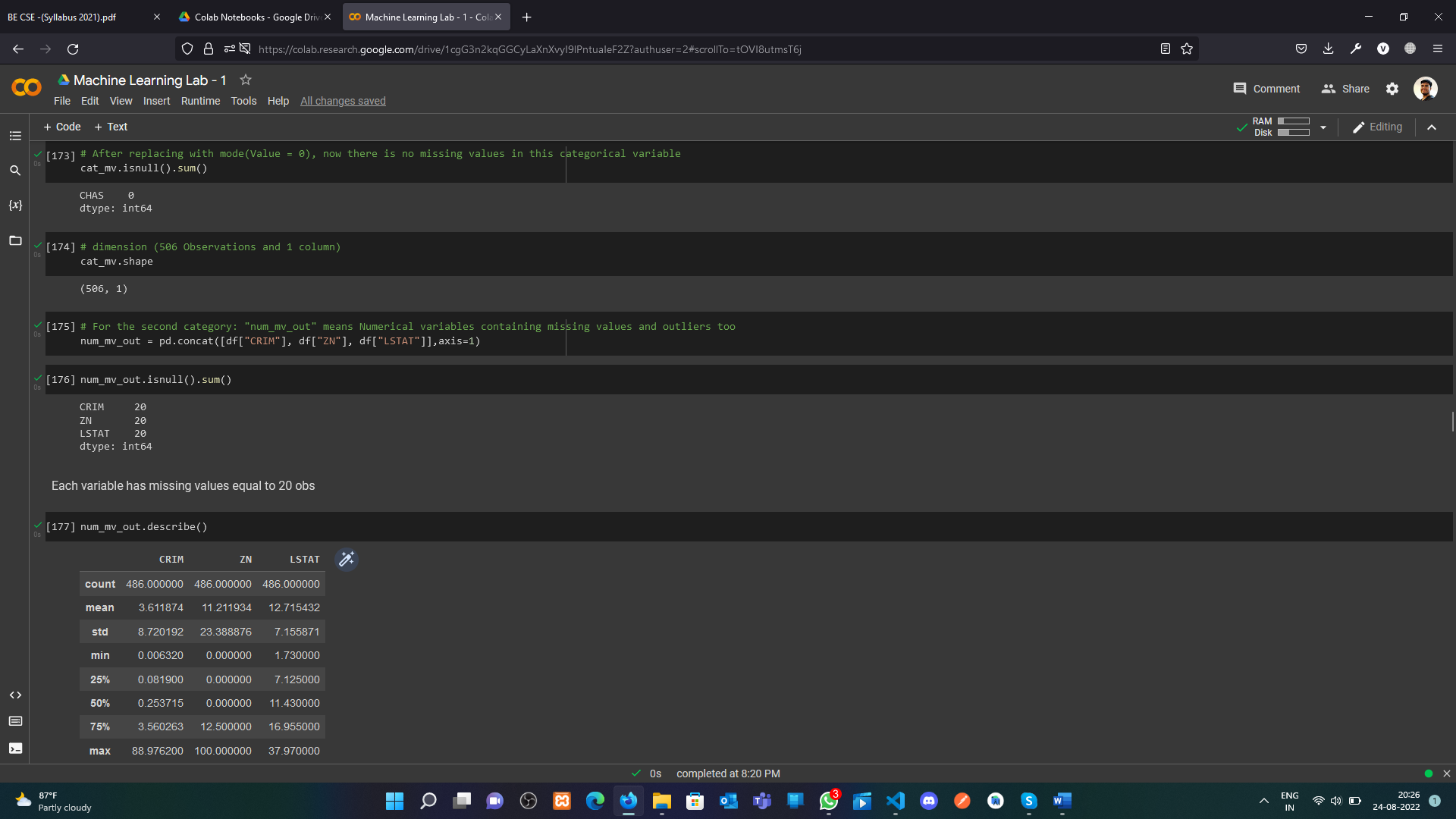


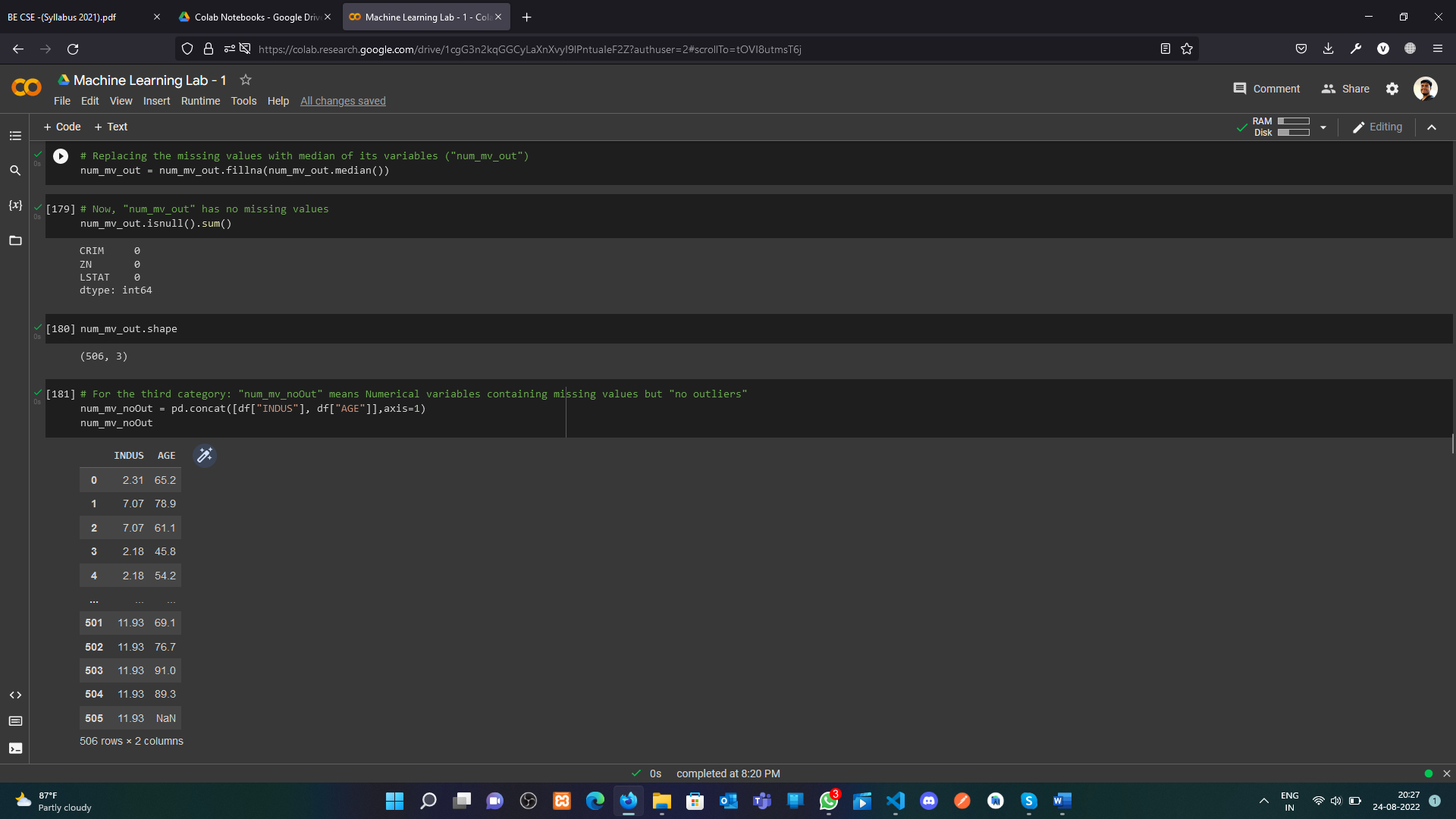


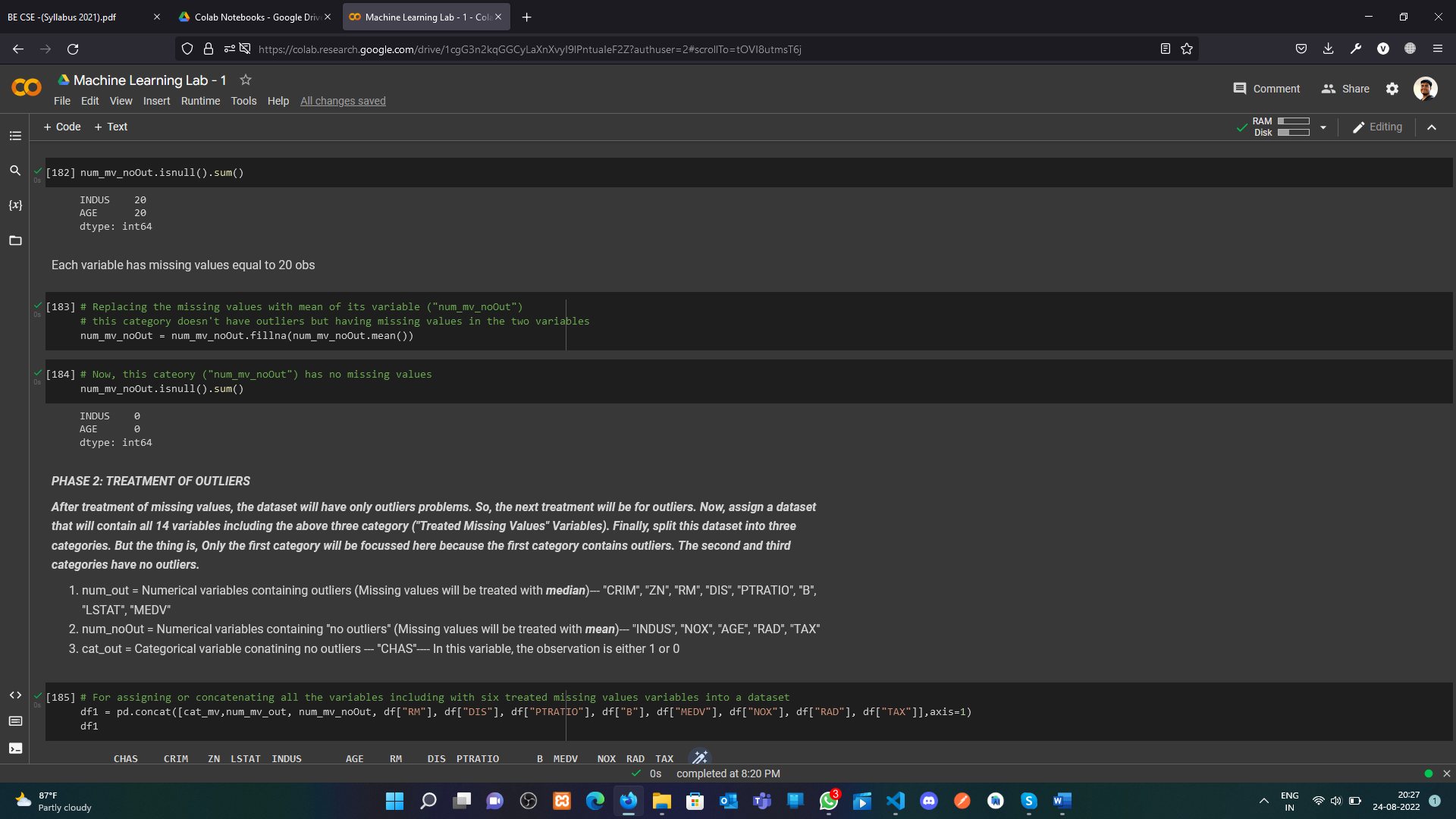


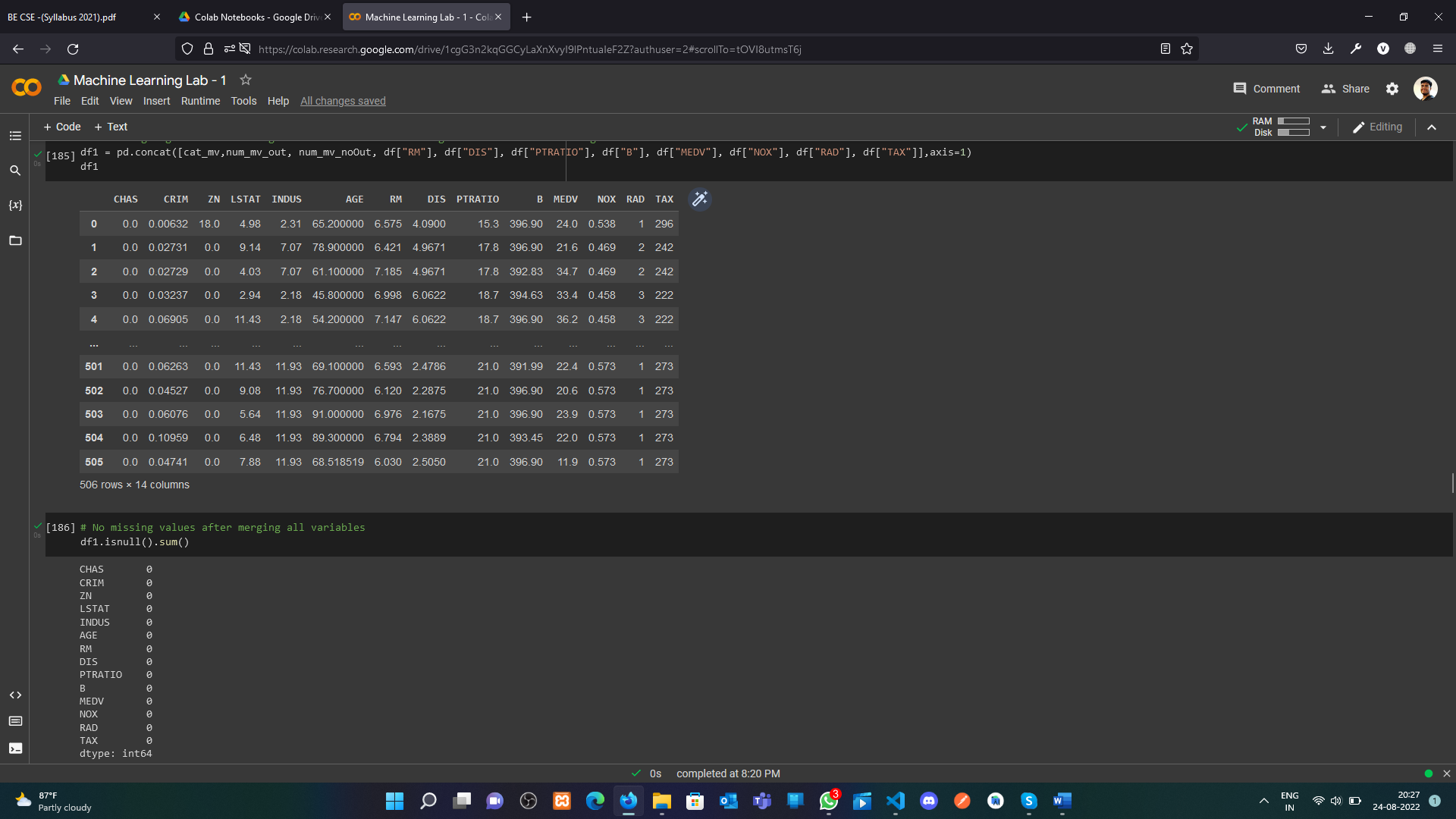




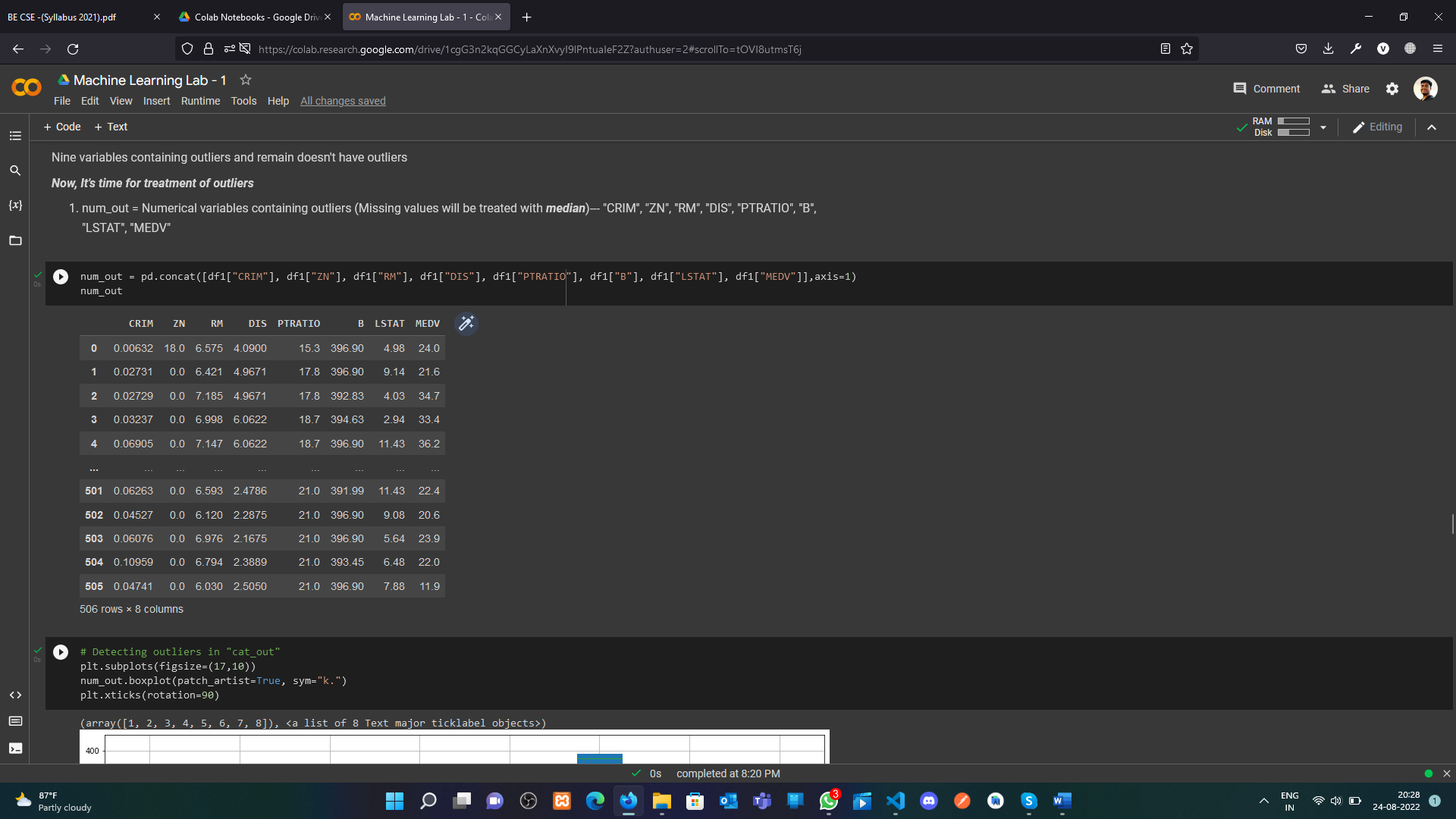


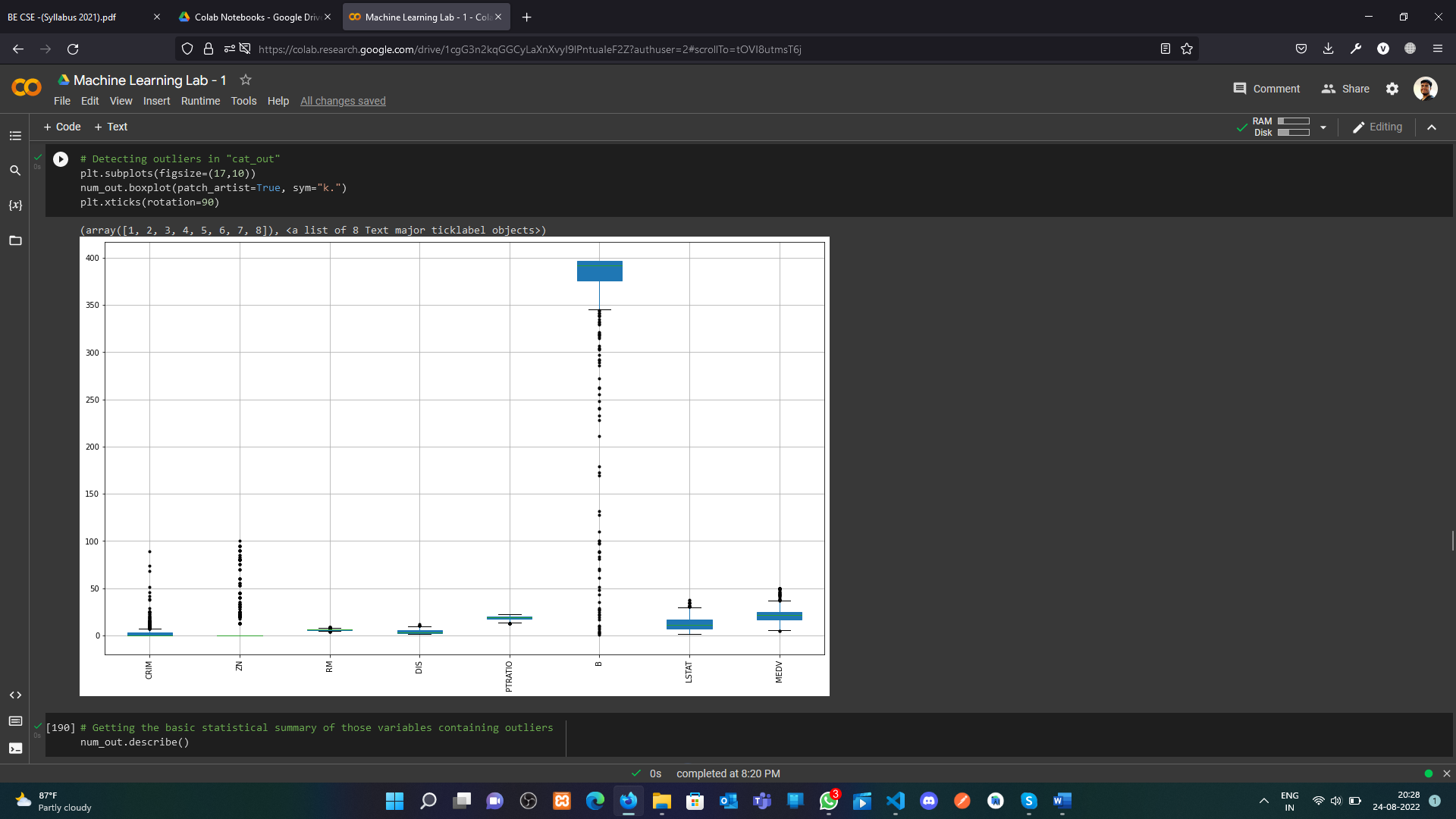


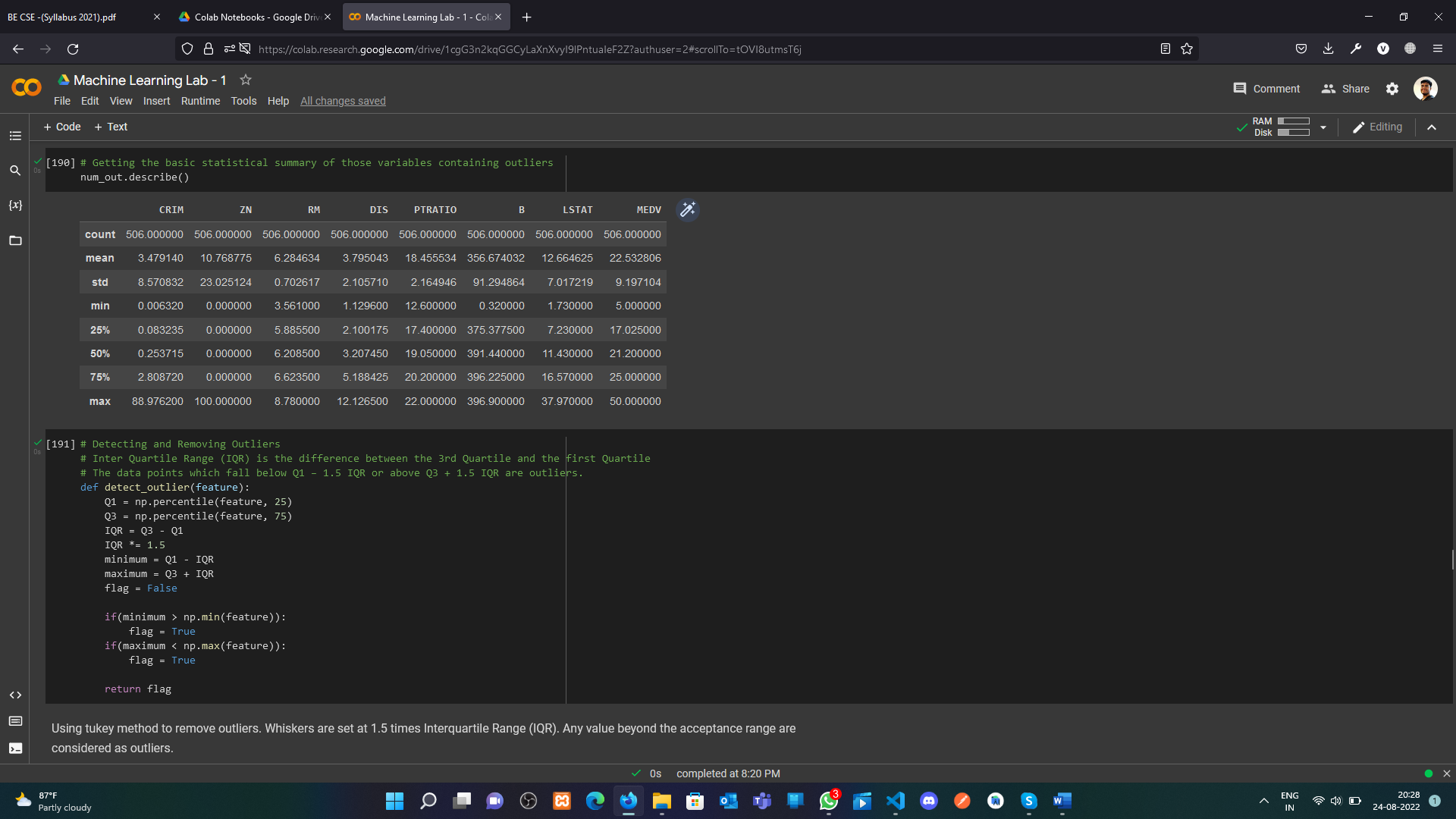




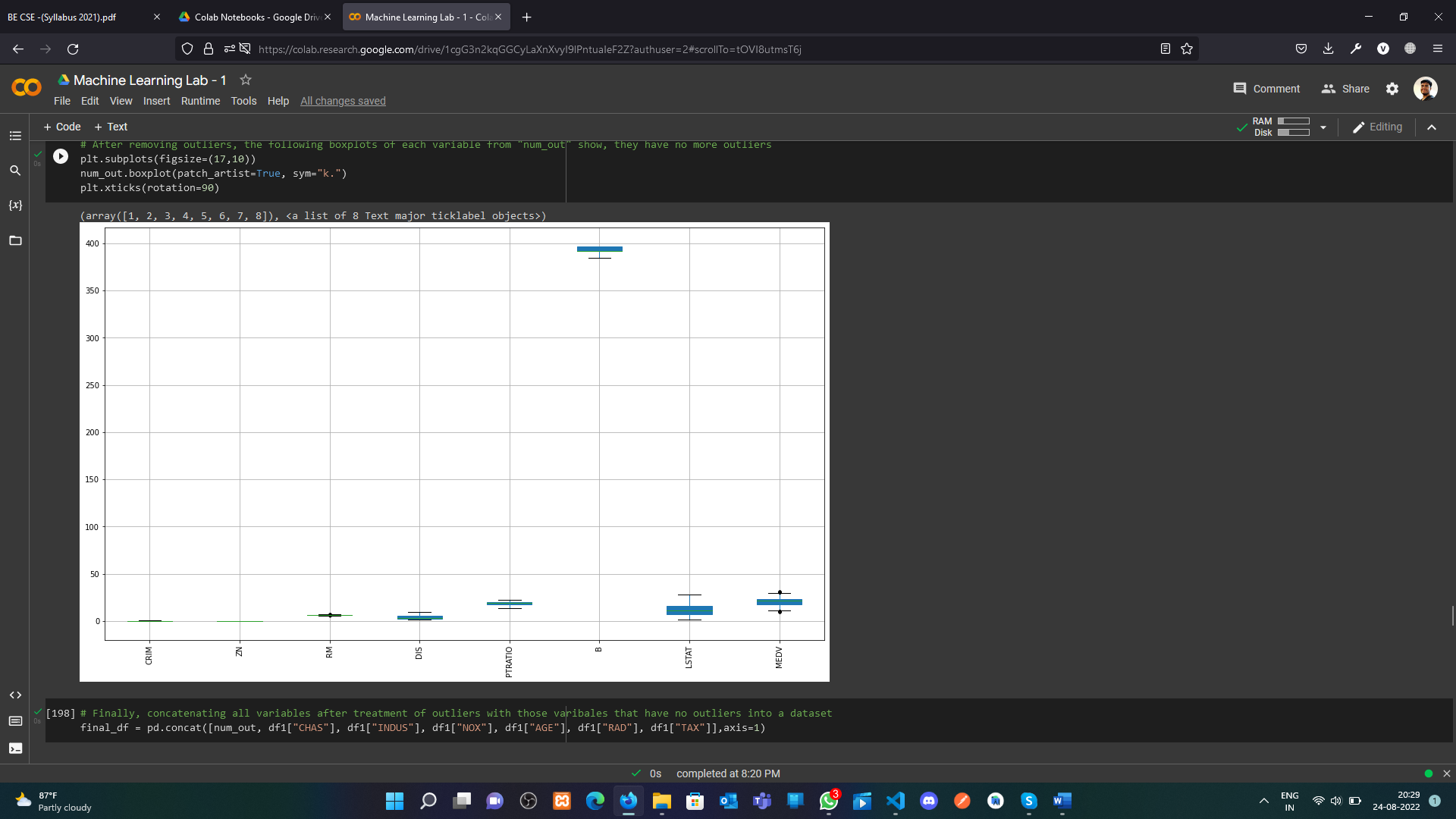
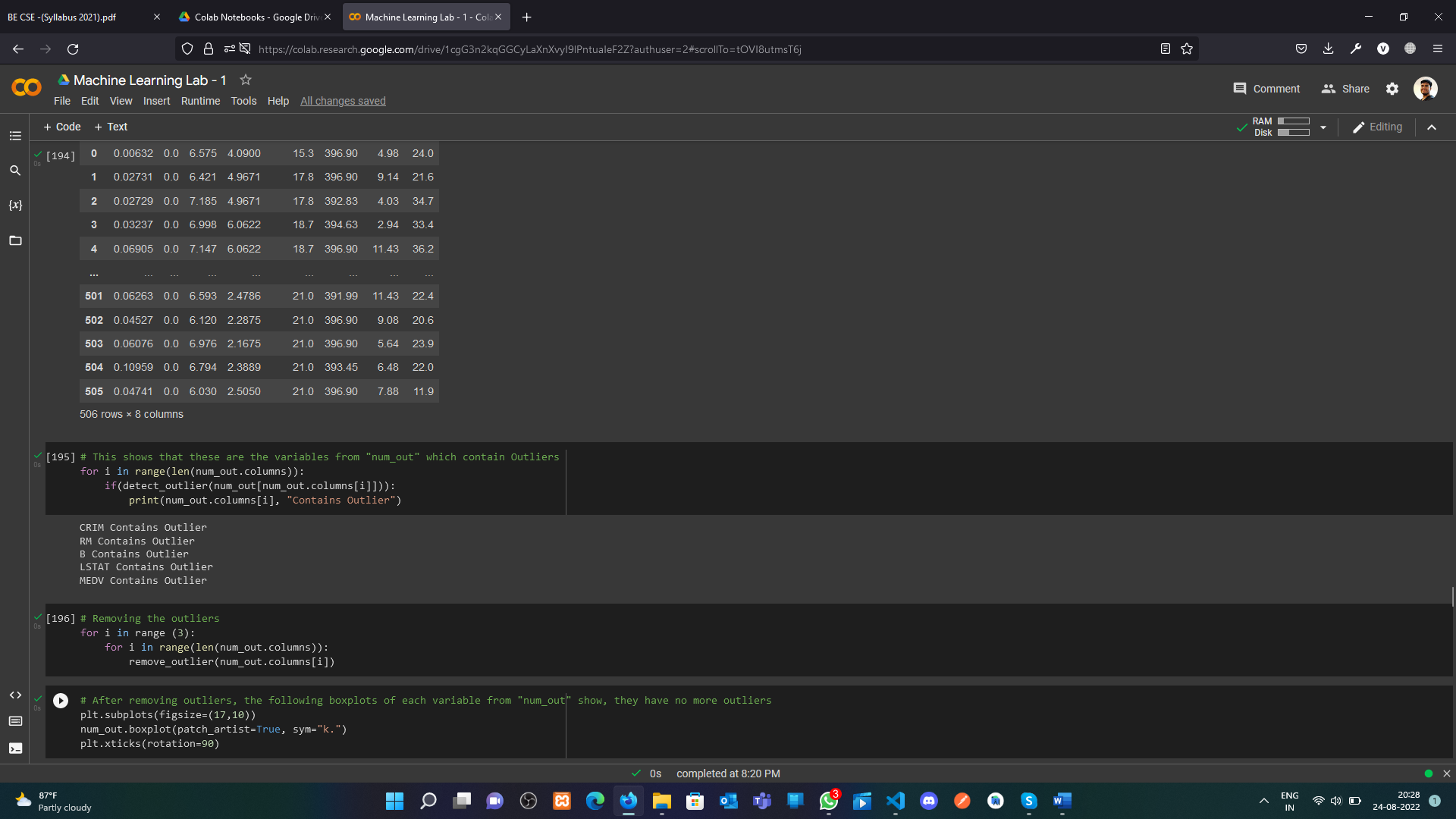


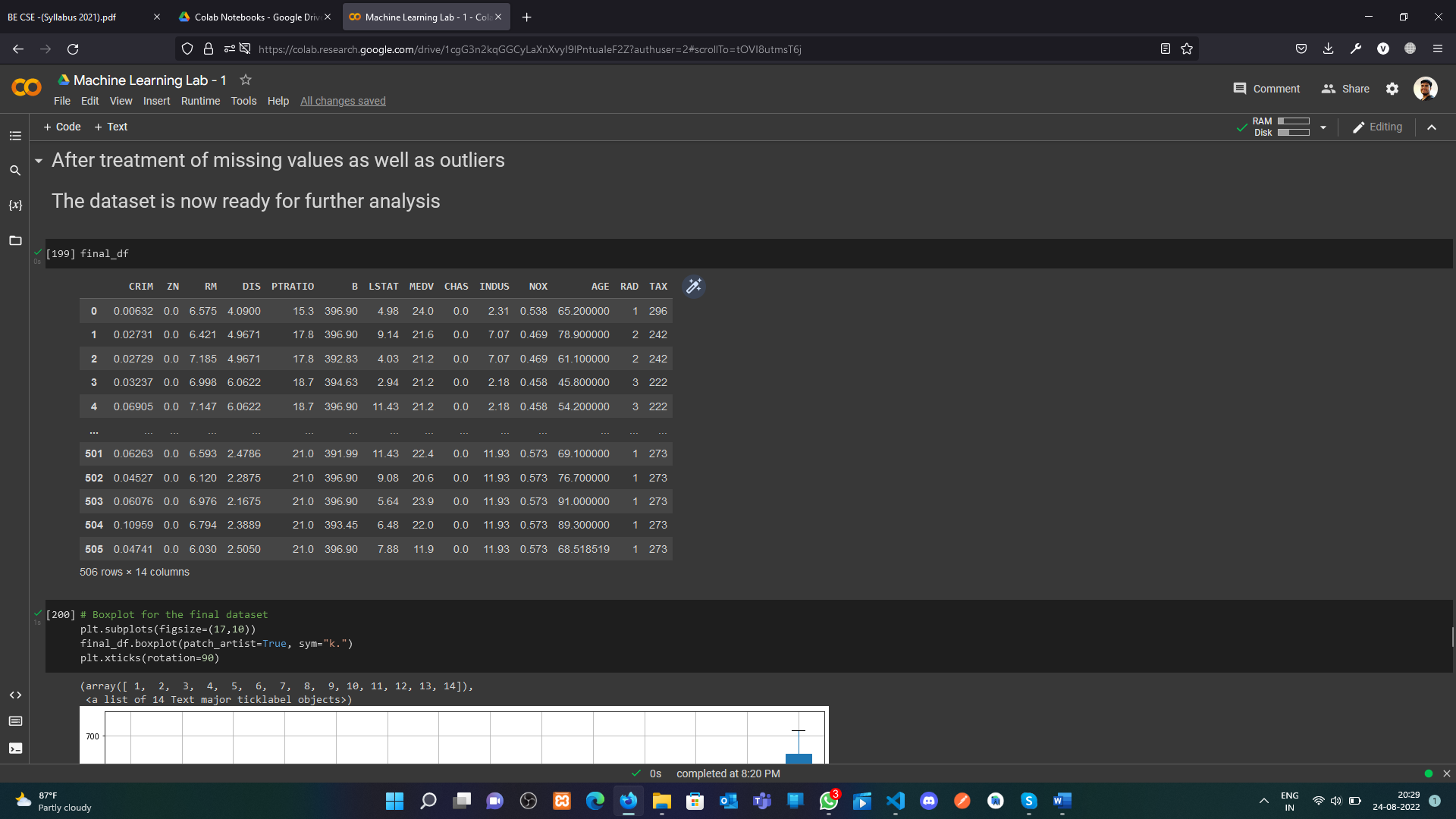


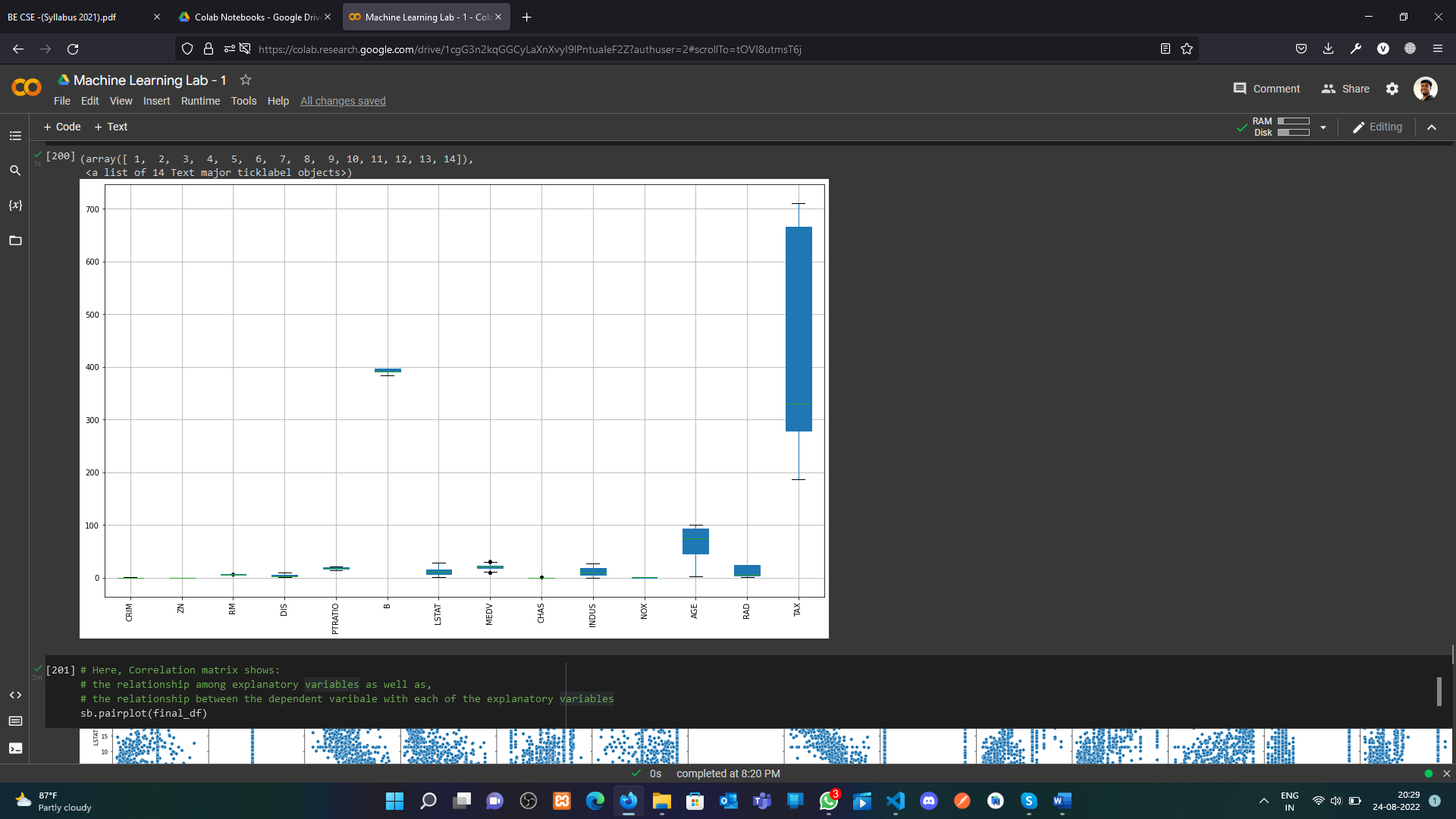




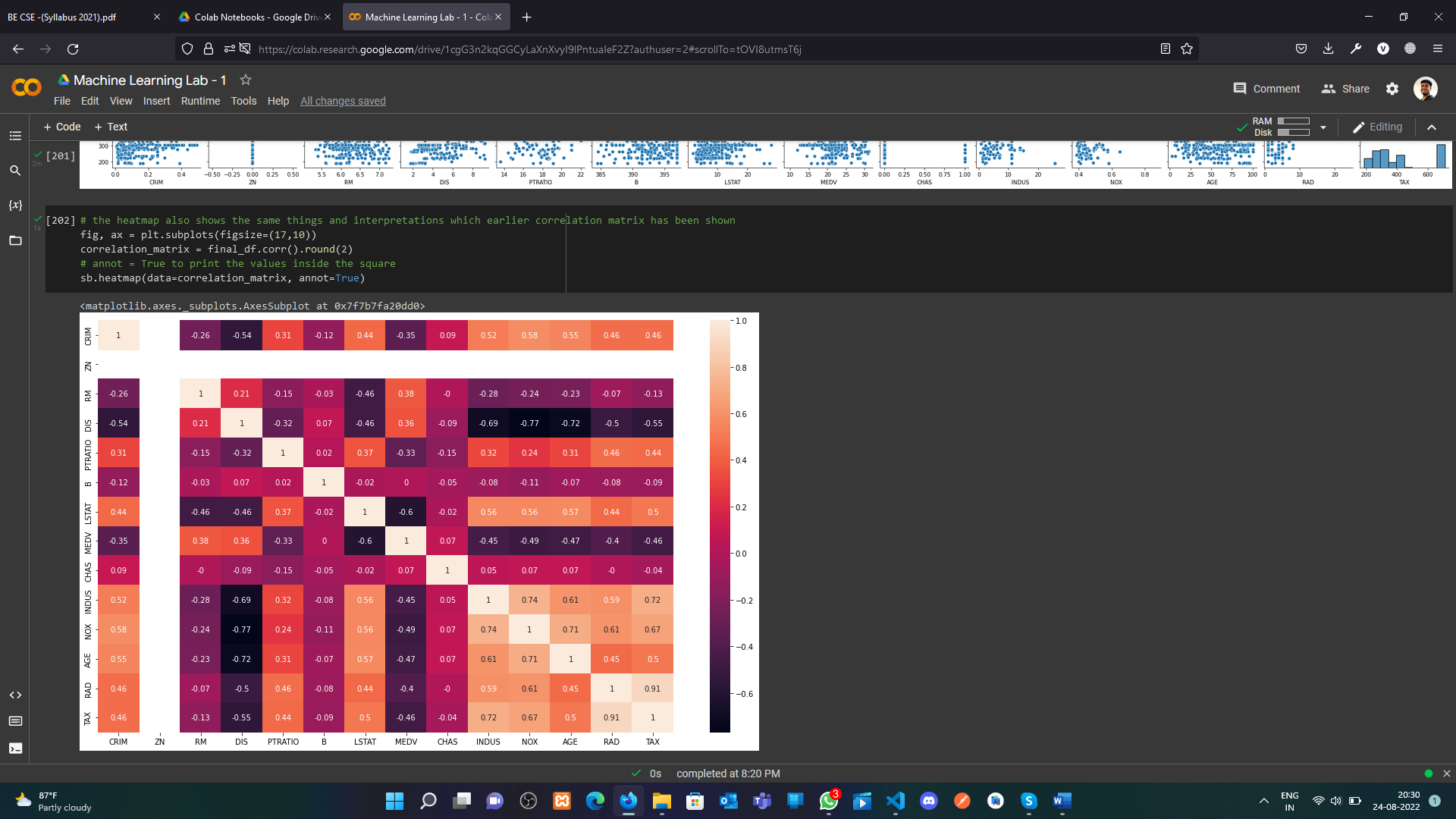


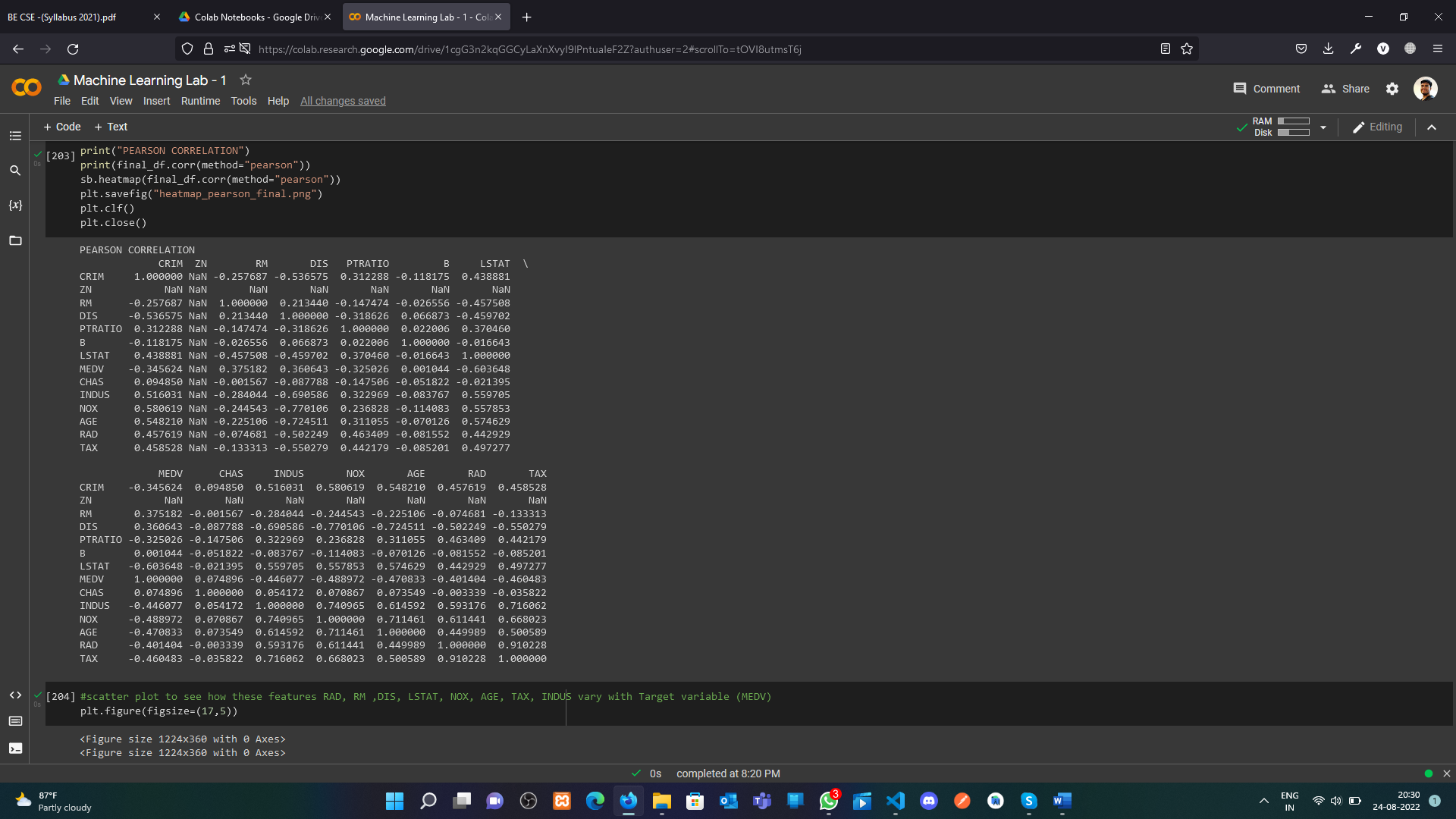


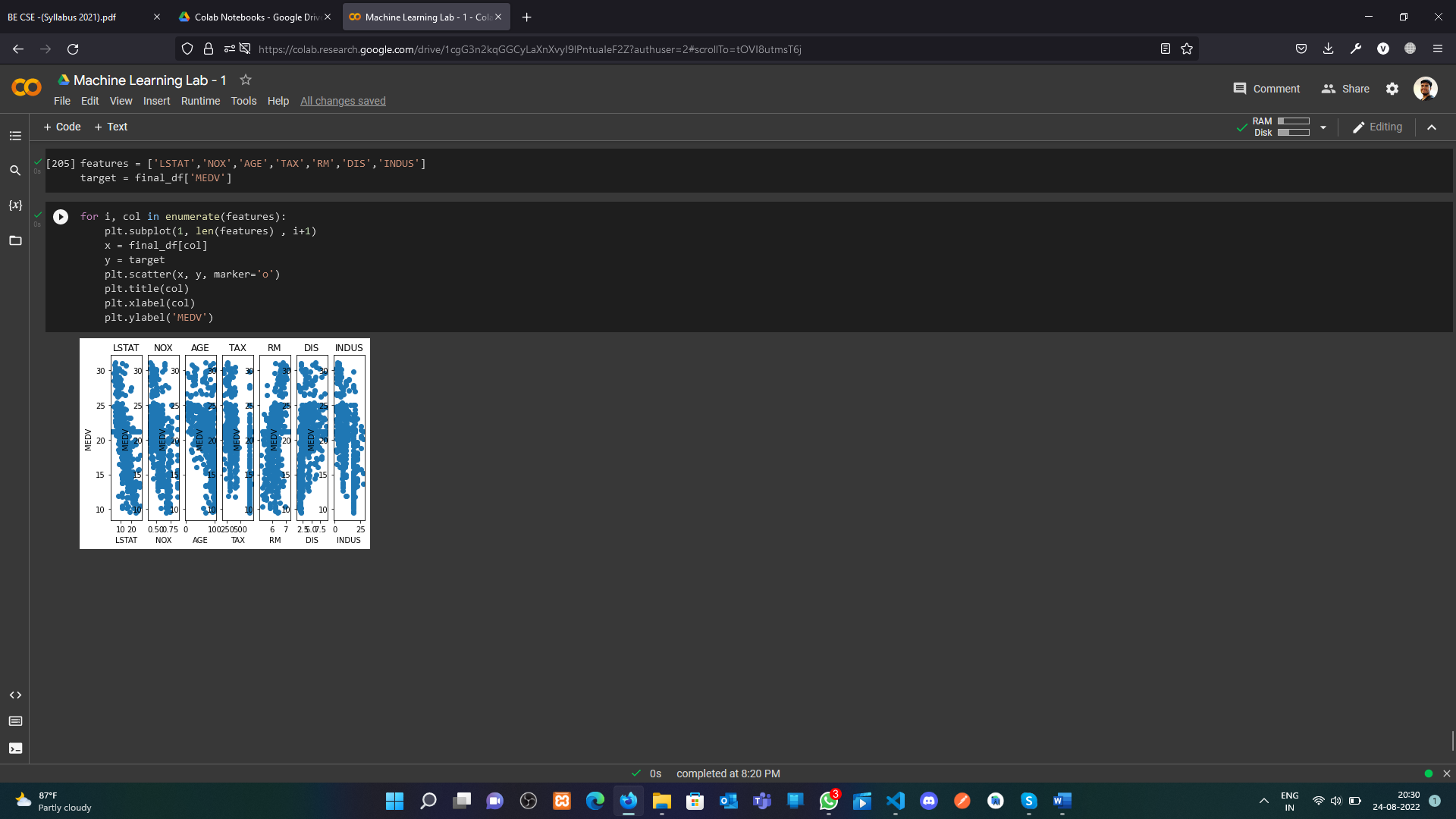




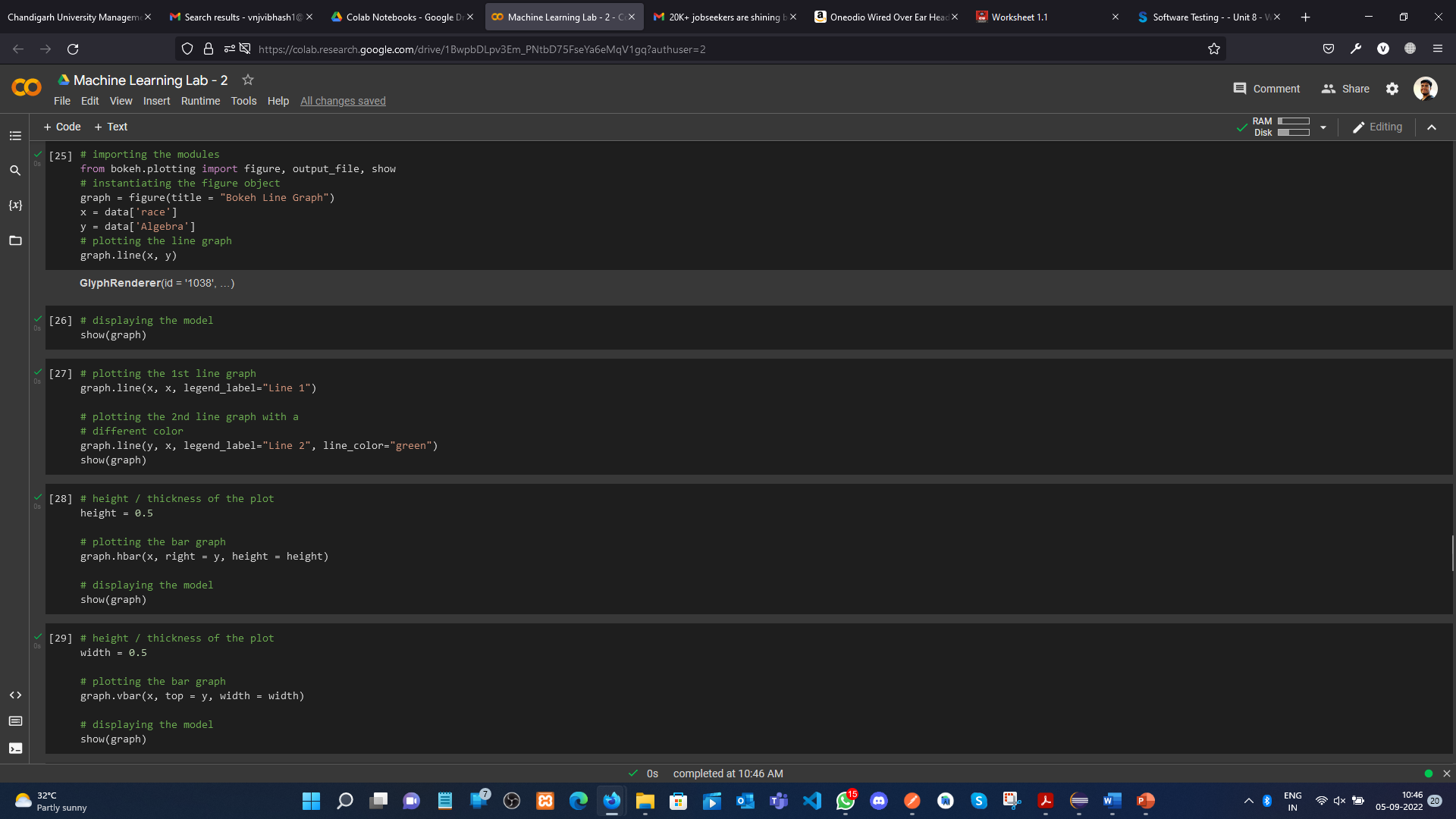








**Executed some of the Code for Bokeh Library But Didn’t get the Output.**



**Learning outcomes (What I have learnt):**

**1.** Data manipulationusing padas library

**2.** Data plotting using seaborn library

**3.** Missing values rectification

**4.** Outliers treatments

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |