**CURNEU ASSESSMENT**

**PROJECT DOCUMENTATION**

**SD03Q017**

**NAME: Priyadharshini R**

**ROLL NO: 1833039**

**Course Name: M.Sc. Decision and Computing Sciences**

**Email id:** [**priyadharshinir.my@gmail.com**](mailto:priyadharshinir.my@gmail.com)

**Problem 1: Heart Failure prediction**

**Predict Heart Failure, create a model to assess the likelihood of a death by heart failure event from scratch.**

PROBLEM STATEMENT:

Use the Machine Learning Workflow to process and transform "heart\_failure\_clinical\_records\_dataset.csv" to create prediction model. This model predicts which people have likelihood of a death by heart failure with 70% or greater accuracy.

1. **Prepare the problem:**

* **Import libraries**

The very first step is to import the all the libraries and packages required to get out expected results. Some of the necessary ML packages are Numpy, Pandas, Matplotlib, Seaborn and scikit-Learn.

*import pandas as pd*

*import os*

*import matplotlib.pyplot as plt*

*from operator import add*

*import seaborn as sns*

*%matplotlib inline*

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

*from sklearn.naive\_bayes import GaussianNB*

*from sklearn.metrics import confusion\_matrix*

*from sklearn.metrics import precision\_score*

*from sklearn.metrics import recall\_score*

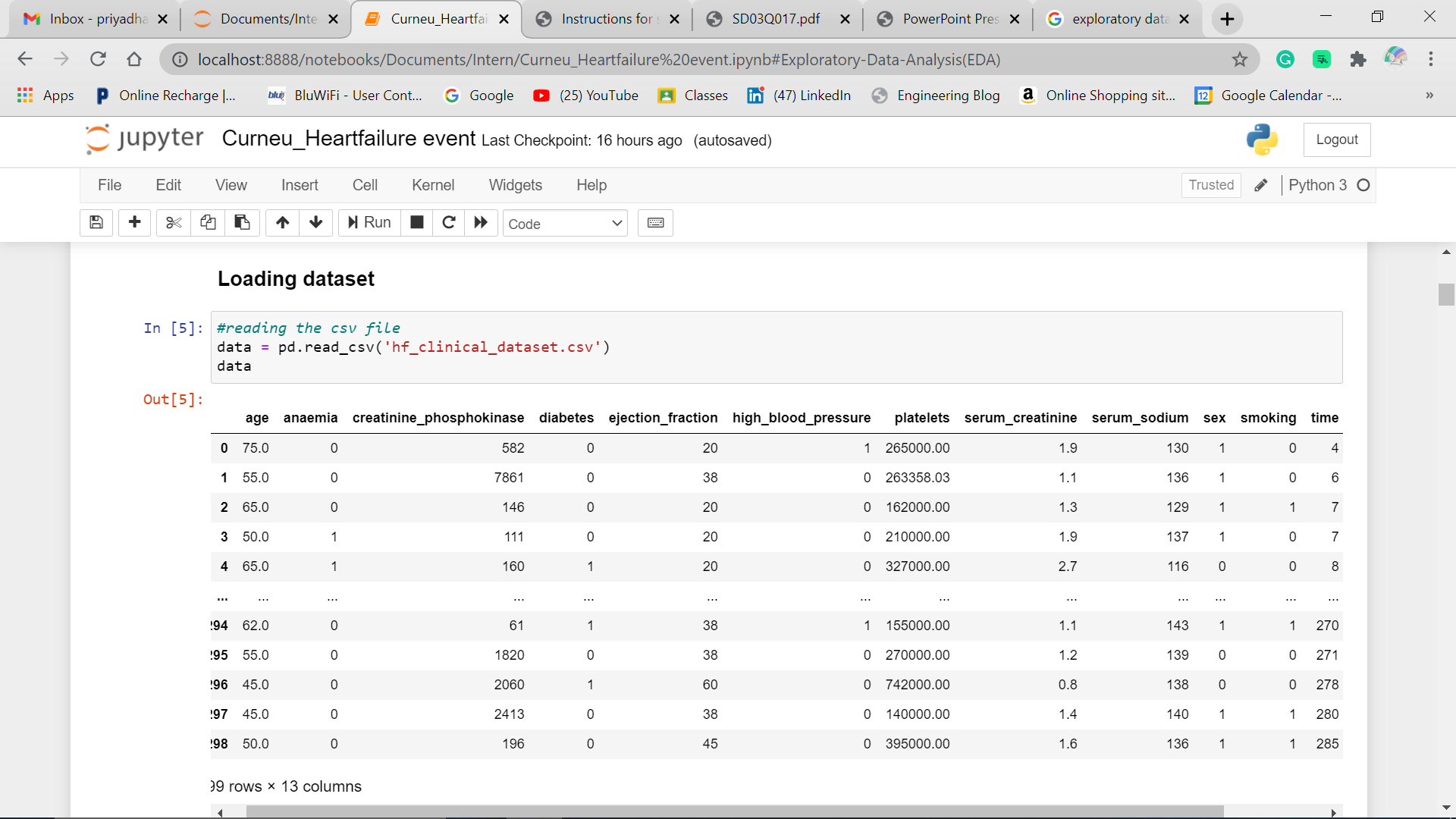
* **Load dataset**

Once the libraries are loaded, the dataset has to be loaded. Pandas is the easiest way to do this task. Not only csv files other formats like html, json, pickled files etc can also be loaded using pandas.

The dataset has to be located in the same directory as your working directory or we will need to provide the complete path prefixed with ‘/’ within the function.

*#reading the csv file*

*data = pd.read\_csv('hf\_clinical\_dataset.csv')*



1. **Summarize the given dataset:**

Once the dataset is loaded, we have to see the how many rows and columns are present, what are the data types of each column. To do this Pandas data frame can be used.

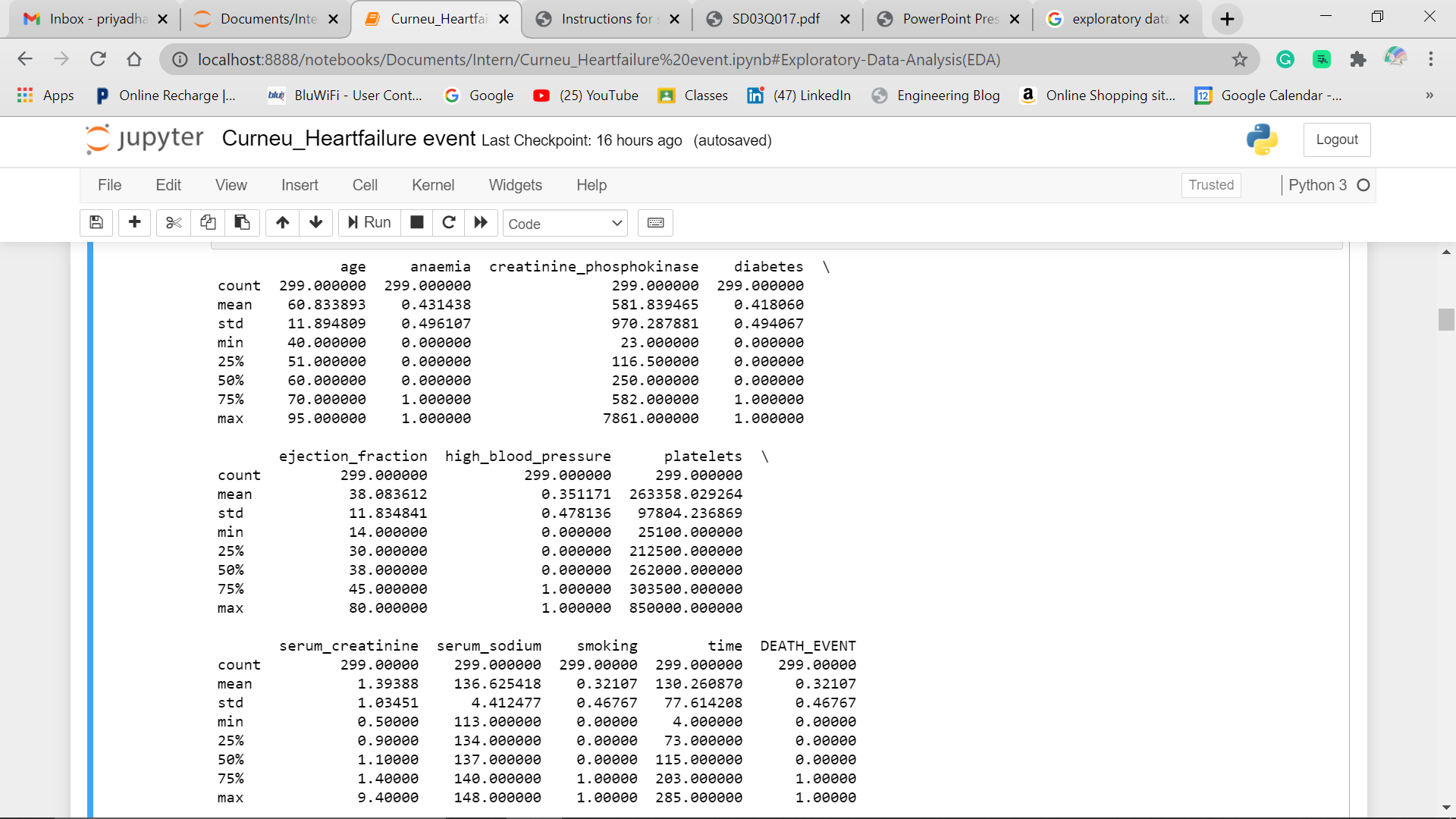
* **Descriptive statistics:**

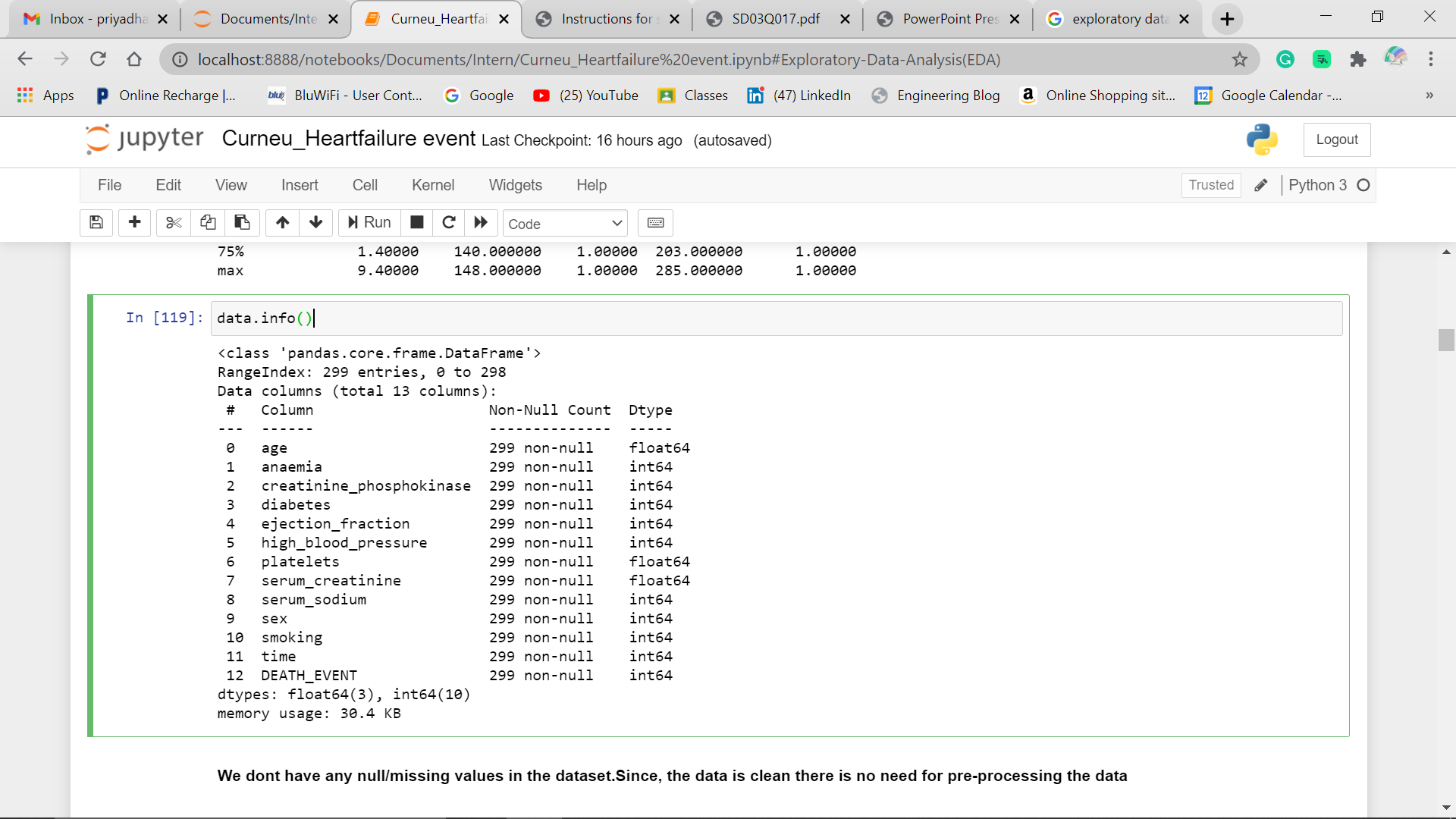
This describes the statistics like mean, media, percentiles etc, for each column in the dataset.

And the info() function gives us the values , their types in the dataset and presence of missing values can also be identified correctly.

*print(data.describe())*

*data.info()*





1. **Explanatory DataAnalysis(EDA)**

* EDA is the process of analysing and investigating the datasets and summarise their main characteristics, often by data visualisation methods
* Here, we will be analysing every attribute in dataset, along with shape of dataset. Correlation between the target and other attributes also analysed.
* **Understanding Columns:**

*info = [" age",*

*"1:Yes, 0:No",*

*"creatinine\_phos",*

*"1: Yes, 0: No",*

*"ejection\_fraction",*

*"1: Yes, 0: No",*

*"platelets","serum\_creatinine",*

*"serum\_sodium",*

*" 1:male, 0: female",*

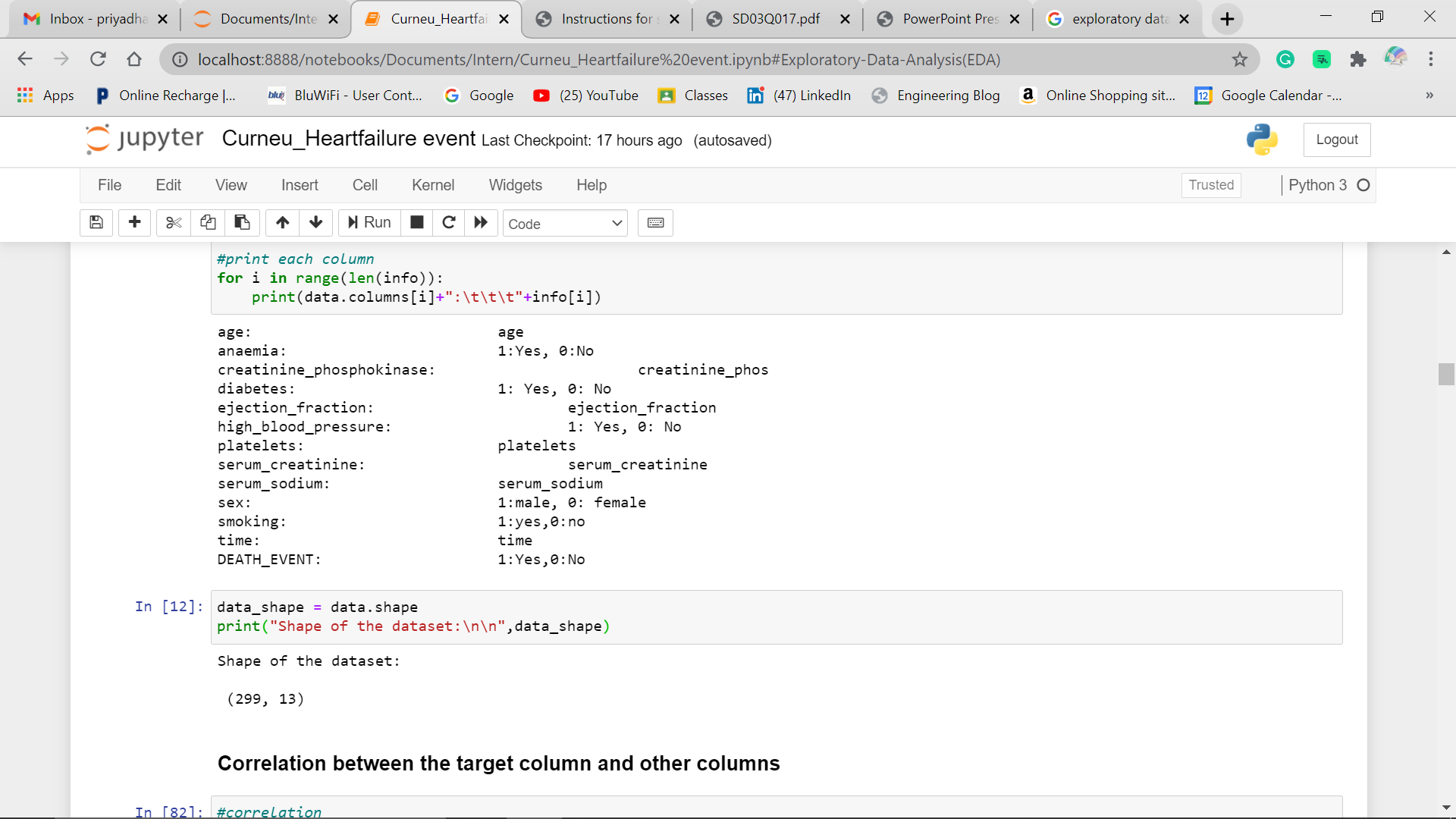
*"1:yes,0:no",*

*" time","1:Yes,0:No"]*

*#print each column*

*for i in range(len(info)):*

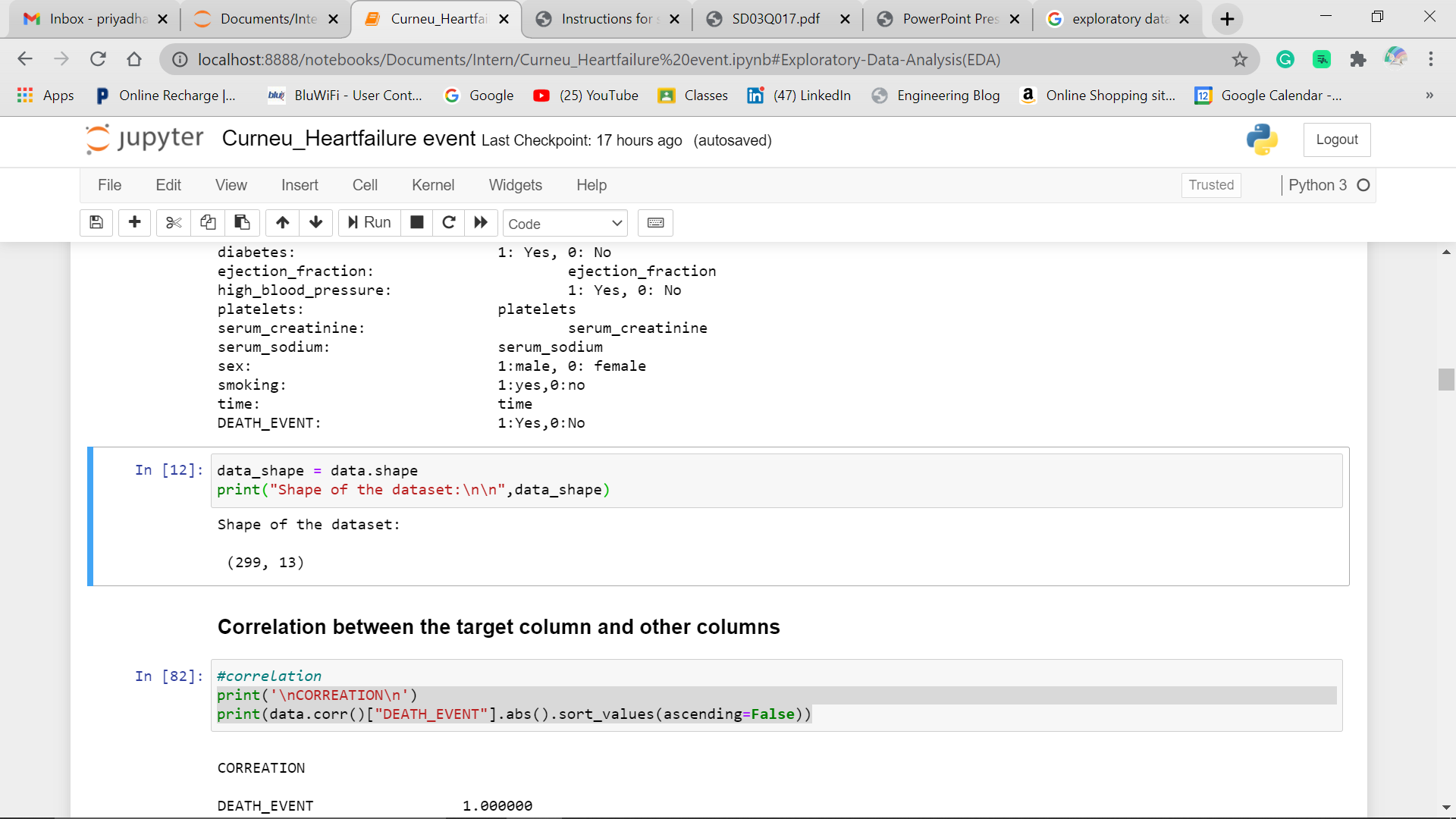
*print(data.columns[i]+":\t\t\t"+info[i])*



* **Shape of dataset:**

*data\_shape = data.shape*

*print("Shape of the dataset:\n\n",data\_shape)*



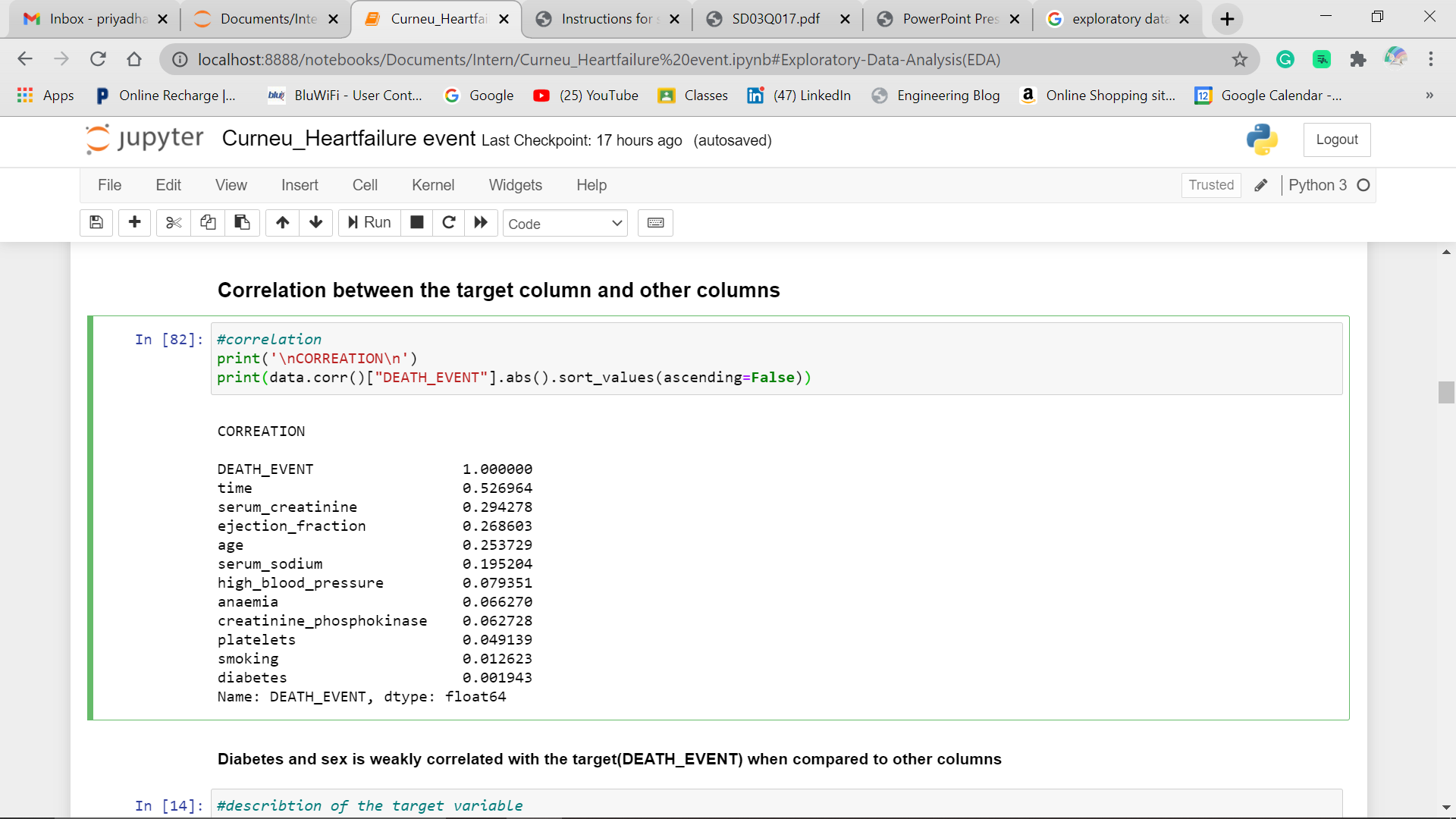
* **Correlation between the target and other attributes also analysed**

Correlation is defined as the relationship between 2 variables. It ranges from -1 to +1.

* 0 – no correlation
* +1 – positive correlation (Highly)
* -1 - negative correlation (Highly)

*print('\nCORREATION\n')*

*print(data.corr()["DEATH\_EVENT"].abs().sort\_values(ascending=False))*



1. **Data Visualisation:**

* Data visualisation are very important as they are quickest way to understand the data patterns.
* Visualisation using Matplotlib, Seaborn can be used to check the correlation within the features and the target.
* Histograms for checking the spread and skewness
* Seaborn to quickly plot the visulaisation of the whole data to check multicollinearity etc.
* **Data distribution**
* This helps us to understand the distribution of dataset for each attribute and plots give clear visualisation.
* Here, matplotlib and seaborn libraries are used to plot the graph(histogram)

#*data distributions*

*import matplotlib.pyplot as plt*

*from operator import add*

*import seaborn as sns*

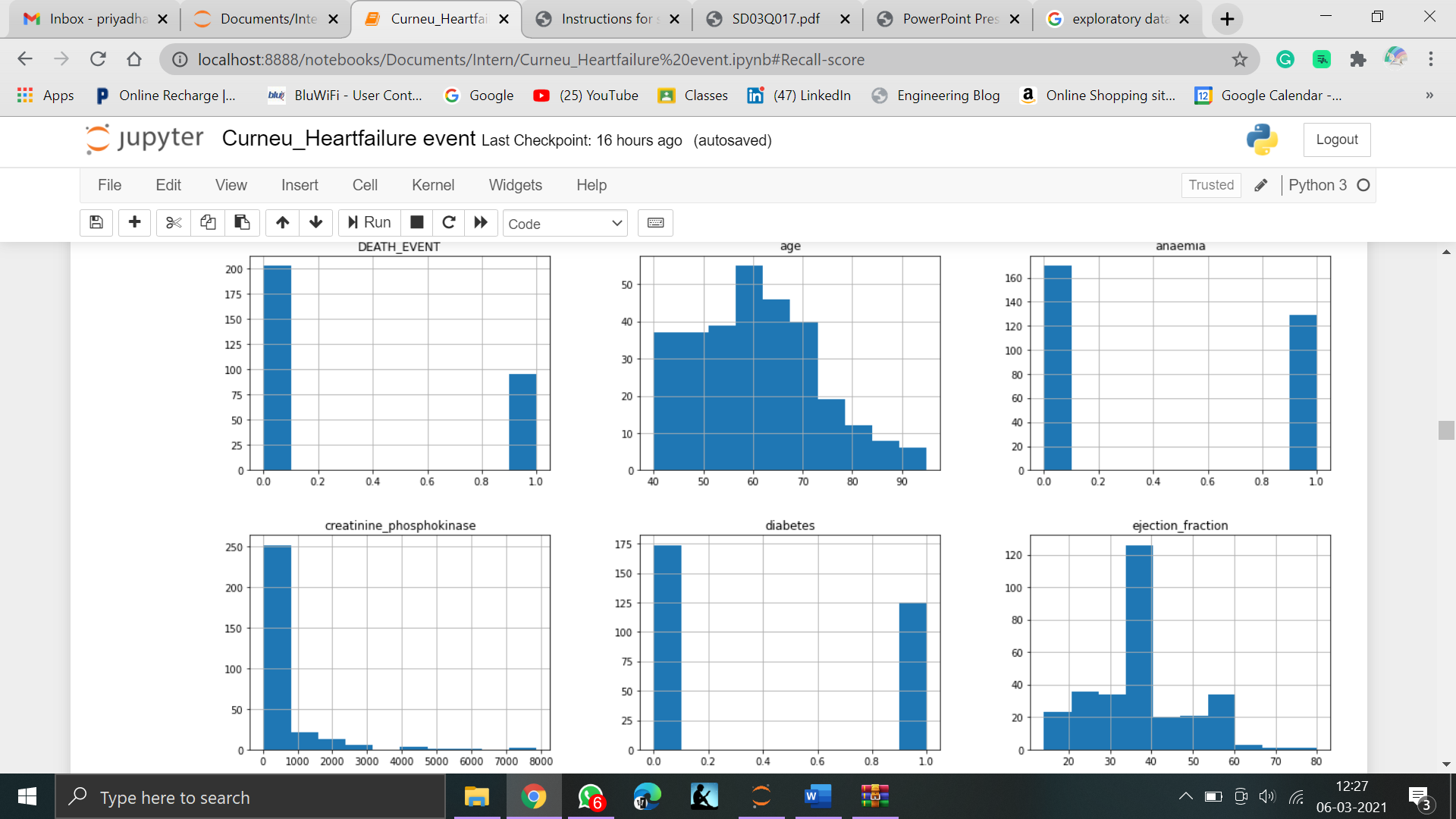
*%matplotlib inline*

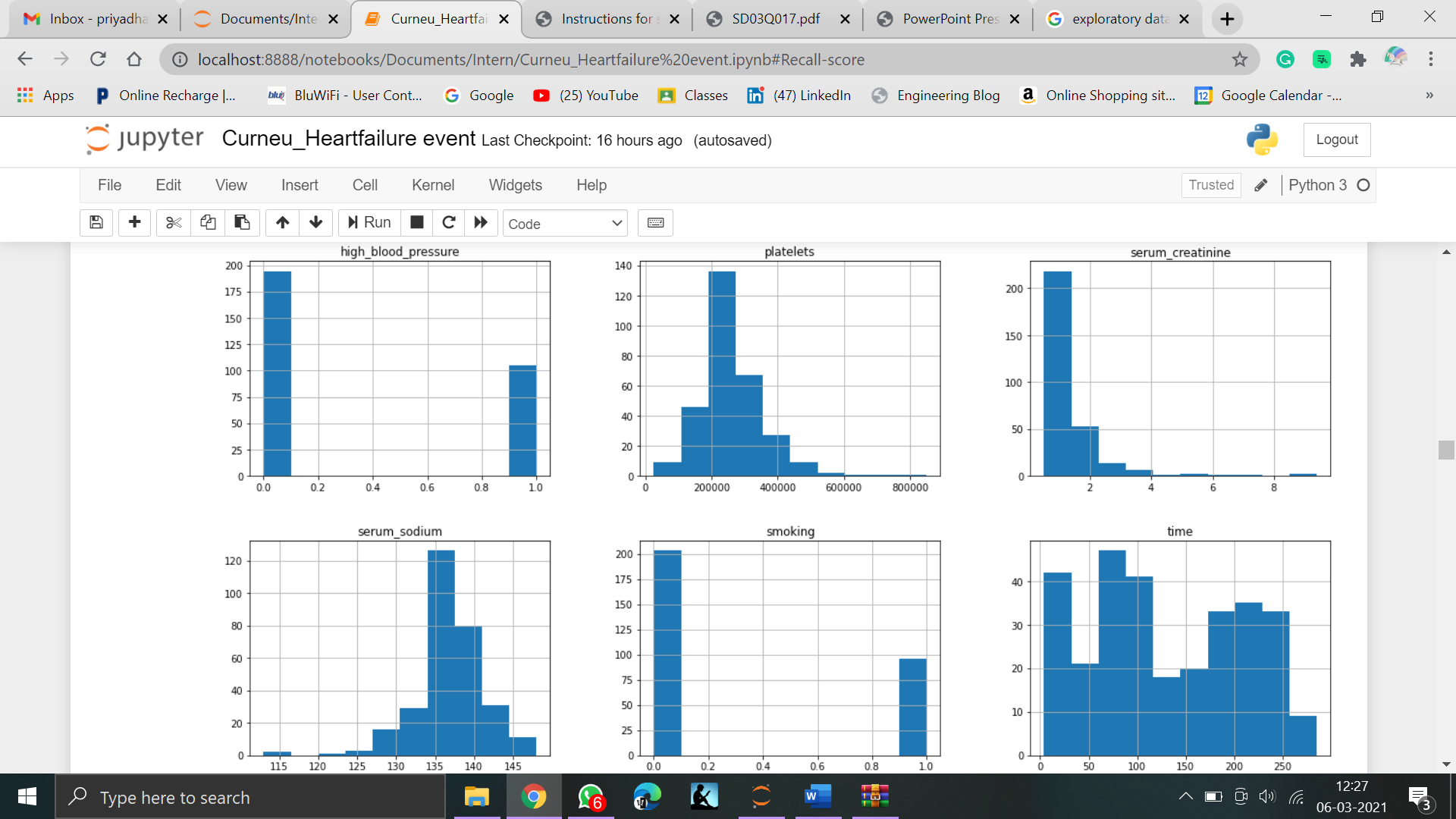
*fig = plt.figure(figsize = (20,20))*

*ax = fig.gca()*

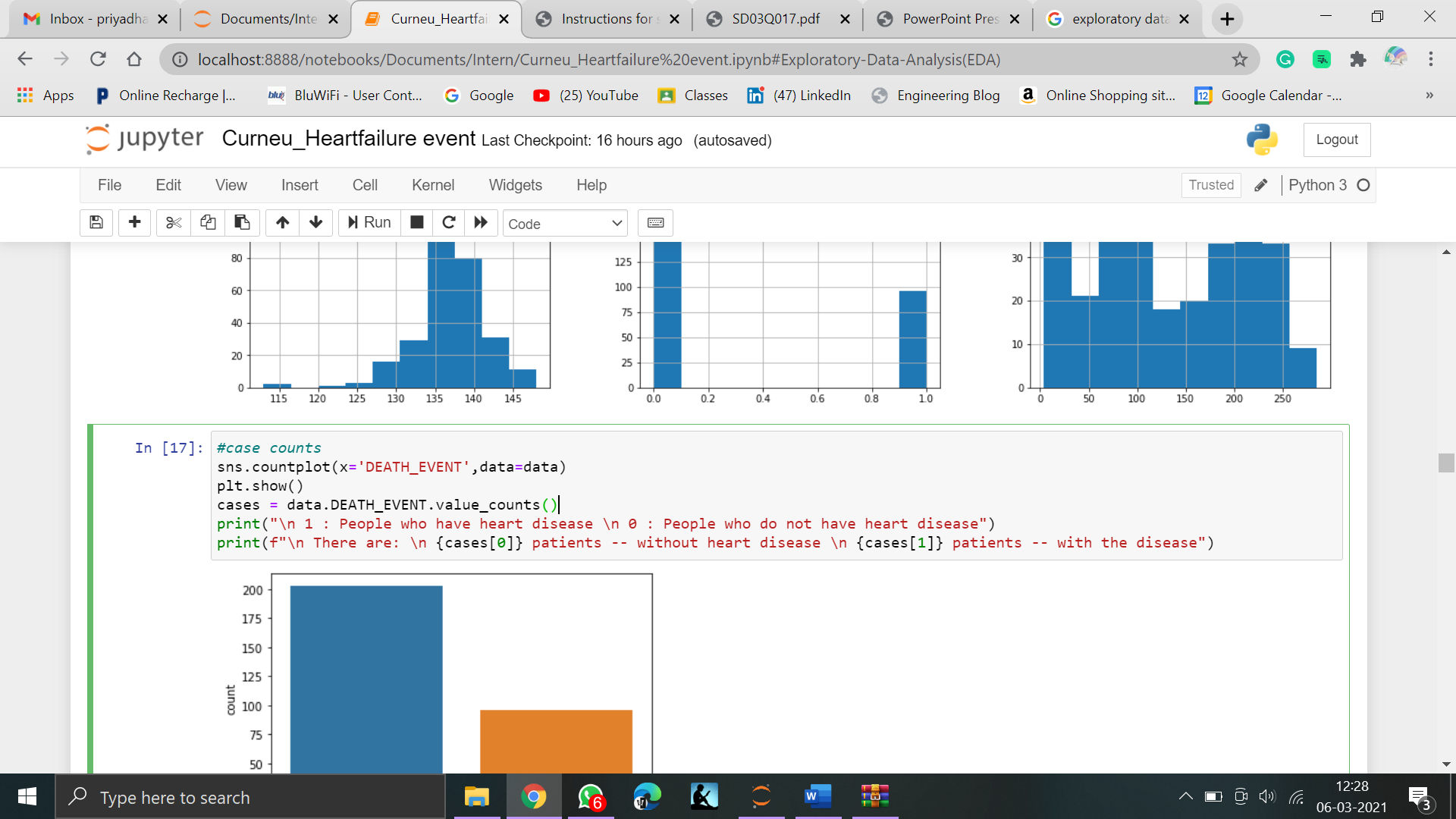
*data.hist(ax = ax)*

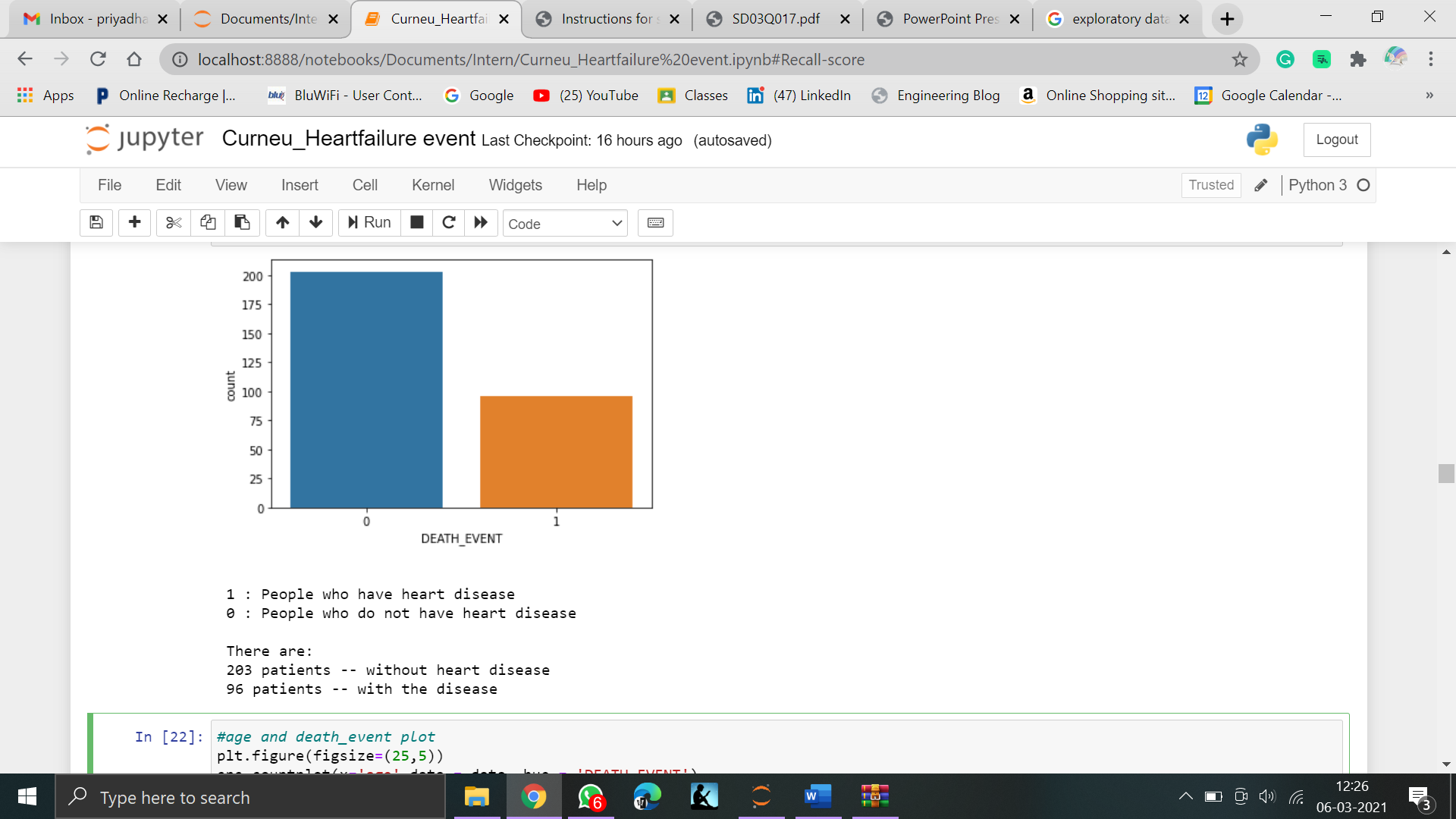
*plt.show()*



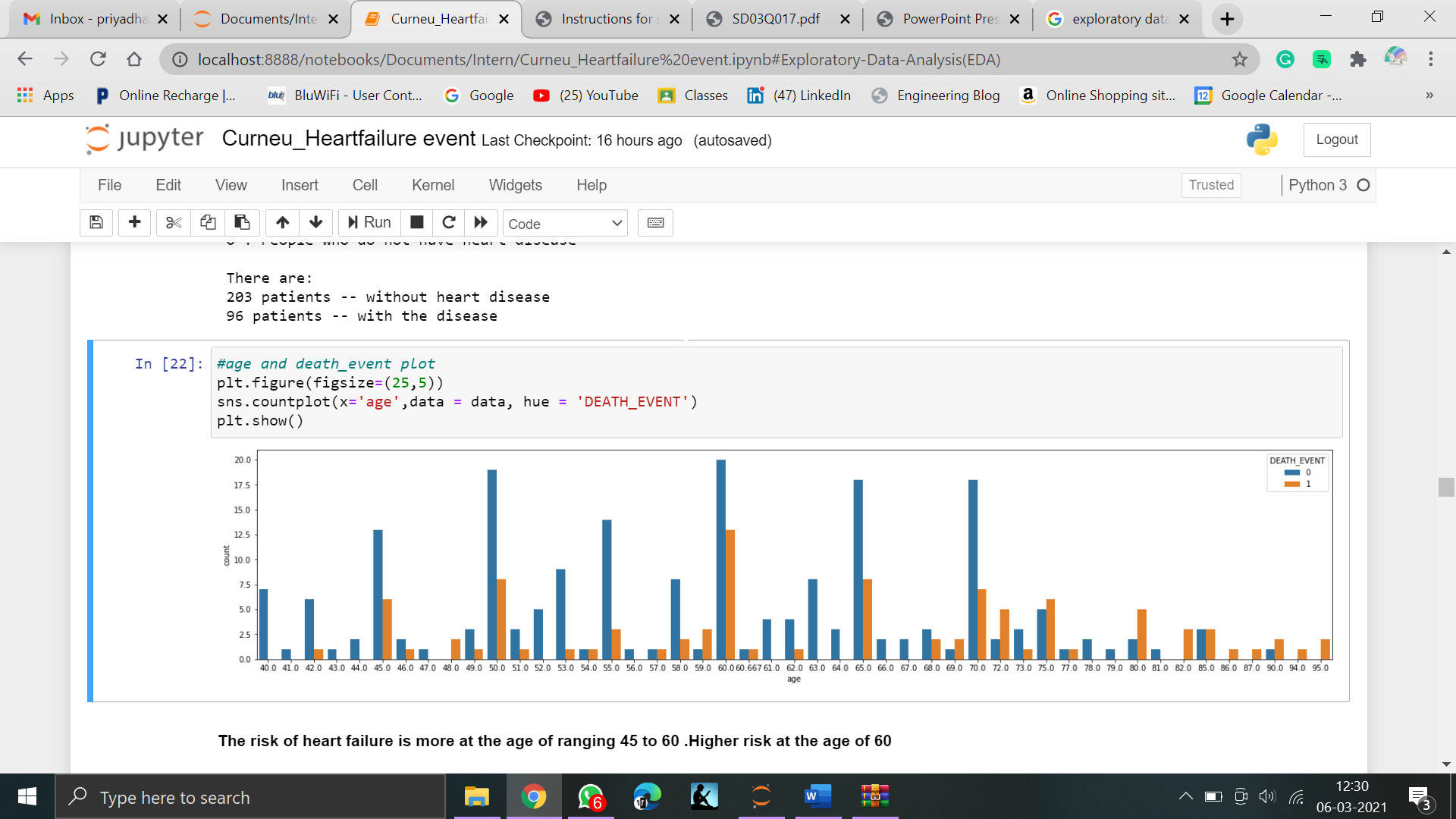


* **Case counts:**
* This helps to analyse the death COUNT from the dataset.
* 1 represents people with heart failure
* 0 represents people without heart failure
* 203 patients without heart disease
* 96 patients with heart disease

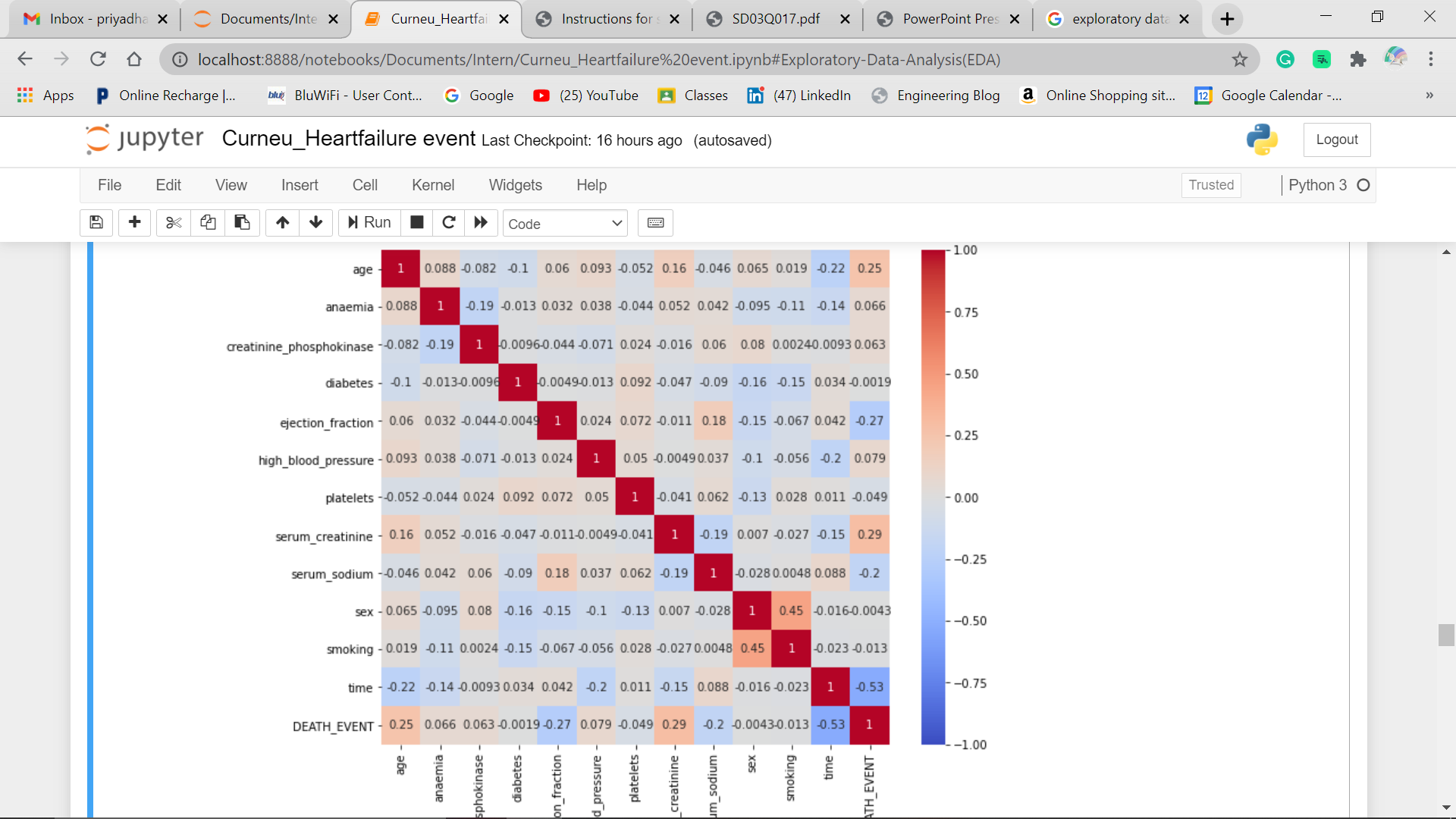




* **Age and Death event plot:**
* This helps us to visualise that which age group has high risk of heart failure\
* From the graph, it is evident that risk of heart failure at the age of ranging 45 to 60. And higher risk the age of 60

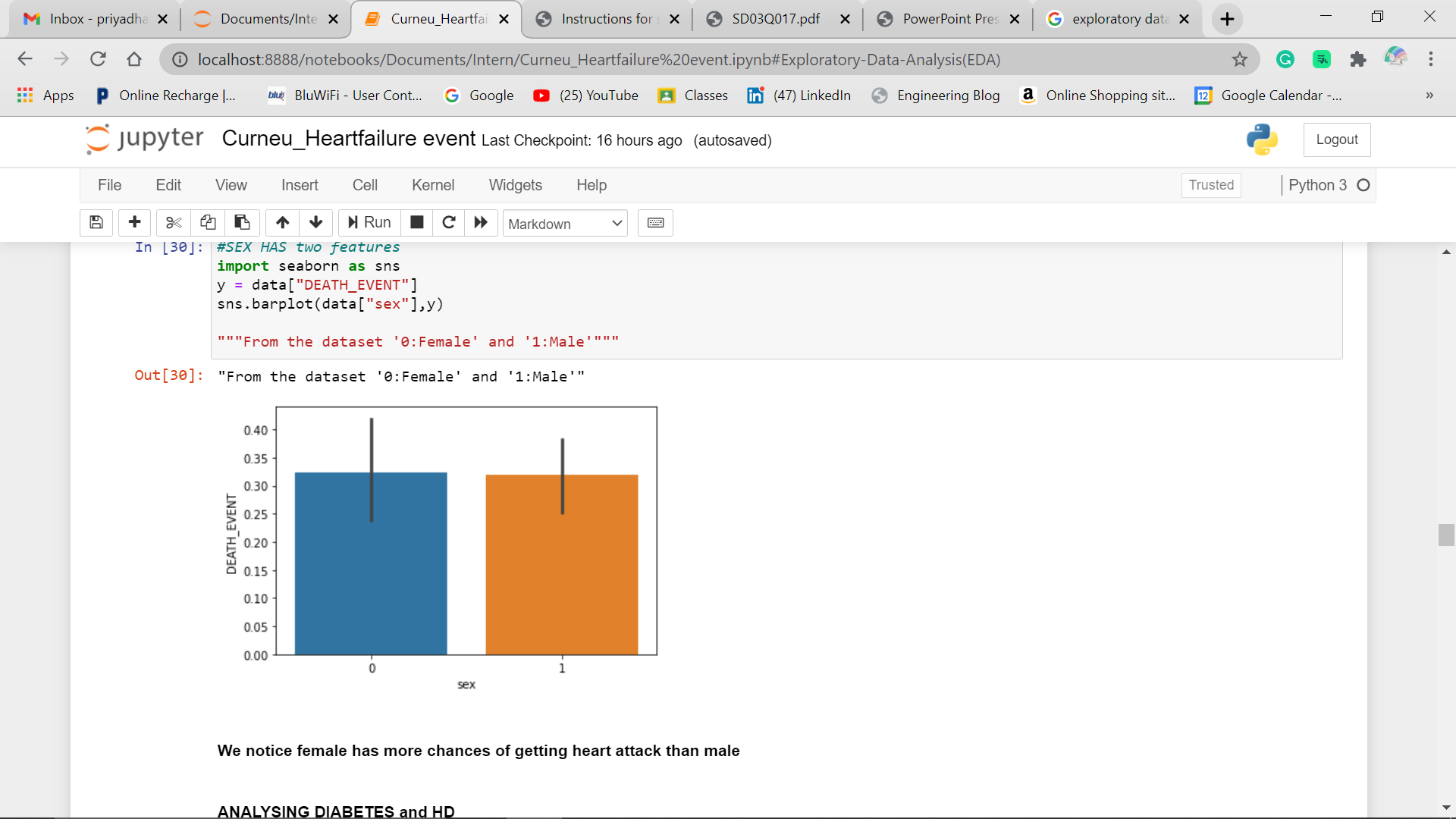


* **Correlation heat map:**
* Seaborn library is used to plot Heatmap
* Correlation ranges from -1 to +1. Larger the number and darker the colour, the correlation is higher between variable
* 0 - no correlation
* +1 - positive correlation
* -1 - negative correlation

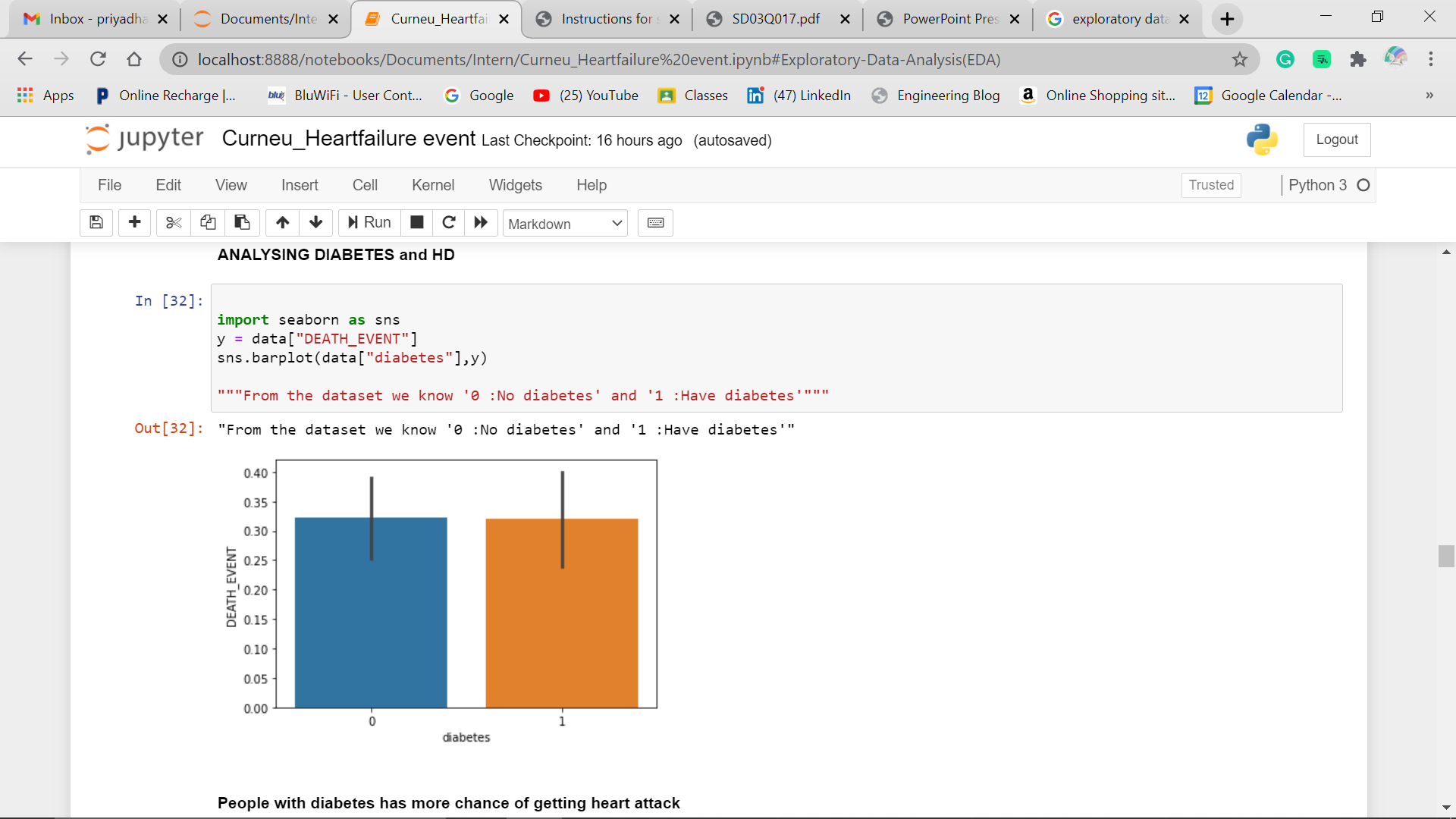


* **Analysing sex and HD**

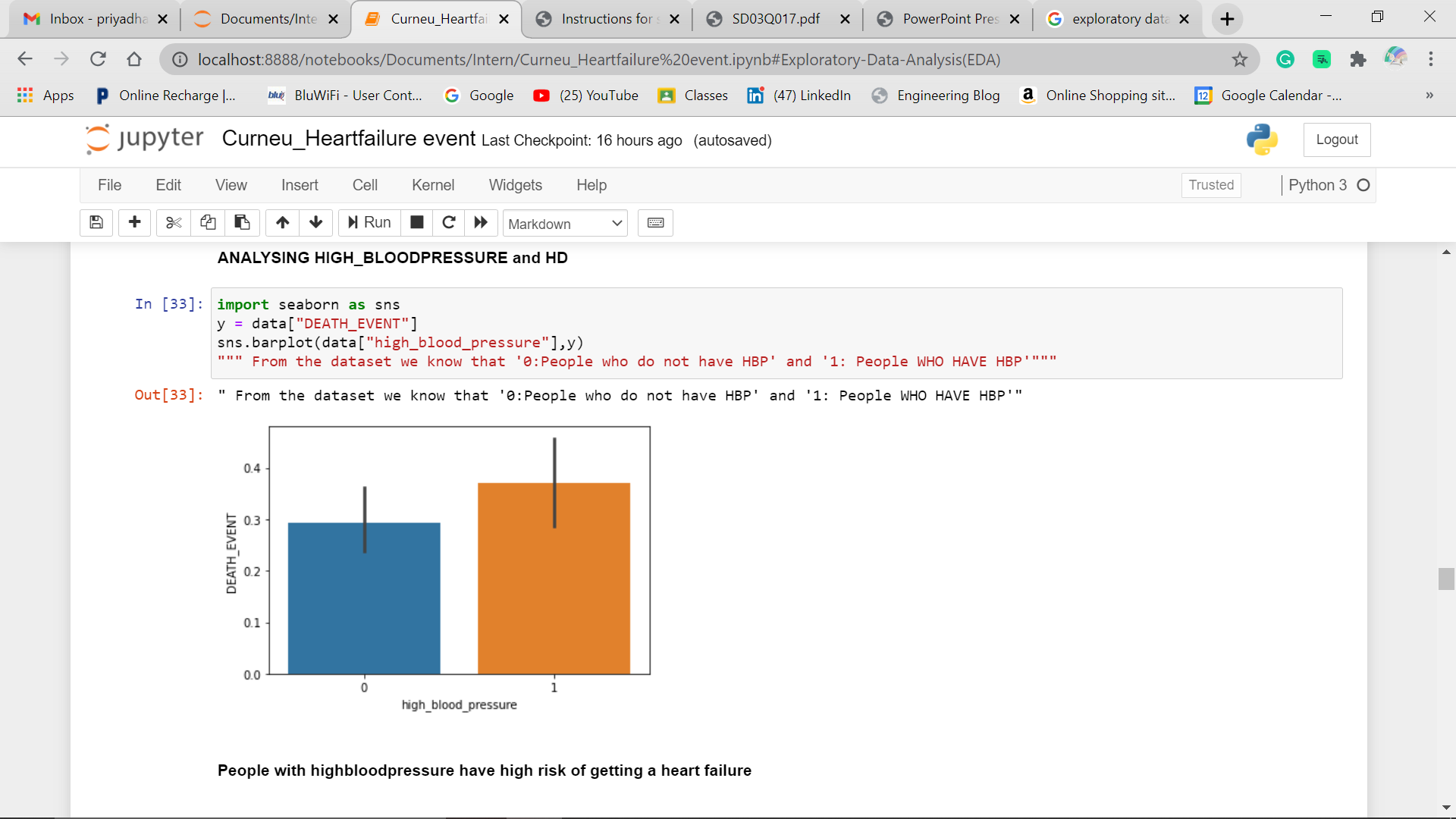




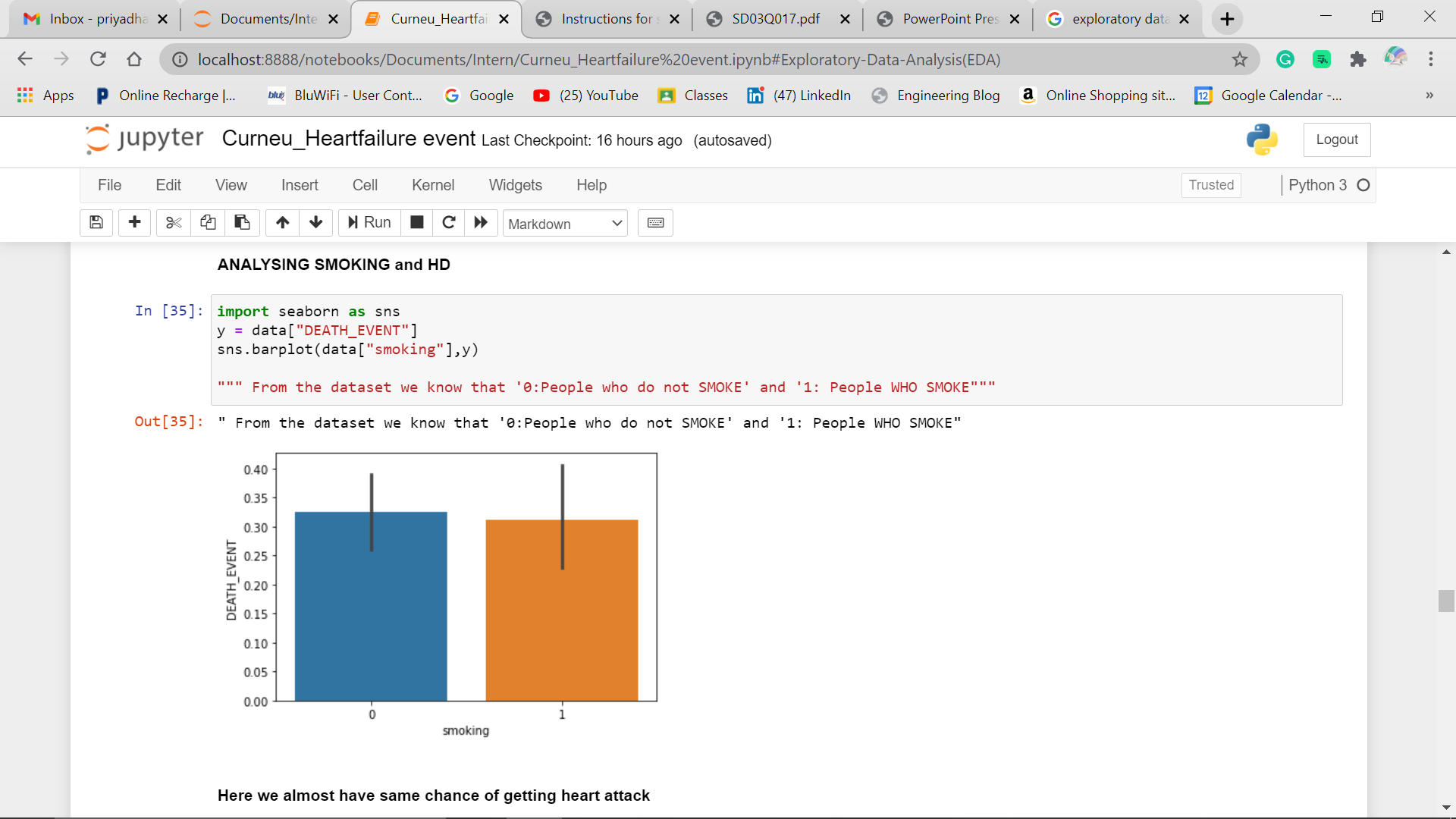
* **Analysing diabetes and HD**



* **Analysing HBP and HD**



* **Analysing smoking and hd**



From the above plots the risk factors of Heart Failure are:

* High Blood Pressure
* Diabetes

1. **Prepare data:**

Once we understand the data, what it has and what it looks like, we will have to transform the data in order to make it suitable for algorithm process and work more efficiently in order to make accurate results

* **DATA CLEANING:**

Since the dataset has no missing or null values. We can proceed with next step

* **FEATURE SELECTION:**

Feature selection is the process of selecting a certain number of most useful features which will be used to train the model. Reducing features saves lot of time.

1. **Evaluate Algorithms:**

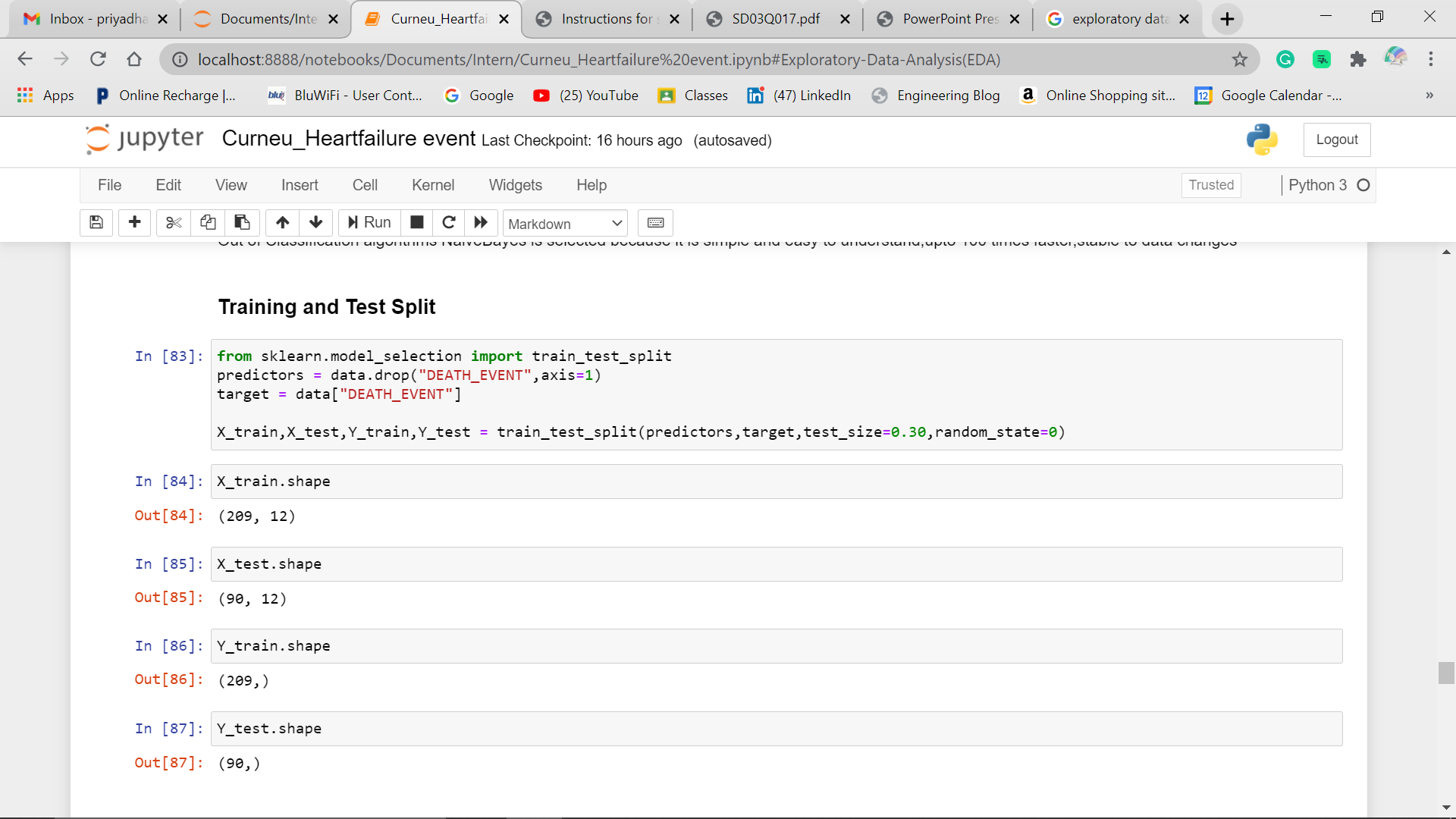
Once our data is pre-processed, we can process to check the performance of various regression/classification algorithm based on the problem statement.

**ALGORITHM SELECTION:**

* Since the problem statement states we have to predict the death by heart failure, Supervised machine learning algorithm can be used.
* The dataset has discrete value, so classification algorithm is selected.
* Out of Classification algorithms Naïve Bayes is selected because it is simple and easy to understand, upto 100 times faster, stable to data changes

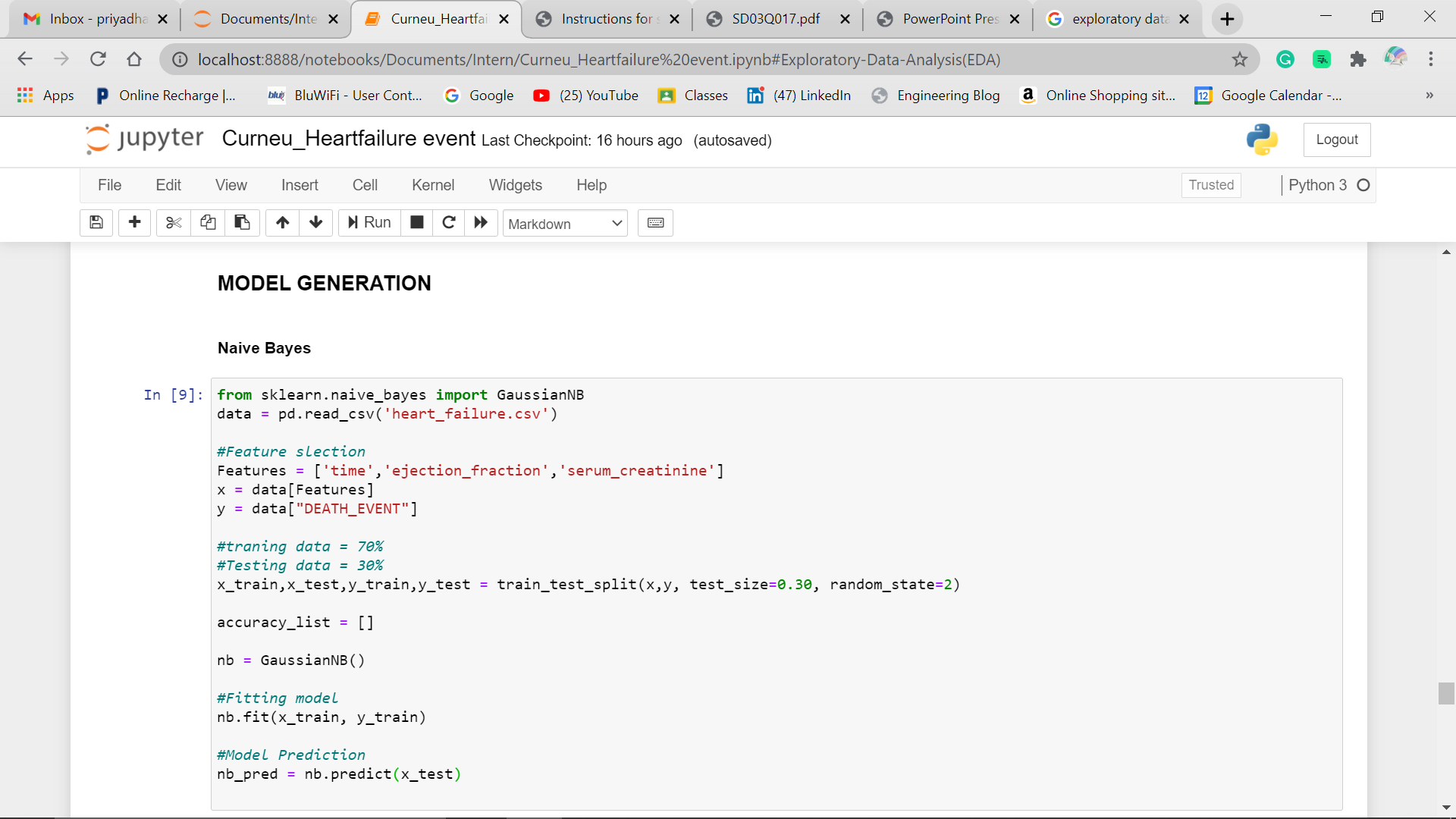
**Training and testing split:**

* We are splitting the dataset into:
  + 70% training
  + 30% testing



**Model generation**

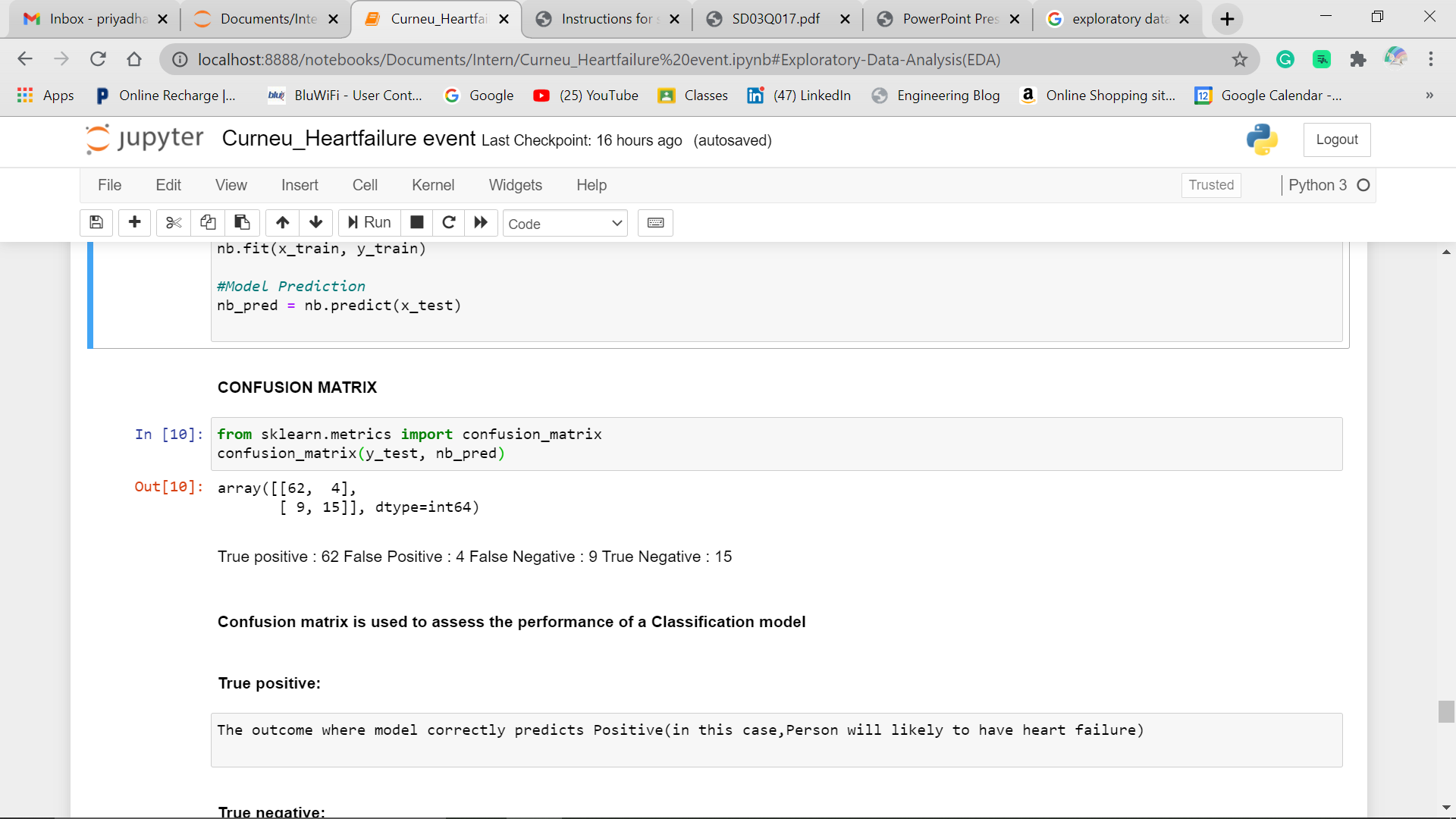
* **Once the splitting of data is done, we have to fit the training data**
* **And then, predict the testing data**



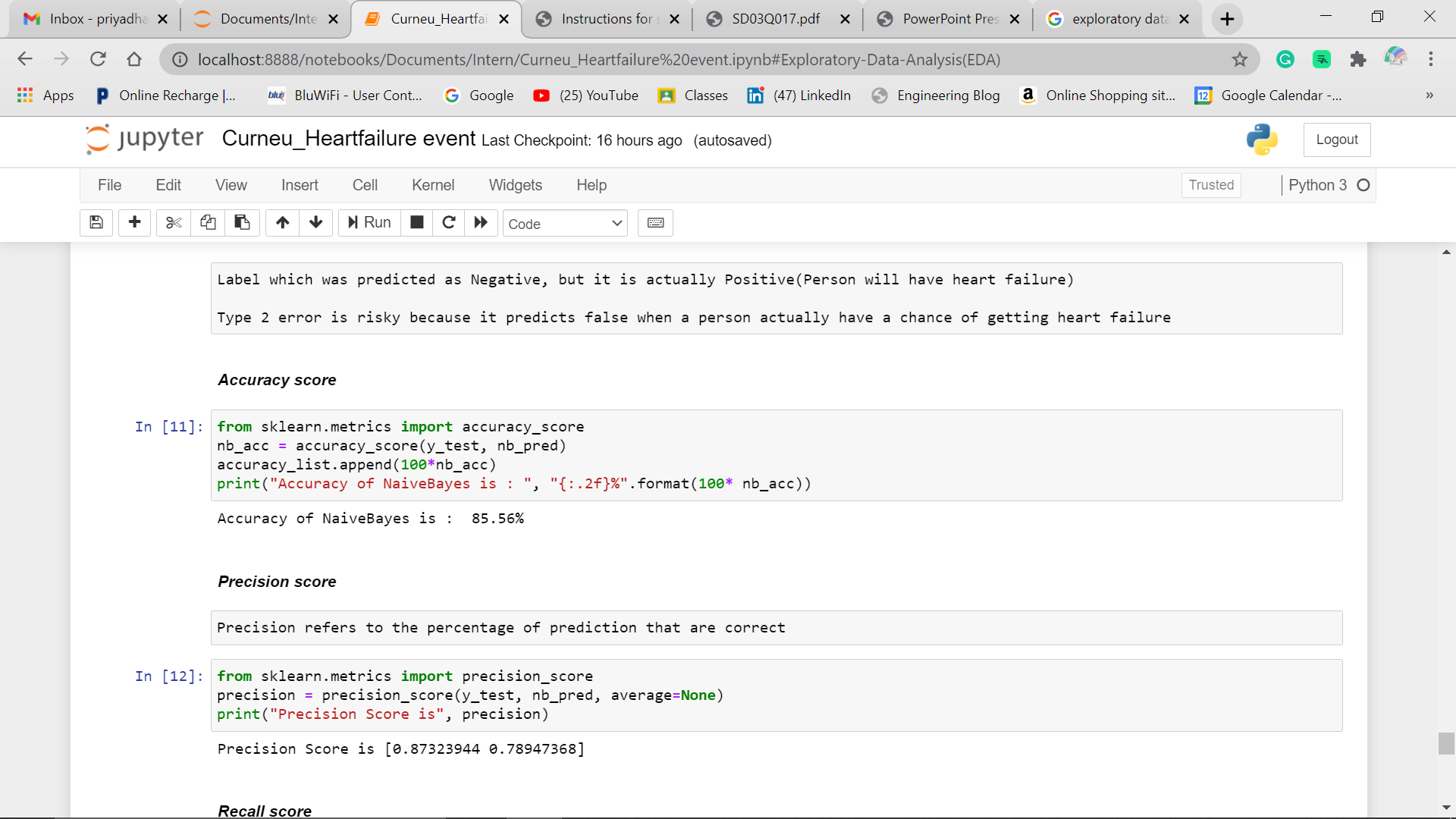
**EVALUATION METRIC:**

Evaluation is done to check whether your model preformed well or not. The following are the most commonly used metrics

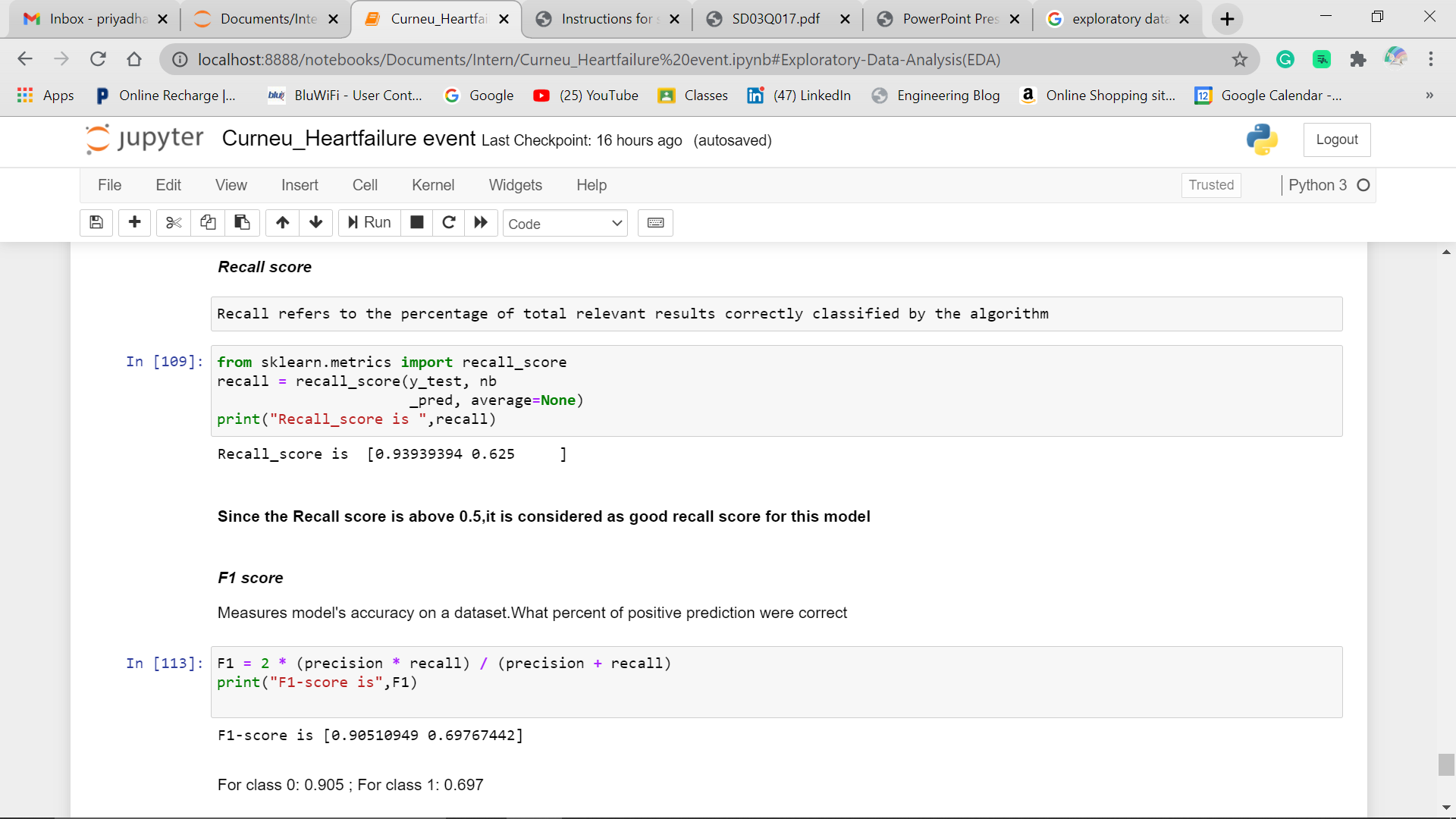
1. **Confusion matrix**
   1. Confusion matrix is used to assess the performance of a Classification model
   2. From the output, the type 2 error (False negative)is risky because it predicts negative when it is actually positive.in our case, predicting no risk of heart failure when there is a chance.



1. **Accuracy and precision:**
   1. The model has the accuracy score of about 85.56 % which is quite good.
   2. Precision refers to the percentage of prediction that are correct



1. **Recall and F1 score**
   1. Recall refers to the percentage of total relevant results correctly classified by the algorithm
   2. Since the Recall score is above 0.5 for class 0 and 1, it is considered as good recall score for this model.



Therefore, here we created a supervised machine learning model to access the likelihood of death, due to heart failure with an accuracy of about **85.56% which is more that 70% .**

The last step is Deploying the machine learning model and intergrate it into existing production environment where it can take input and return an output.