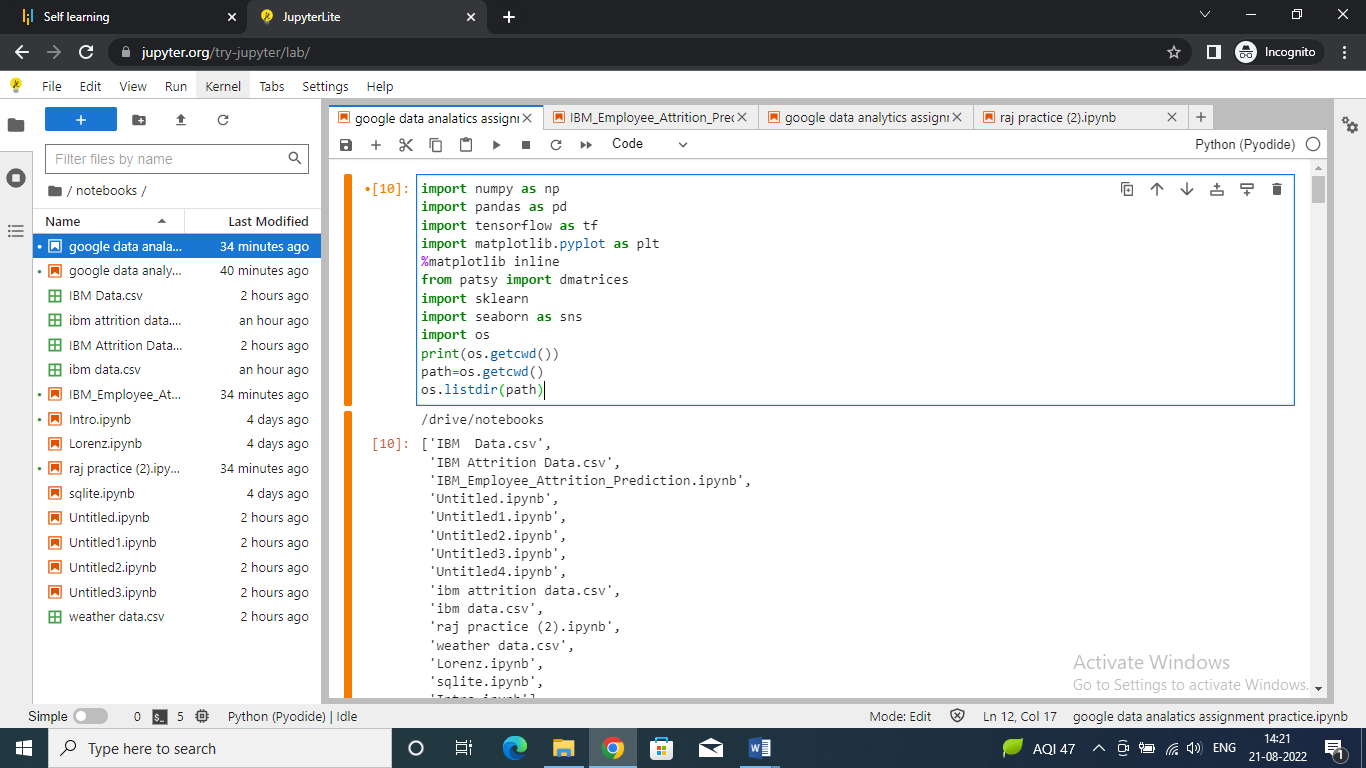
Code with output

Step 1 : Importing necessary libraries

Code:

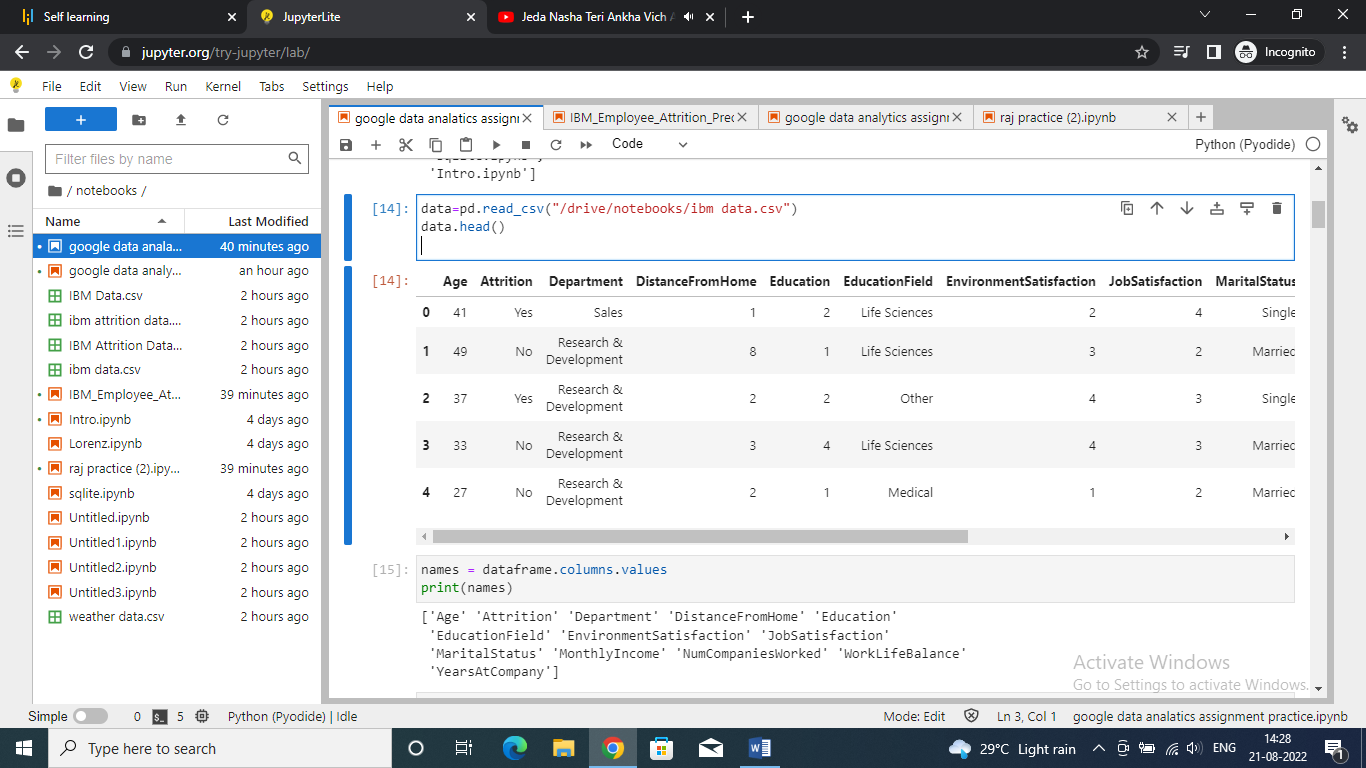
|  |
| --- |
| import numpy as np  import pandas as pd  import tensorflow as tf  import matplotlib.pyplot as plt  %matplotlib inline  from patsy import dmatrices  import sklearn  import seaborn as sns  import os  print(os.getcwd())  path=os.getcwd()  os.listdir(path) |



Step 2 : Importing dataset

Code:

|  |
| --- |
| data=pd.read\_csv("/drive/notebooks/ibm data.csv")  data.head() |



Step 3 : showing column names

Code:

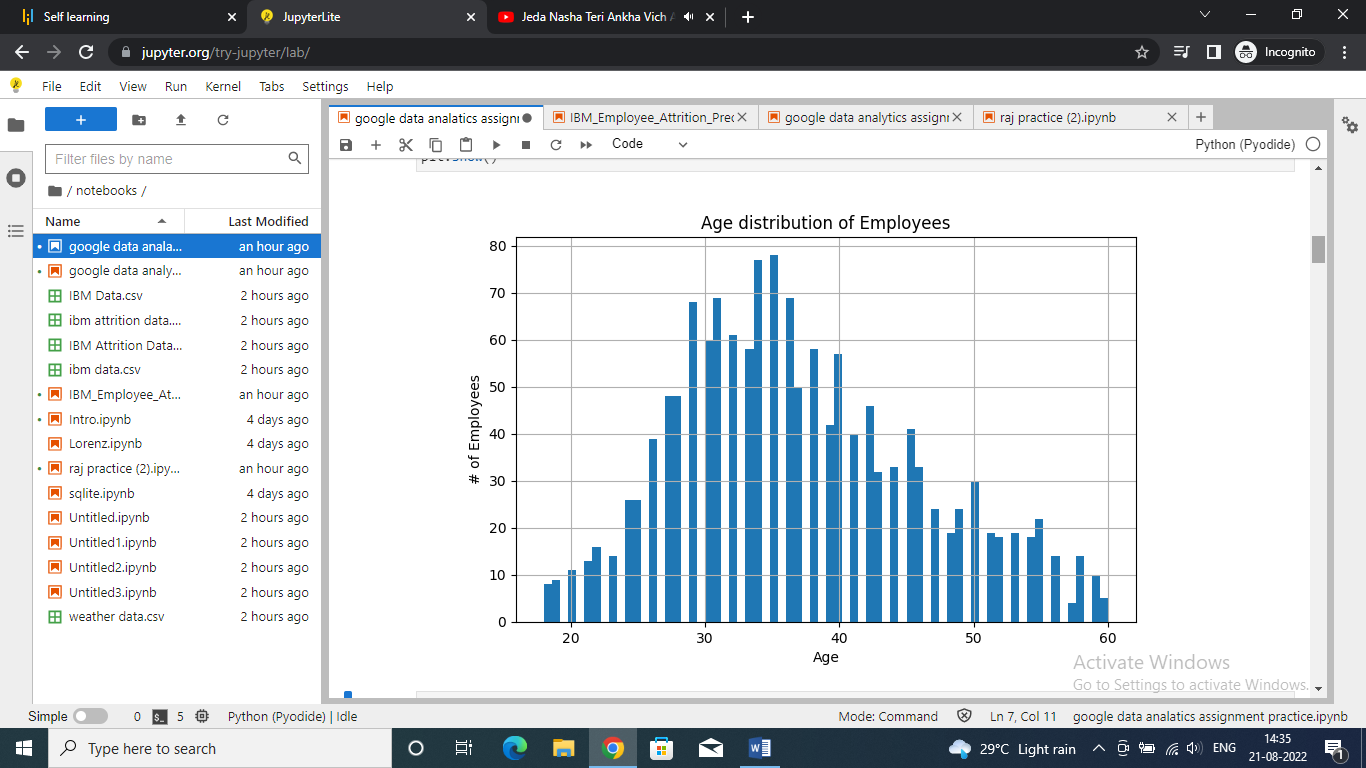
|  |
| --- |
| names = dataframe.columns.values  print(names) |



Step 4 : creating histogram of age for employees

Code:

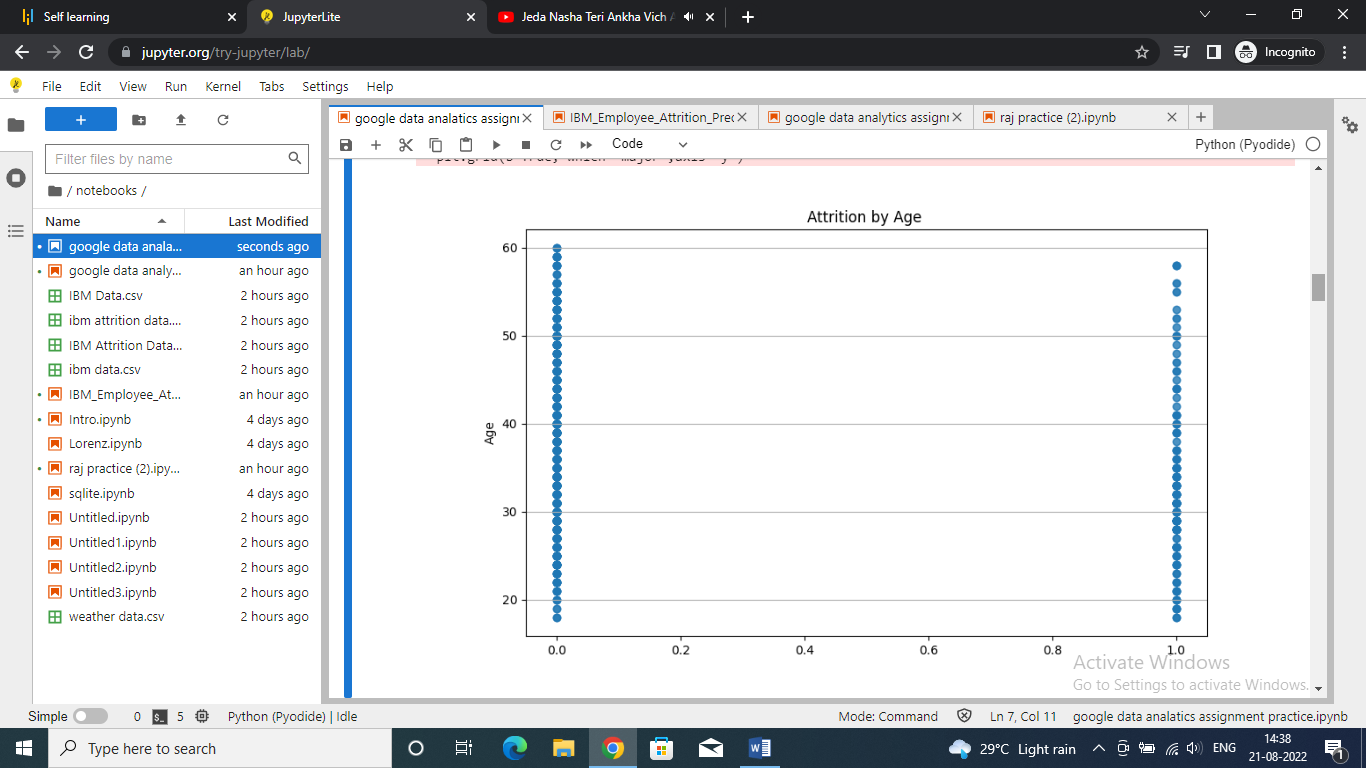
|  |
| --- |
| # histogram for age  import matplotlib.pyplot as plt  plt.figure(figsize=(8,5))  dataframe['Age'].hist(bins=70)  plt.title("Age distribution of Employees")  plt.xlabel("Age")  plt.ylabel("# of Employees")  plt.show() |



Step 5 : explore data for attrition by age

Code:

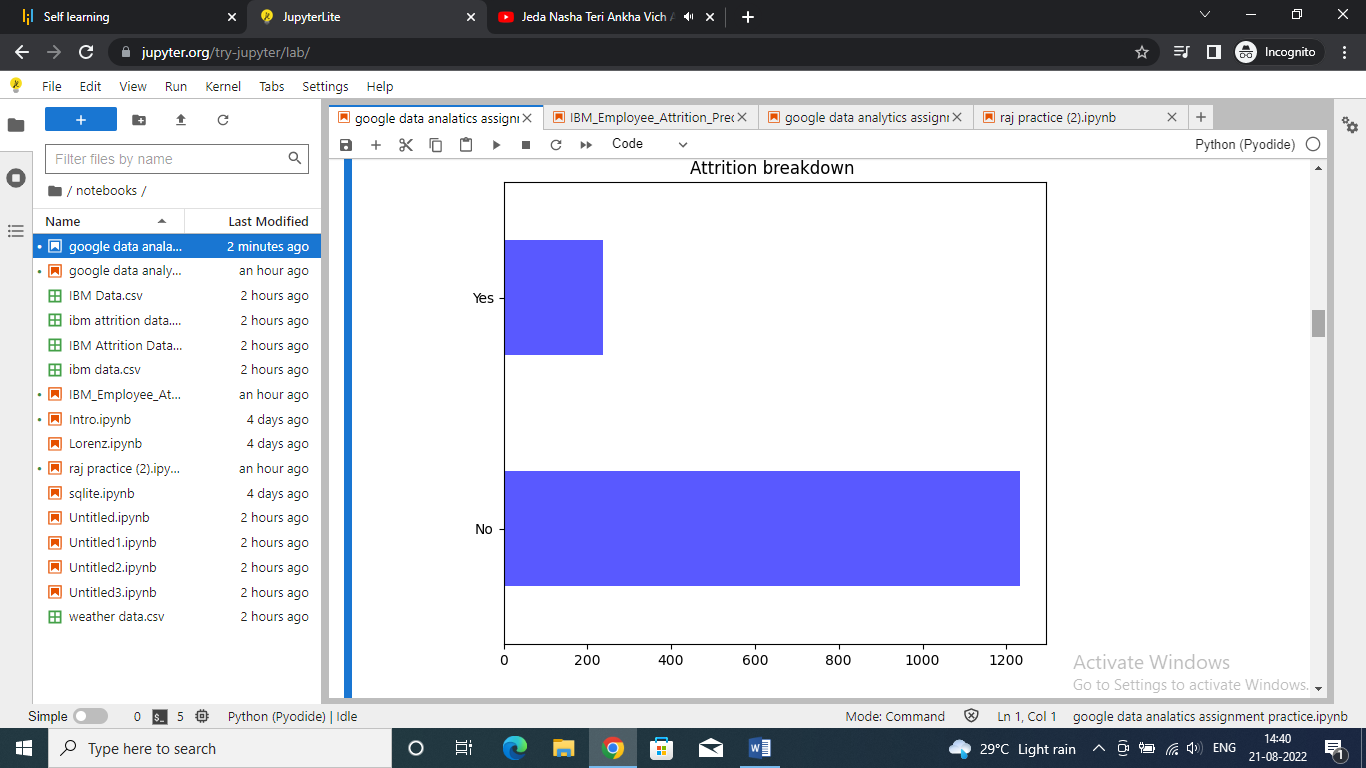
|  |
| --- |
| # explore data for Attrition by Age  plt.figure(figsize=(10,6))  plt.scatter(dataframe.Attrition,dataframe.Age, alpha=.55)  plt.title("Attrition by Age ")  plt.ylabel("Age")  plt.grid(b=True, which='major',axis='y')  plt.show() |



Step 6 : display bar graph for data of left employess breakdown

Code:

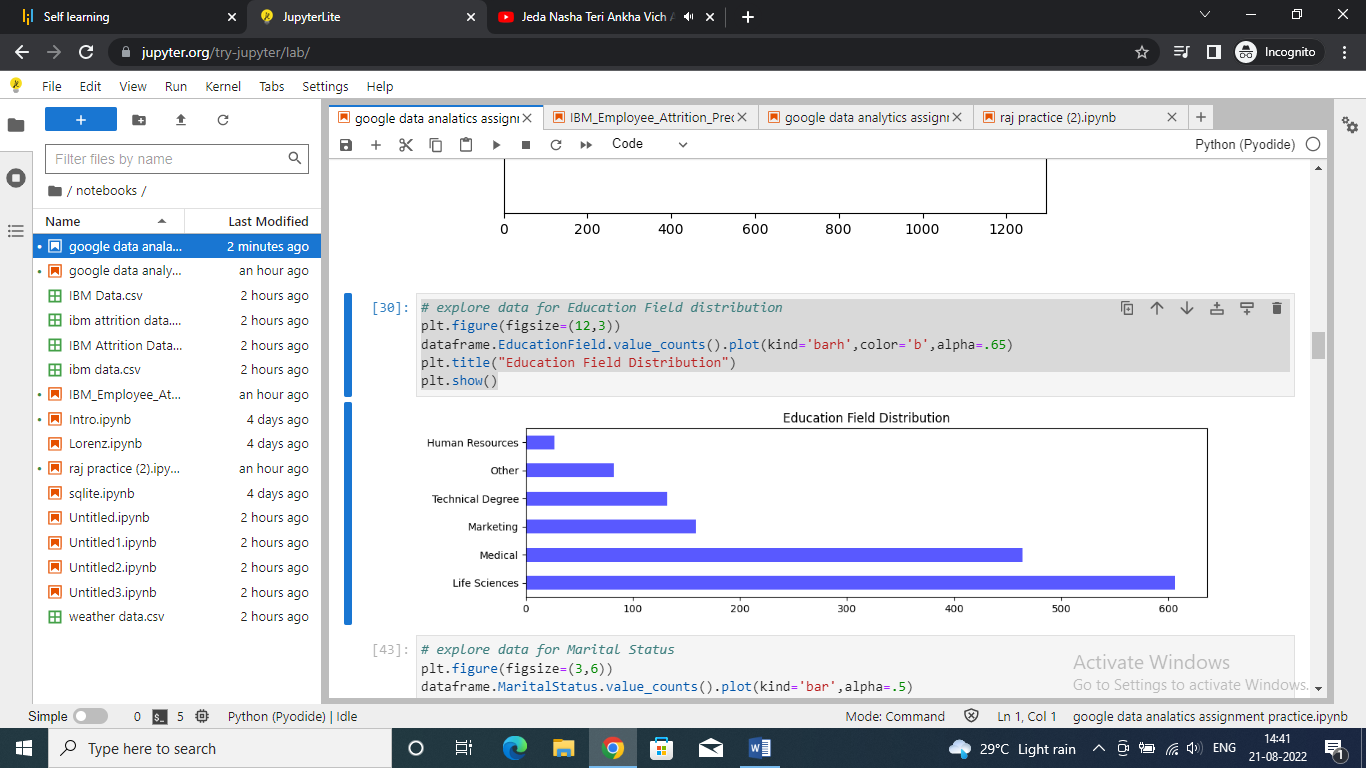
|  |
| --- |
| # explore data for Left employees breakdown  plt.figure(figsize=(7,6))  dataframe.Attrition.value\_counts().plot(kind='barh',color='blue',alpha=.65)  plt.title("Attrition breakdown ")  plt.show() |



Step 7 : plotting bargraph for education field distribution

Code:

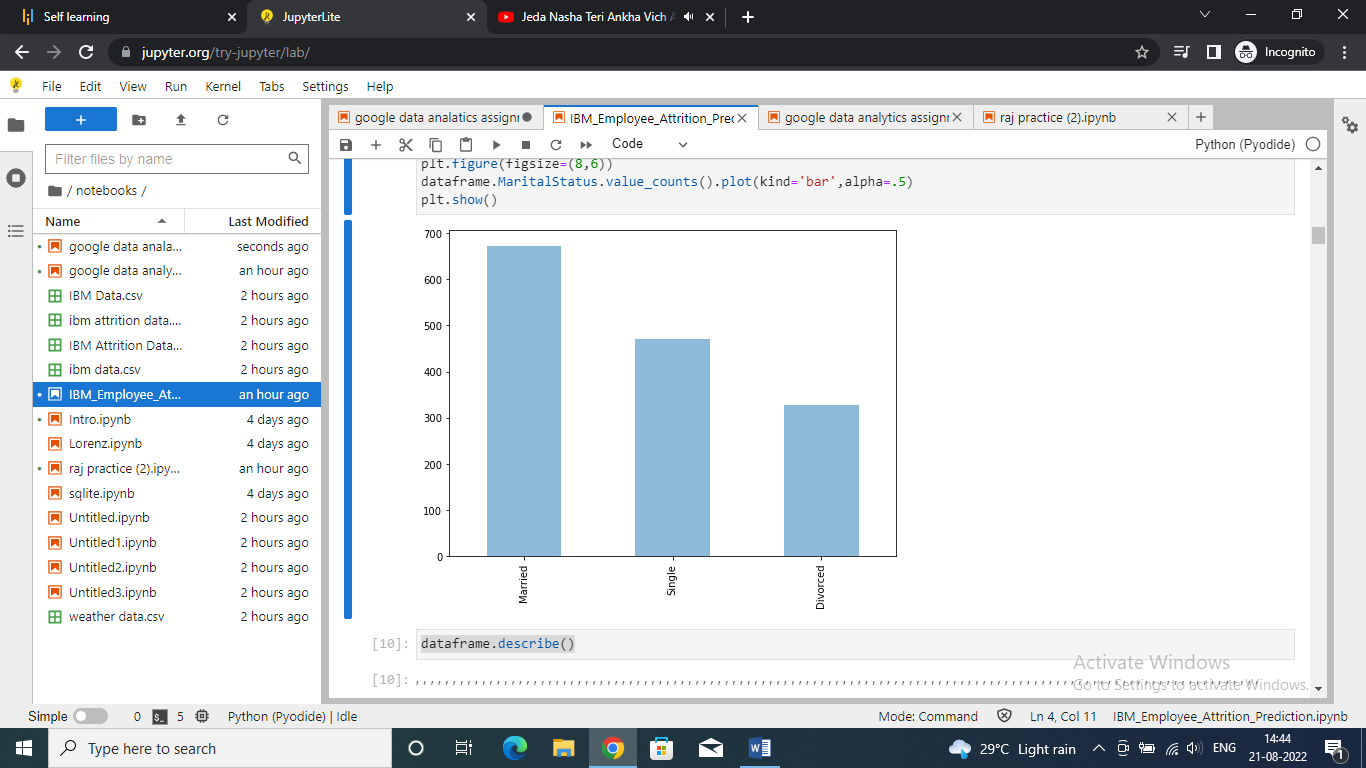
|  |
| --- |
| # explore data for Education Field distribution  plt.figure(figsize=(12,3))  dataframe.EducationField.value\_counts().plot(kind='barh',color='b',alpha=.65)  plt.title("Education Field Distribution")  plt.show() |



Step 8 : exploring data for martial status

Code:

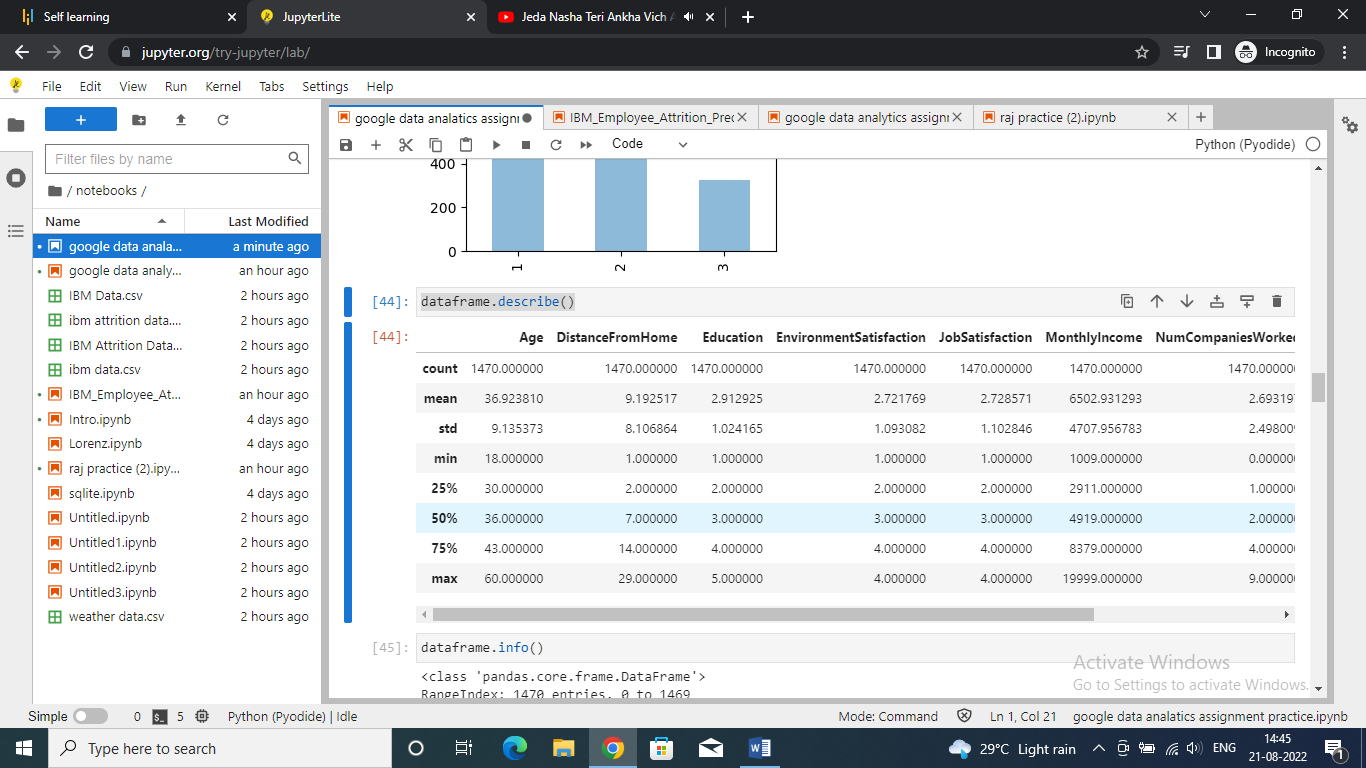
|  |
| --- |
| # explore data for Marital Status  plt.figure(figsize=(3,6))  dataframe.MaritalStatus.value\_counts().plot(kind='bar',alpha=.5)  plt.show() |



Step 9 : describing dataframe

Code:

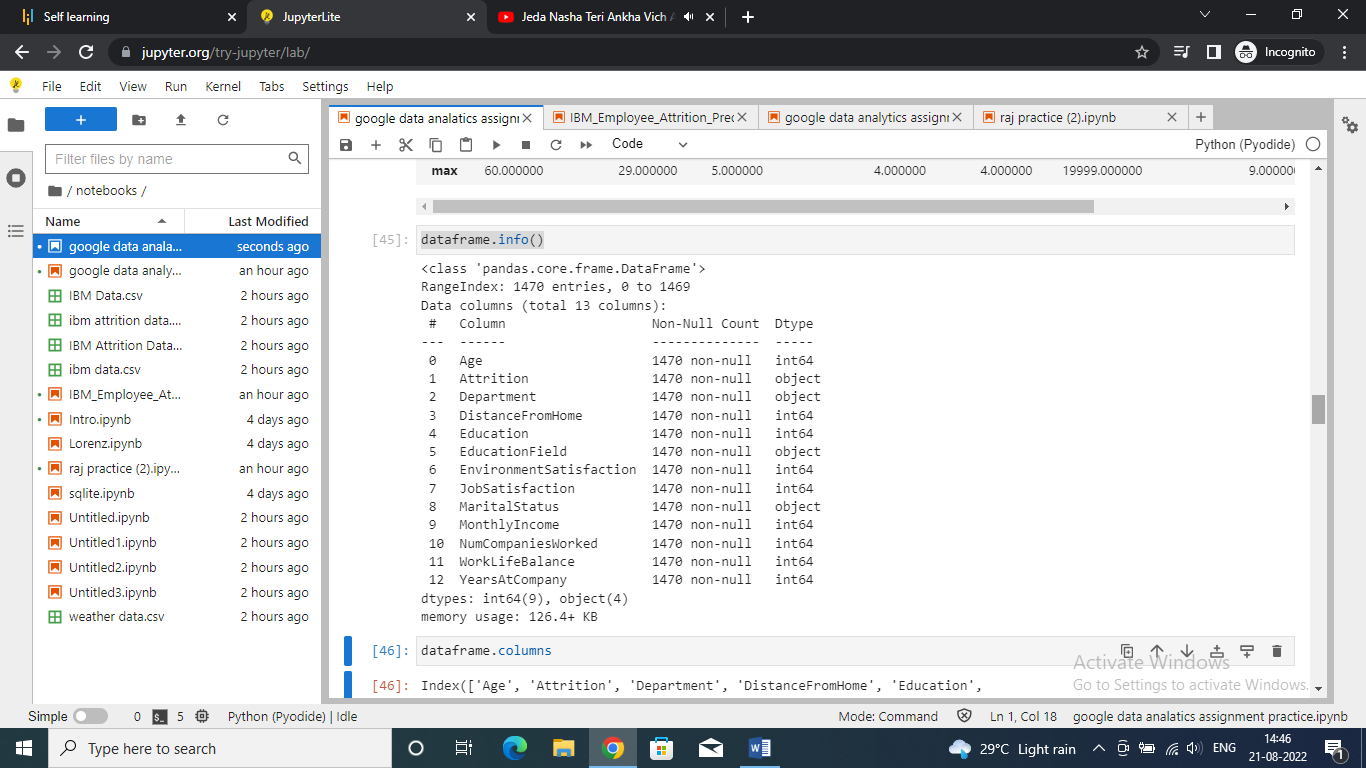
|  |
| --- |
| dataframe.describe() |



Step 10 : information about dataset

Code:

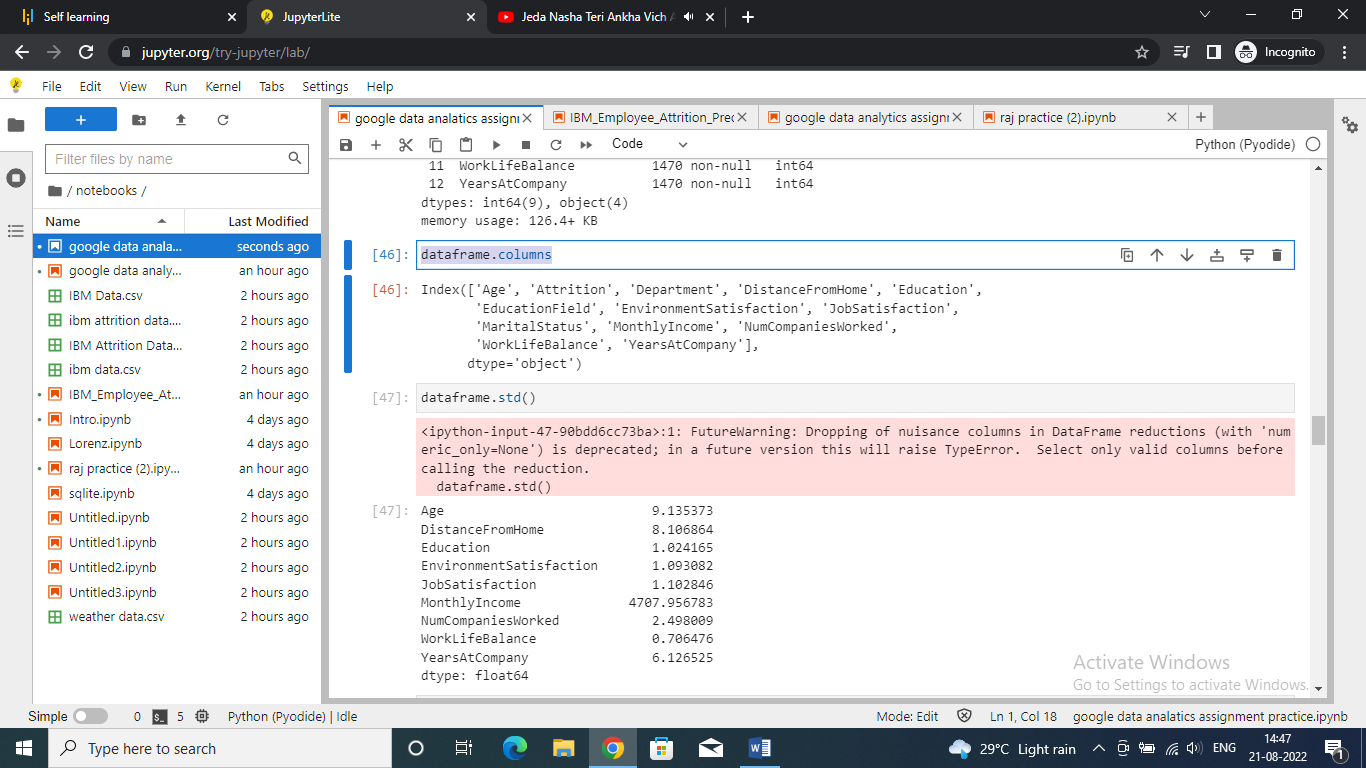
|  |
| --- |
| dataframe.info() |



Step 11 : displaying dataframe columns

Code:

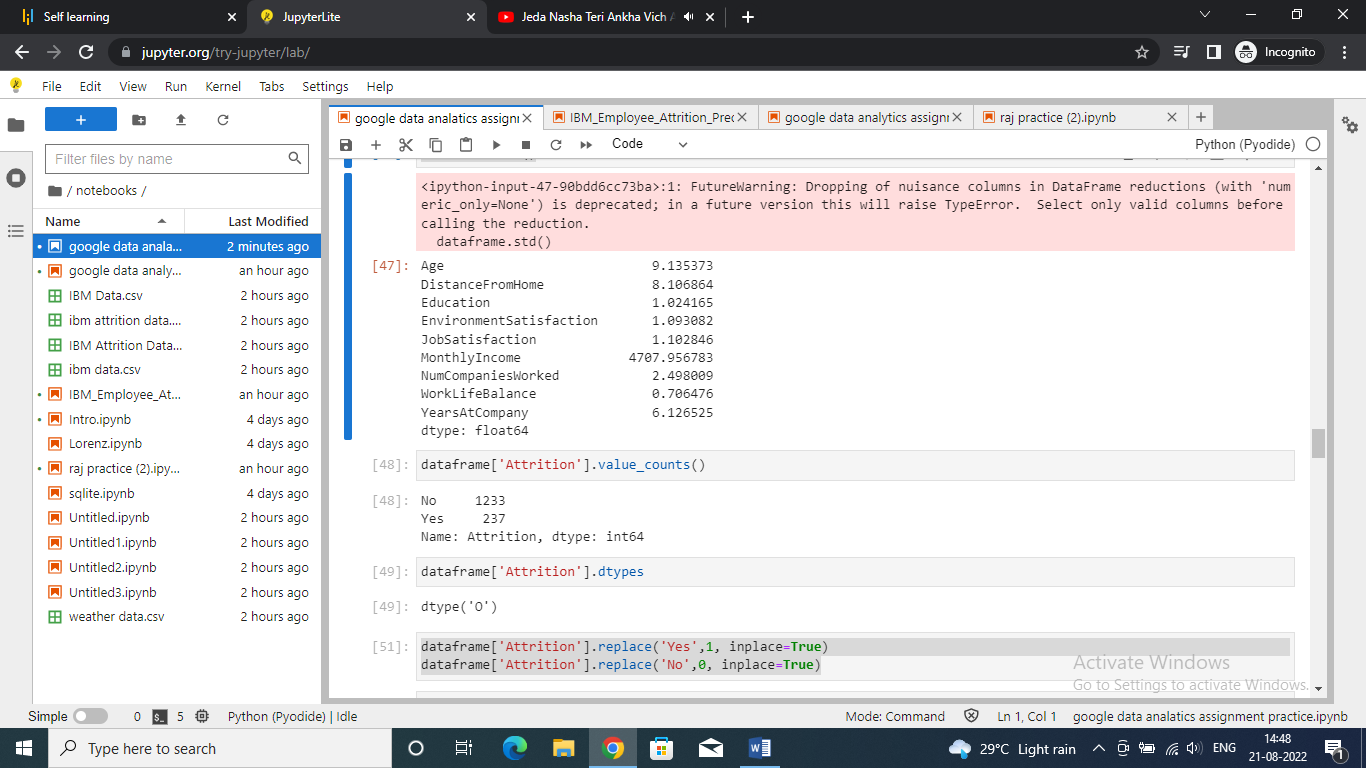
|  |
| --- |
| dataframe.columns |



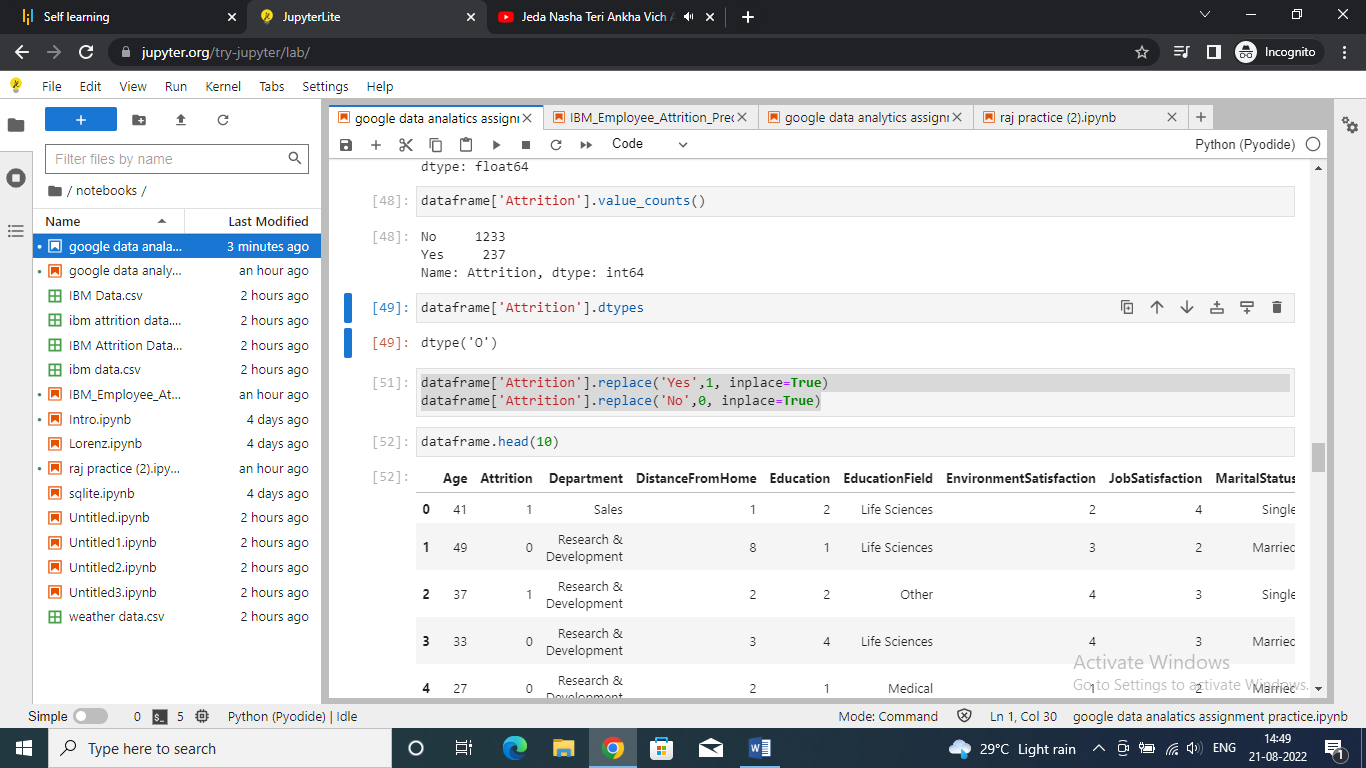
Step 12 : showing standard deviation in dataframe

Code:

|  |
| --- |
| dataframe.std() |



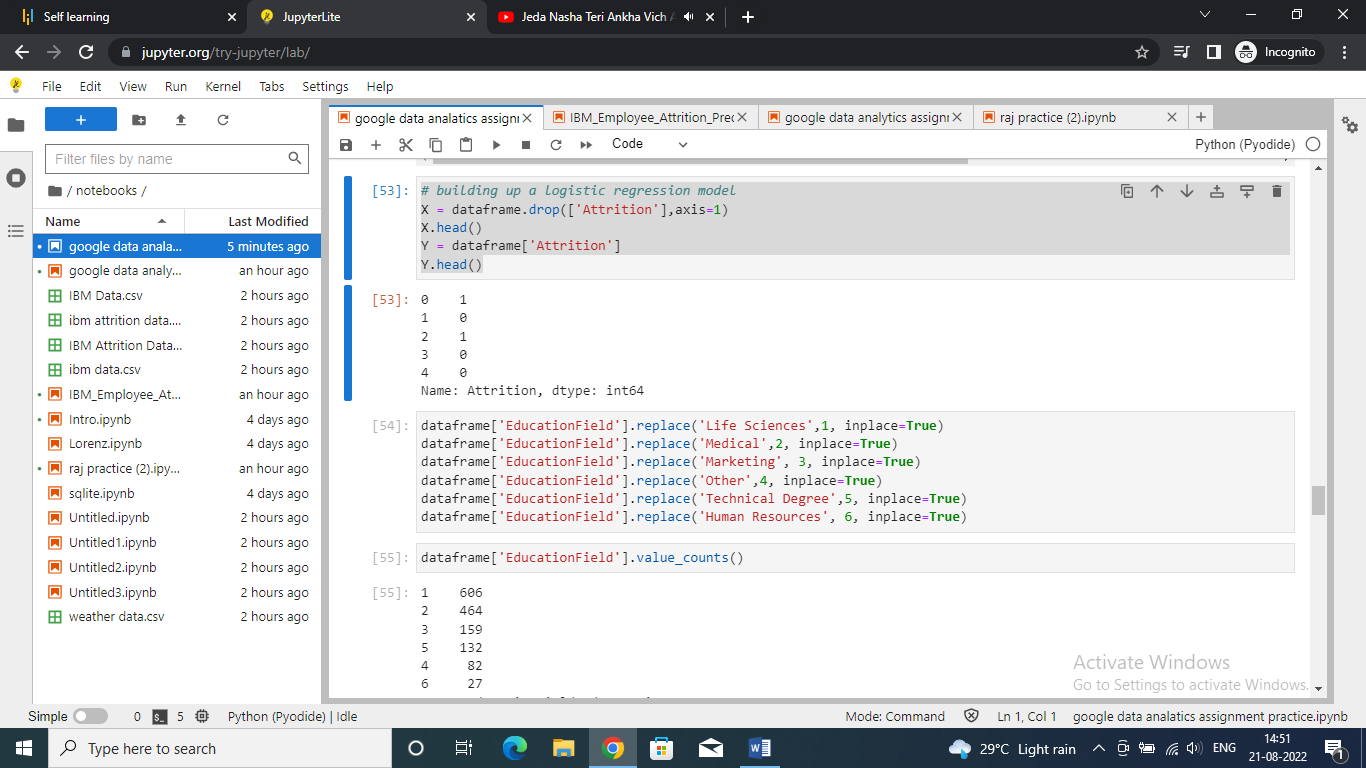
Showing attrition data



Step 13 : building a logistic regression model

Code:

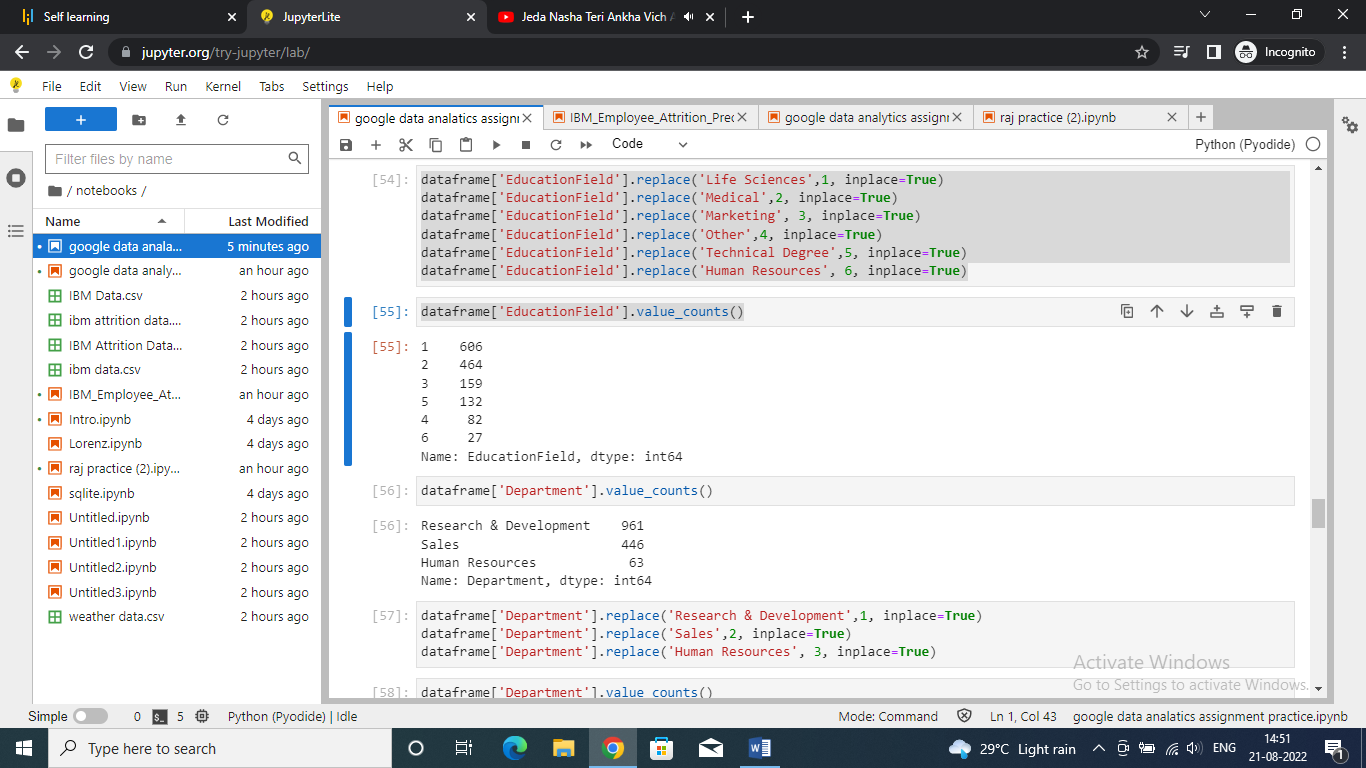
|  |
| --- |
| # building up a logistic regression model  X = dataframe.drop(['Attrition'],axis=1)  X.head()  Y = dataframe['Attrition']  Y.head() |



Step 14 : replacing columns names with numeric values

Code:

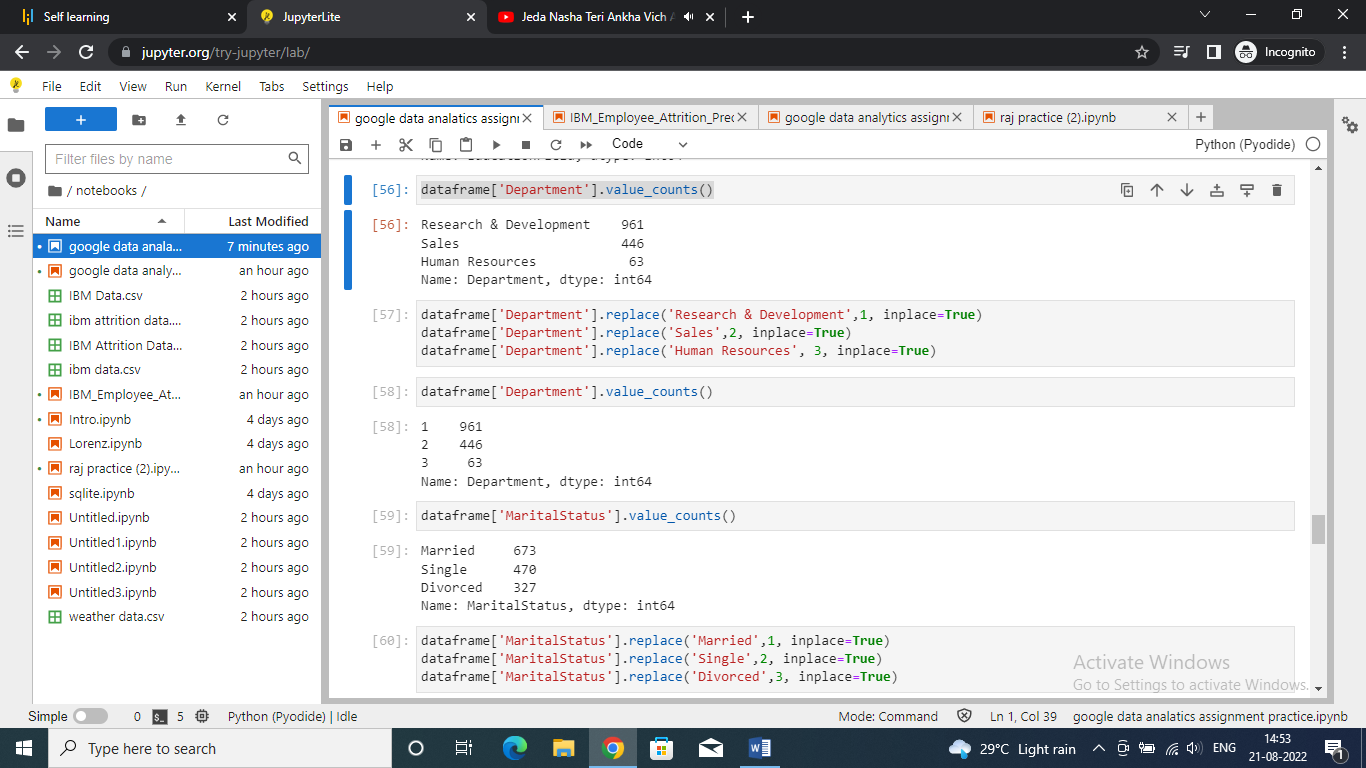
|  |
| --- |
| dataframe['EducationField'].replace('Life Sciences',1, inplace=True)  dataframe['EducationField'].replace('Medical',2, inplace=True)  dataframe['EducationField'].replace('Marketing', 3, inplace=True)  dataframe['EducationField'].replace('Other',4, inplace=True)  dataframe['EducationField'].replace('Technical Degree',5, inplace=True)  dataframe['EducationField'].replace('Human Resources', 6, inplace=True)  dataframe['EducationField'].value\_counts() |



Step 15 : showing the values of department column

Code:

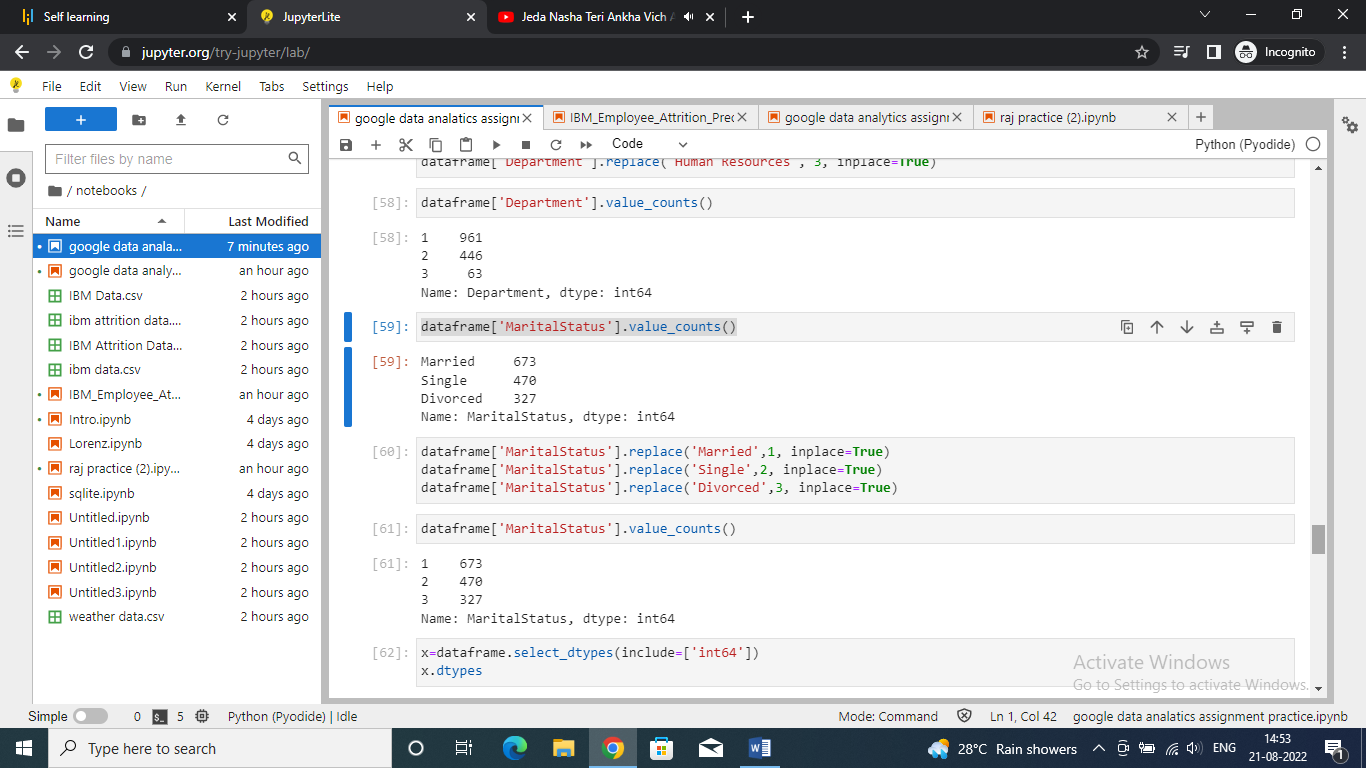
|  |
| --- |
| dataframe['Department'].value\_counts() |



Step 16 : displaying values of martial status

Code:

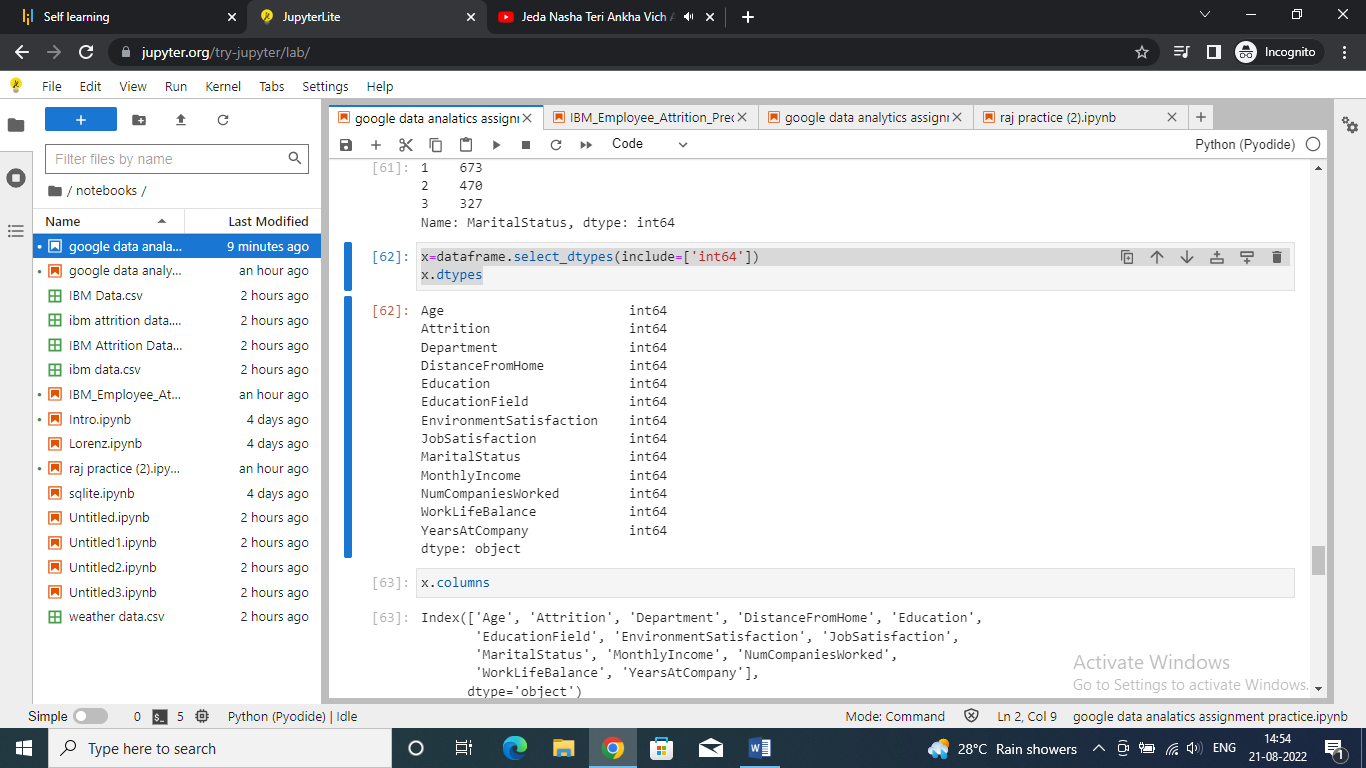
|  |
| --- |
| dataframe['MaritalStatus'].value\_counts() |



Step 17 : displaying datatypes of various columns

Code:

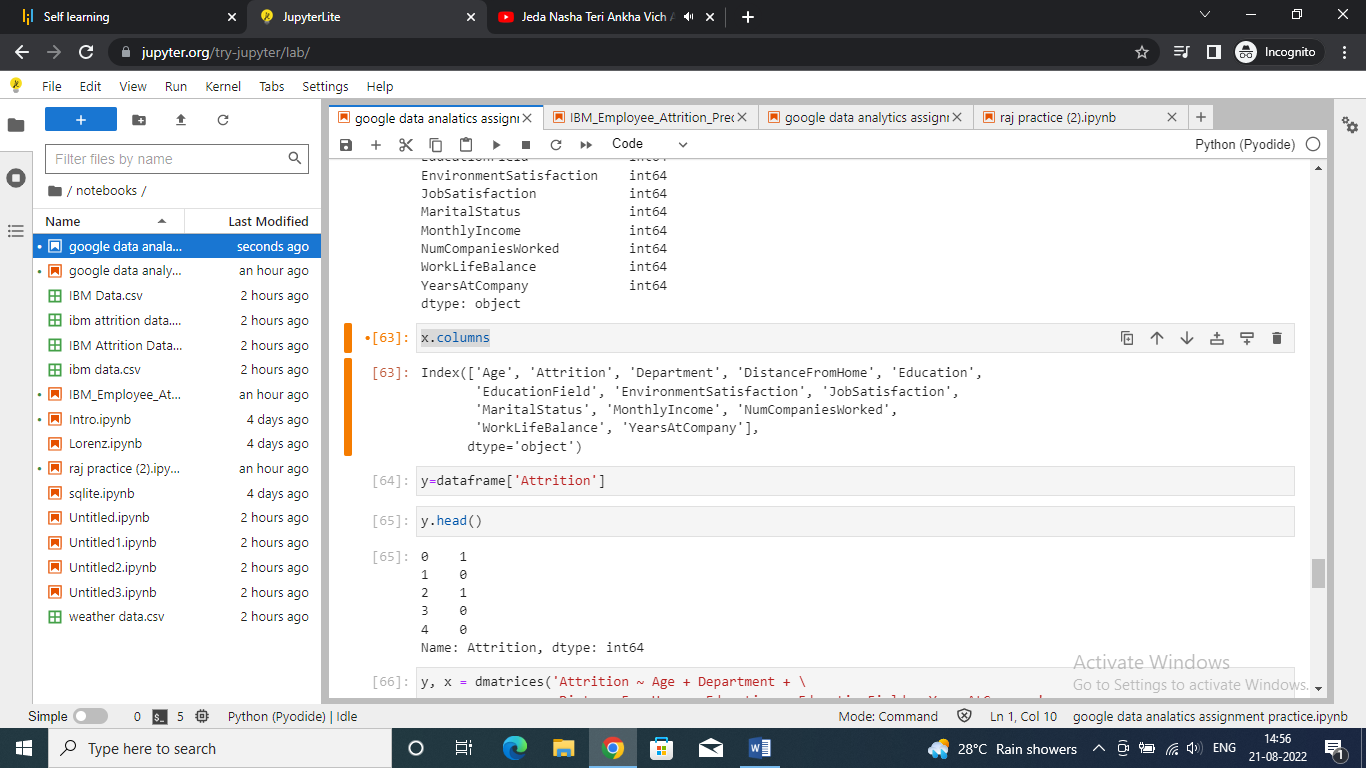
|  |
| --- |
| x=dataframe.select\_dtypes(include=['int64'])  x.dtypes |



Step 18 : displaying columns in x(dataframe named as x)

Code:

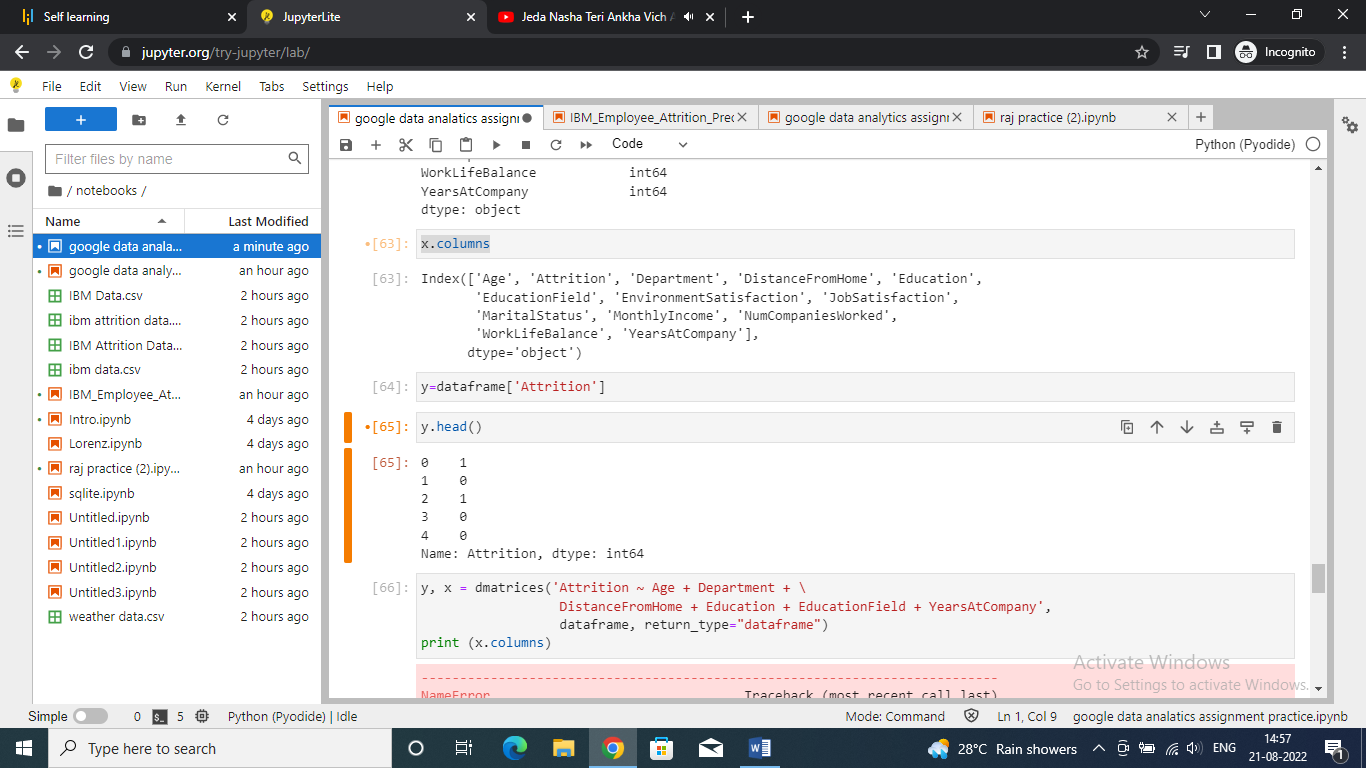
|  |
| --- |
| x.columns |



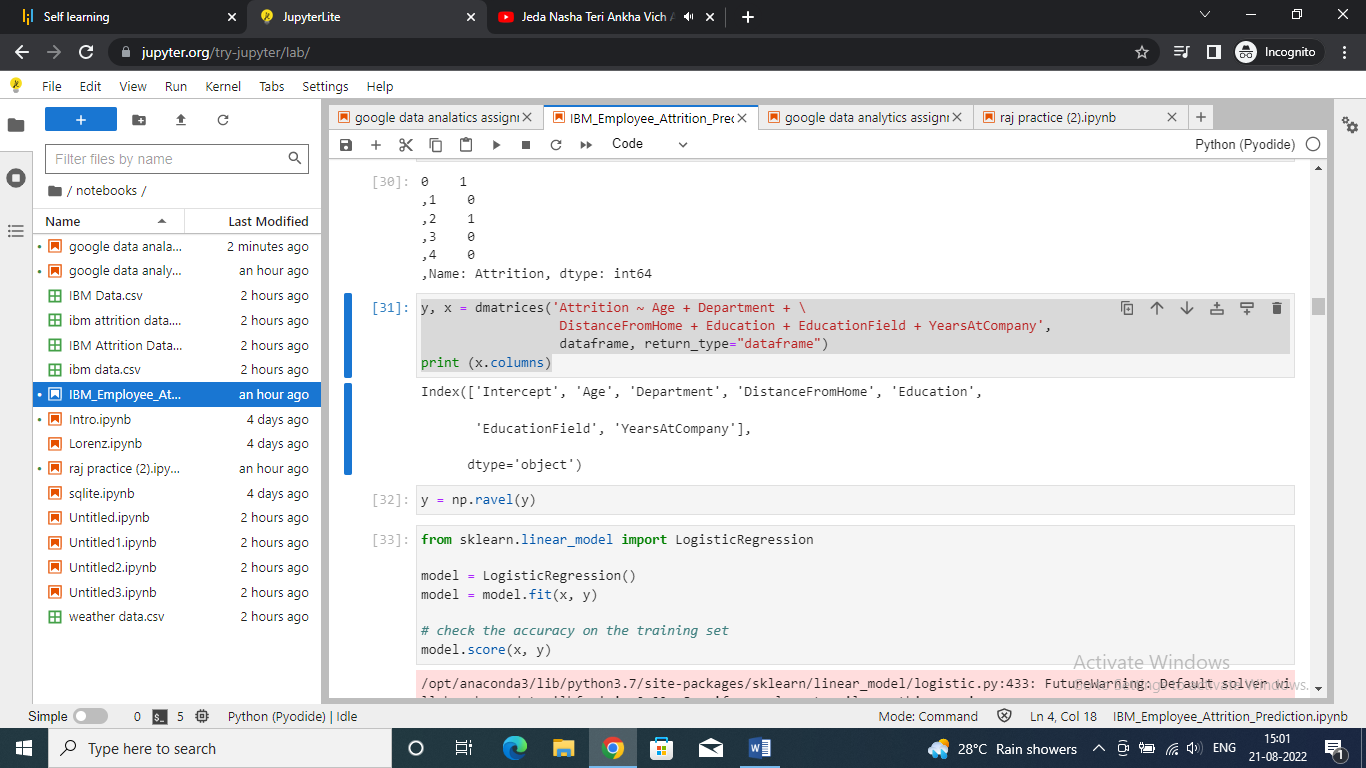
Step 19 : displaying showing attrition and data of y

Code:

|  |
| --- |
| y=dataframe['Attrition']  y.head() |



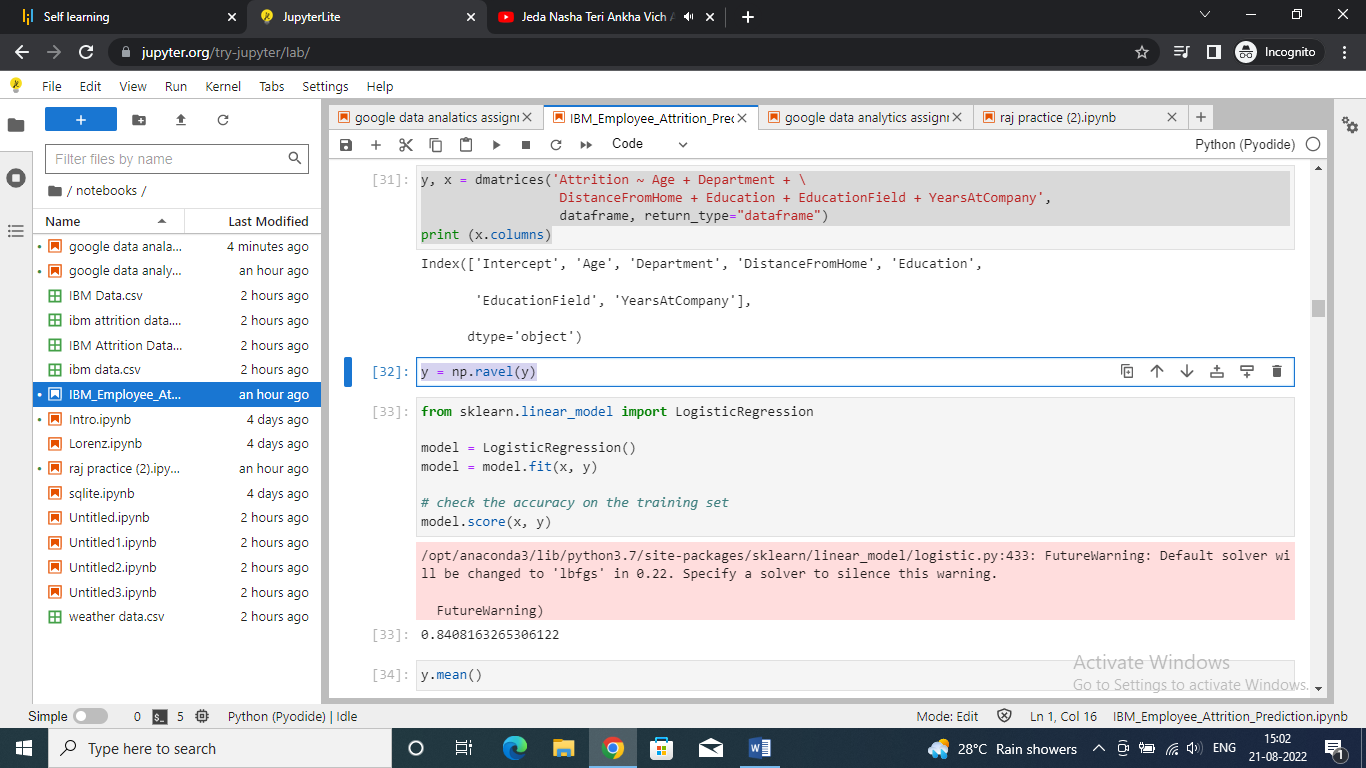
|  |
| --- |
| y, x = dmatrices('Attrition ~ Age + Department + \  DistanceFromHome + Education + EducationField + YearsAtCompany',  dataframe, return\_type="dataframe")  print (x.columns) |



Step 21 : testing accuracy of model

Code:

|  |
| --- |
| y = np.ravel(y)  from sklearn.linear\_model import LogisticRegression  model = LogisticRegression()  model = model.fit(x, y)  # check the accuracy on the training set  model.score(x, y) |



2

Step 22 : testing mean of y

Code:

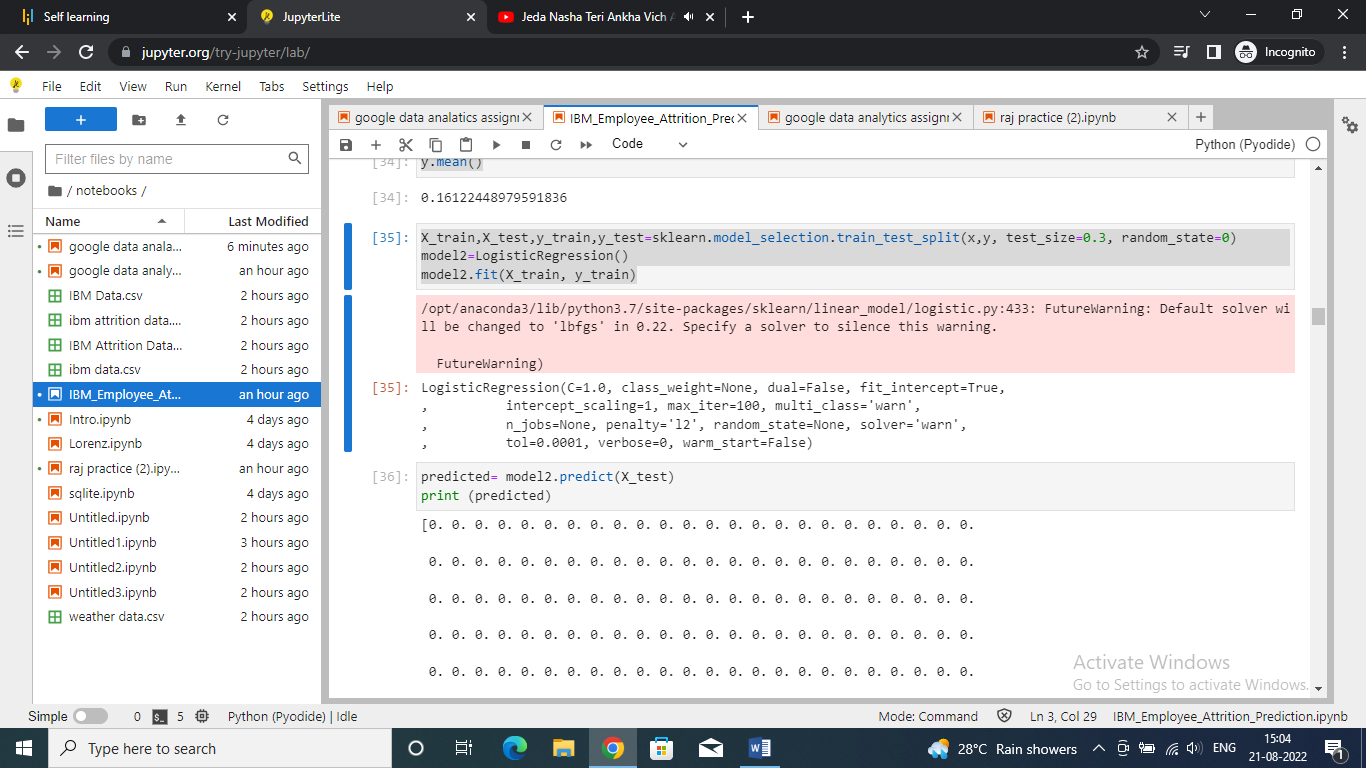
|  |
| --- |
| y.mean() |

0.16122448979591836

Step 23:

Code:

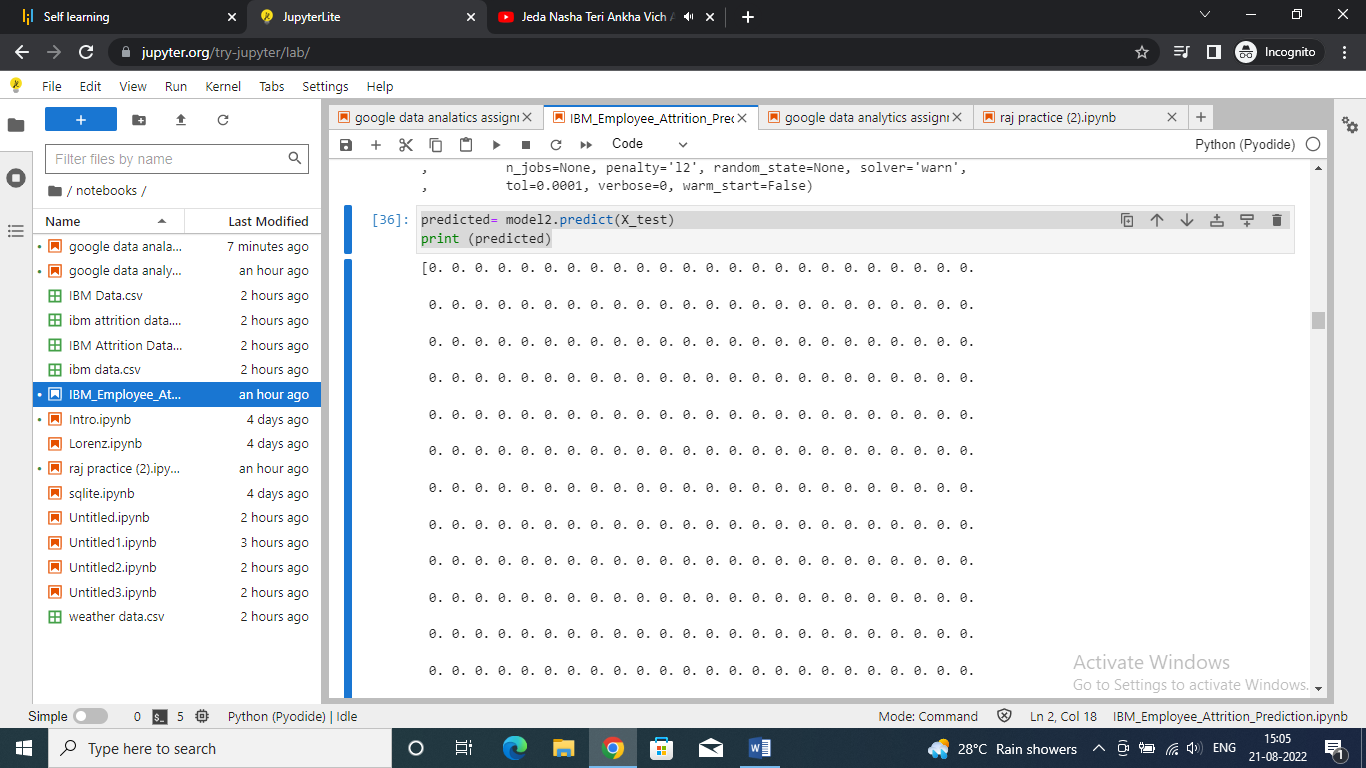
|  |
| --- |
| X\_train,X\_test,y\_train,y\_test=sklearn.model\_selection.train\_test\_split(x,y, test\_size=0.3, random\_state=0)  model2=LogisticRegression()  model2.fit(X\_train, y\_train) |



Step 24 : checking model prediction

Code:

|  |
| --- |
| predicted= model2.predict(X\_test)  print (predicted) |



Step 25 : checking model for x\_test

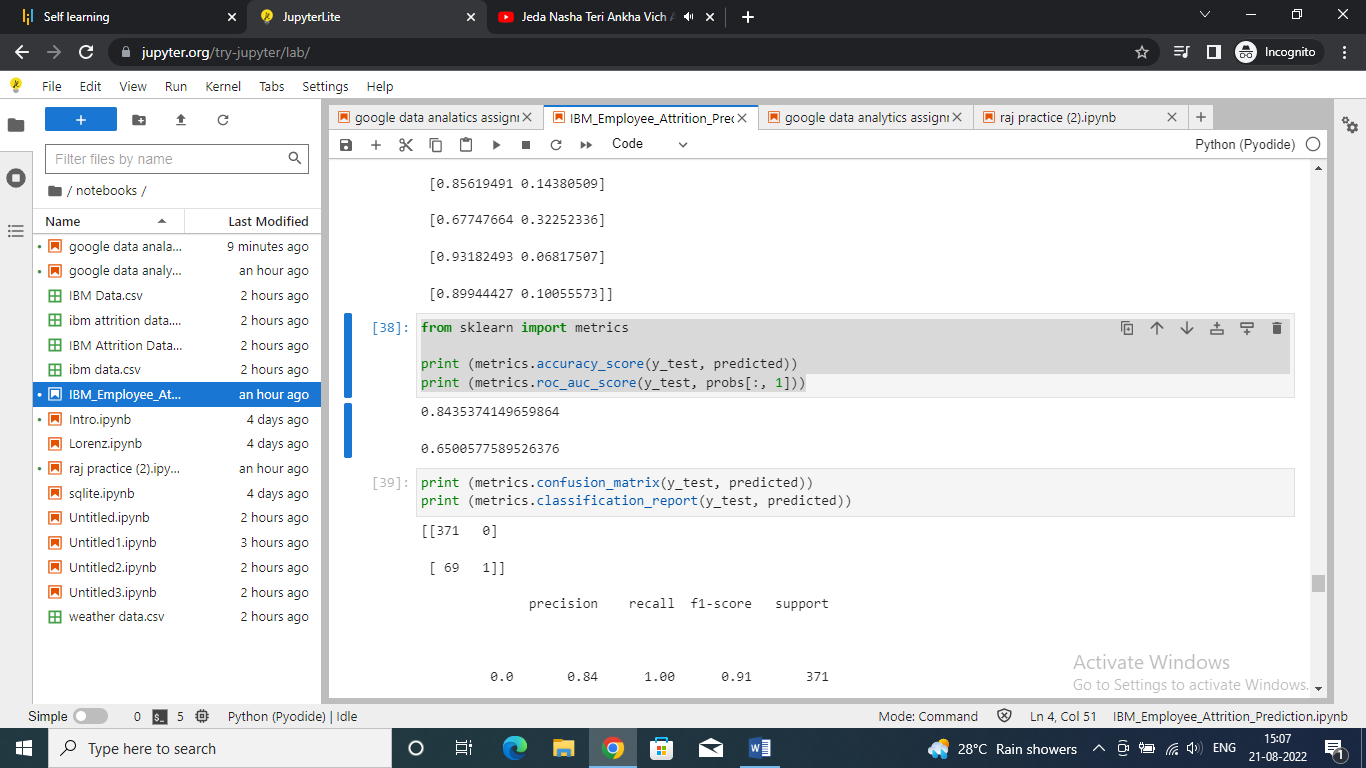
Code:

|  |
| --- |
| probs = model2.predict\_proba(X\_test)  print (probs) |

Step 26 : displaying metrics of predicted and tested probability

Code:

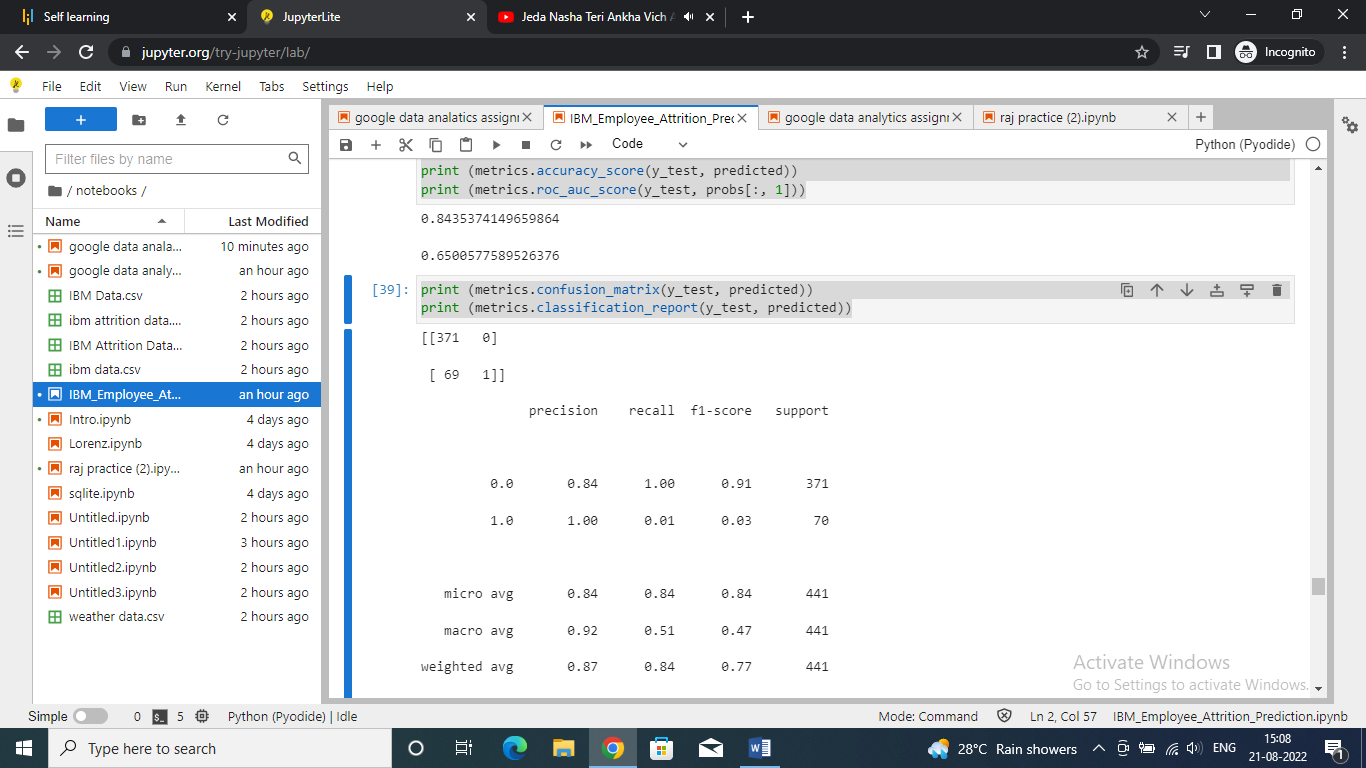
|  |
| --- |
| from sklearn import metrics  print (metrics.accuracy\_score(y\_test, predicted))  print (metrics.roc\_auc\_score(y\_test, probs[:, 1])) |



Step 27 : displaying models2 data

Code:

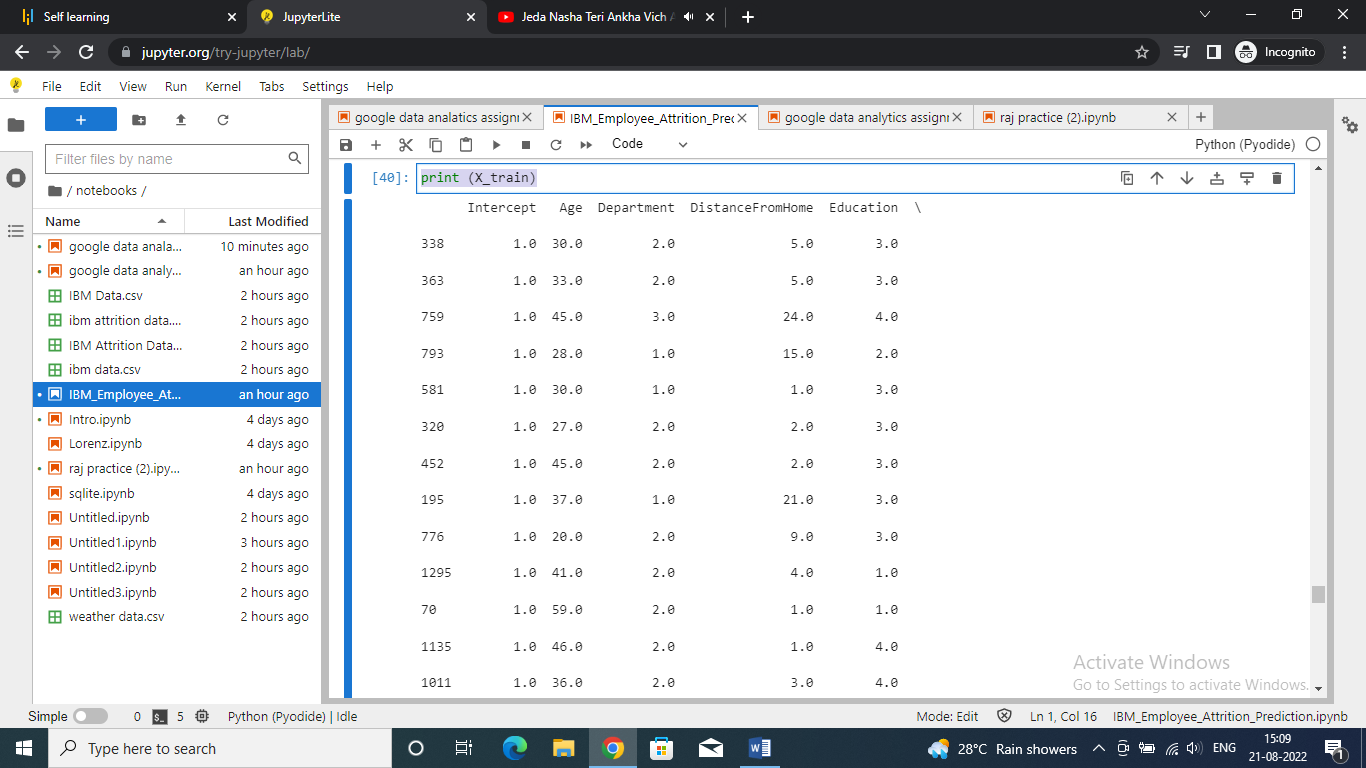
|  |
| --- |
| print (metrics.confusion\_matrix(y\_test, predicted))  print (metrics.classification\_report(y\_test, predicted)) |



Step 28 : display values for x\_train

Code:

|  |
| --- |
| print (X\_train) |



Step 28 : checking values to according to parameters to check the probability of attrition of employee

Code:

|  |
| --- |
| #add random values to according to the parameters mentioned above to check the proabily of attrition of the employee  kk=[[1.0, 23.0, 1.0, 500.0, 3.0, 24.0, 1.0]]  print(model.predict\_proba(kk)) |

