

Data science and machine intelligence

Assignment -1(Individual)





**Program Graduate Diploma in Information Technology (GDIT)**

**Study Block 2019-04**

**Course Code IX726001**

**Course Title Data Science and Machine Intelligence**

**Assessment No 01**

**Assessment Type Individual**

**Total Marks 95 marks**

**Assessment Weightage 25%**

**Exam Date and Time Week 2 of study block**

**Due Date 15 September 20019, Sunday 11:59 PM**

**Student Name: Vimitaben Mukeshchandra Vaidya**

**Students ID: 1000063758**

# **Task 1. Data Visualization and Analysis – Case Study: Little Women**

## **Action I. Separate texts into chapters.**

*#to seperate the chapters, THIS BOOK HAS TOTAL 47 CHAPTERS*

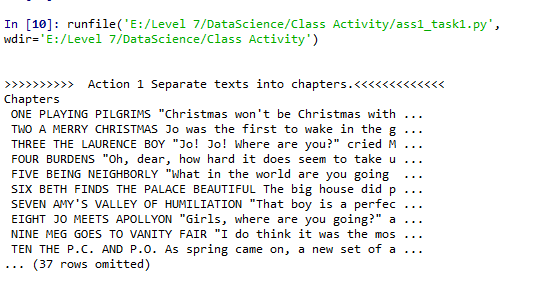
little\_women\_chapters = little\_women\_text.split('CHAPTER')[1:]

*# The chapters of Little Women, in a table*

chapters = Table().with\_column('Chapters\n', little\_women\_chapters)

print(chapters)

### **Result**



## **Action II. Find how many times each character is mentioned in every chapter.**

print('\n')

print('>>>>>>>>>>>Action 2 Find how many times each character is mentioned in every chapter.<<<<<<<<<')

*# Plot the cumulative counts.*

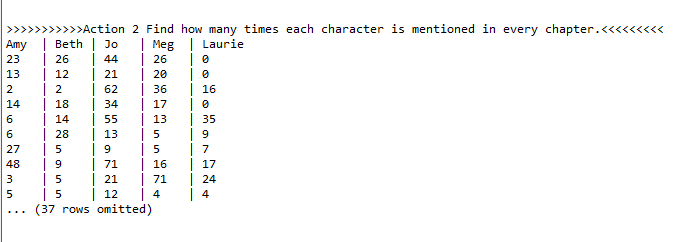
print(counts)

cum\_counts = counts.cumsum().with\_column('Chapter', np.arange(1, 48, 1))

cum\_counts.plot(column\_for\_xticks=5)

plots.title('Cumulative Number of Times Each Name Appears', y=1.08);

### **Result**



## **Action III.** **Using the ‘*datascience*’ library, convert these values in a TABLE construct from the library.**

cum\_counts = counts.cumsum().with\_column('Chapter', np.arange(1, 48, 1))

cum\_counts.plot(column\_for\_xticks=5)

plots.title('Cumulative Number of Times Each Name Appears', y=1.08);

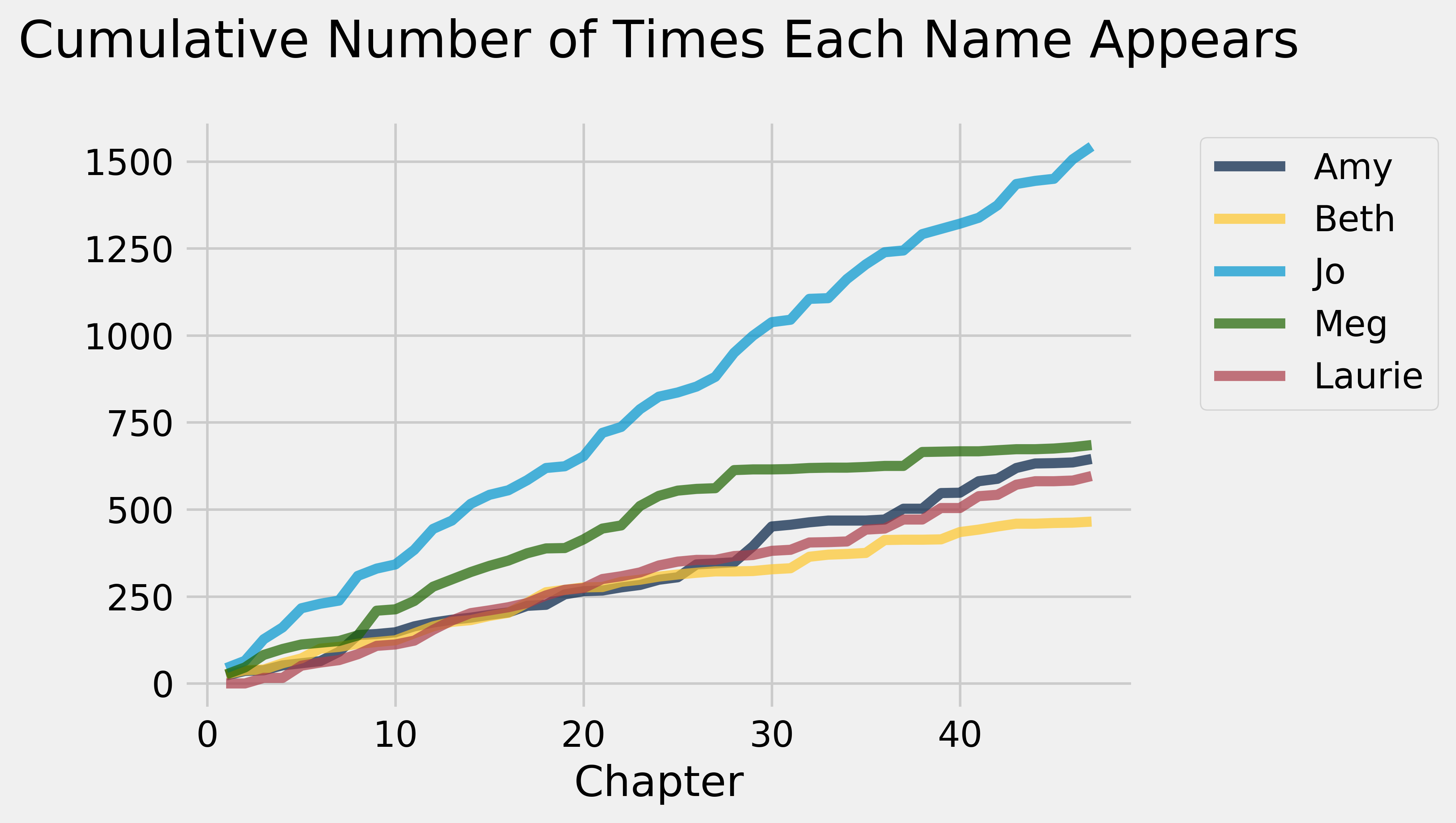
## **Action IV. Using this TABLE construct, plot how many times each character is mentioned in the novel. You will make sure to use the “fivethirtyeight” [URL] plotting style.**

plots.grid(True)

plots.show()

plots.savefig("fivethirtyeight.png",dpi=500,bbox\_inches='tight')

### **Result**



## **Action V. Using the TABLE construct, plot a scatter diagram between the length of each chapter (in terms of total characters in the chapter) and the total number of sentences in the chapter (considering FULL-STOP as a sentence delimiter).**

print('\n')

chars\_periods\_little\_women = Table().with\_columns([

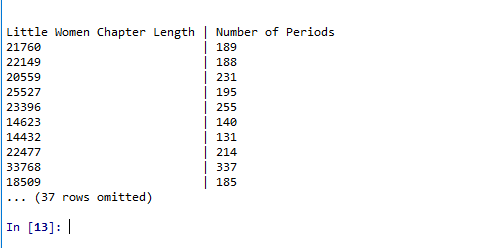
'Little Women Chapter Length', [len(s) for s in little\_women\_chapters],

'Number of Periods', np.char.count (little\_women\_chapters, '.')

])

print(chars\_periods\_little\_women)

### **Result**



## **Action VI. From your data analysis, identify the main character of the novel and justify how you have identified it.**

plots.figure(figsize=(6, 6))

plots.style.use('fivethirtyeight')

plots.scatter(chars\_periods\_little\_women.column(1),

chars\_periods\_little\_women.column(0),

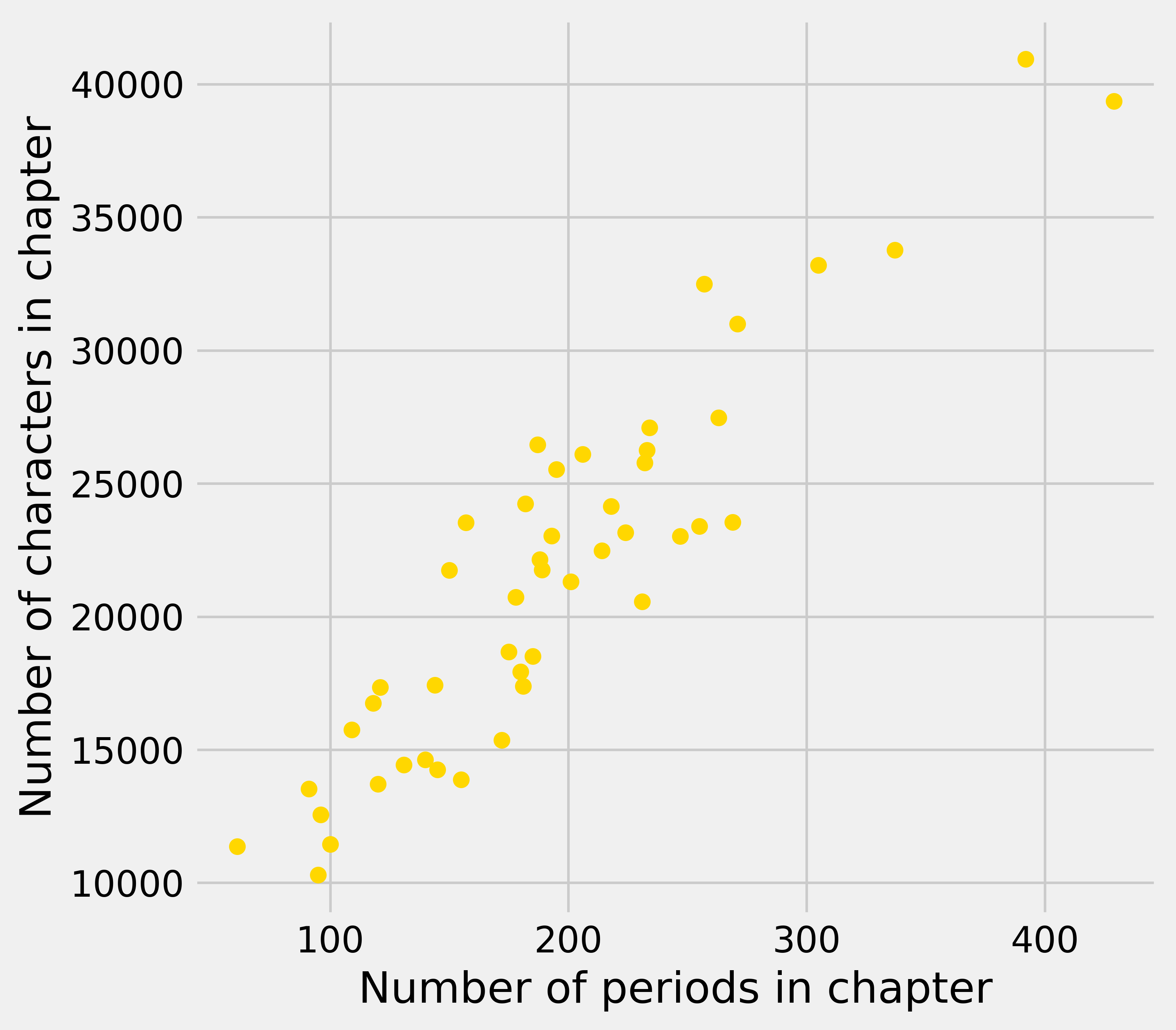
color='gold')

plots.xlabel('Number of periods in chapter')

plots.ylabel('Number of characters in chapter');

plots.savefig("scatter-plot.png",dpi=500,bbox\_inches='tight')

## **Result**



## 

* It is noticeable that how yellow points are roughly clustered around a straight line
* Now look at all the chapters that contain about 100 periods. The plot shows that those chapters contain about 10,000 characters to about 15,000 characters approximately. That’s about 100 to 150 characters per periods
* Indeed, it appears from looking at the plot that on average book tend to have somewhere between 100 and 150 characters between periods, as an approximate estimate.

**Task 2. Linear Regression – Case Study: Financial Data Analysis and Visualization**

**Action I. Give an initial analysis of the data frame by explaining different columns. During this analysis, you need to identify the FEATURES (X) that we can use to predict the response variable/target (y) that is “CLOSE” (column) value for the stock on a specific date.**

After examine the dataset,

* Open is the price of the stock at the beginning of the trading day.
* High is the highest price of the stock on that trading day.
* Low the lowest price of the stock on that trading day, and close the price of the stock at closing time.
* Volume indicates how many stocks were traded.
* Adjusted close is the closing price of the stock that adjusts the price of the stock for corporate actions.
* Stock splits (when the company makes each extant stock worth two and halves the price) and dividends (payout of company profits per share) also affect the price of a stock and should be accounted for.

**Code**

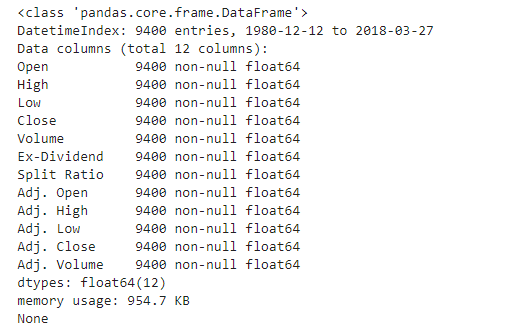
symbol = "WIKI/AAPL"

qndl.ApiConfig.api\_key="\_GQLzvh-scT2USyRMCaQ"

df=qndl.get(symbol)

print(df.info())

**Result**

****

**Action II. Separate the data into Features (X) and response variable (y).**

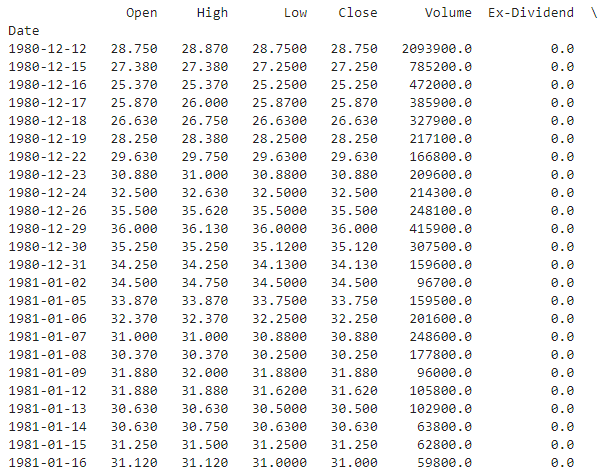
from pandas import DataFrame

X = df[['Open','High','Low','Volume','Ex-Dividend','Split Ratio','Adj. Open','Adj. High','Adj. Low','Adj. Close','Adj. Volume']]

Y = df['Close']

print (df)

**Result**

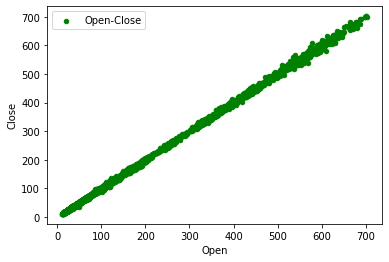
****

**Action III. Use a scatter-plot to plot the OPEN and CLOSE values.**

df.plot.scatter(x='Open',y='Close',color='g',label='Open-Close');

plots.show()

**Result**

****

**Action IV. Find the correlations between each identified feature with CLOSE and determine that these are positive or negative.**

regr = linear\_model.LinearRegression()

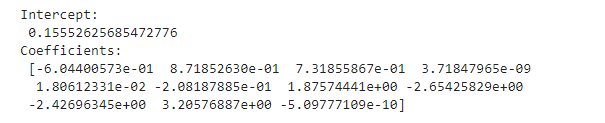
regr.fit(X, Y)

print('Intercept: \n', regr.intercept\_)

print('Coefficients: \n', regr.coef\_)

**Result**





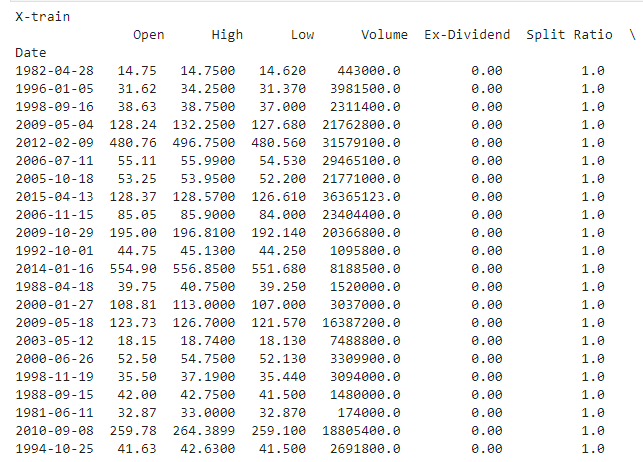
**Action V. Split the data into two parts in a ratio of 0.7 for training and 0.3 for testing.**

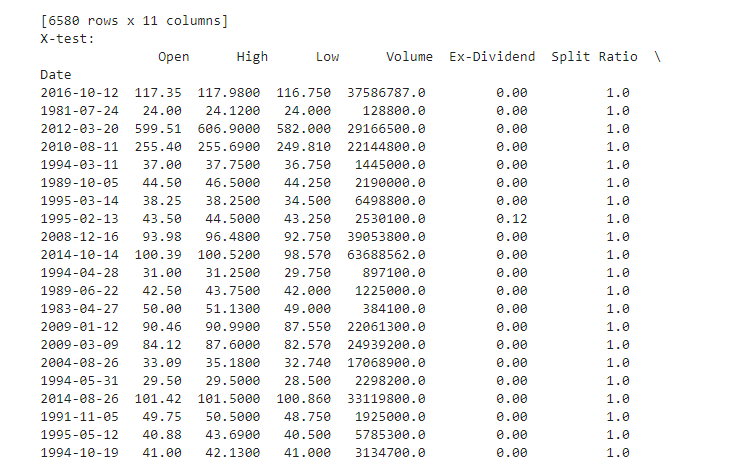
X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,Y,test\_size=0.3,random\_state=0)

print("X-train \n",X\_train)

print("X-test: \n", X\_test)

**Result**



****

**Action VI. Fit a multivariate linear regression model on the training data using all selected features.**

regr = linear\_model.LinearRegression()

regr.fit(X\_train,y\_train)

**Result**

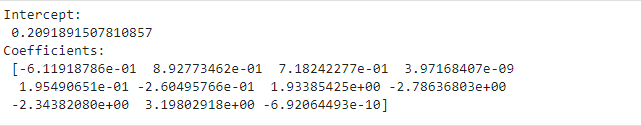


**Action VII. What are the intercept (𝜃0) and coefficients (𝜃1,2,…,𝜃𝑛) of the model?**

print('Intercept: \n', regr.intercept\_)

print('Coefficients: \n', regr.coef\_)

**Result**



**Action VIII. What is the 𝑅2 score for the training data and for the test data? What does this value indicate?**

yhat = regr.predict(X\_test)

SS\_Residual = sum((y\_test-yhat)\*\*2)

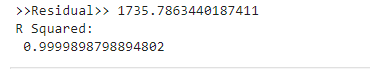
print(">>Residual>>",SS\_Residual)

SS\_Total = sum((Y-np.mean(Y))\*\*2)

r\_squared = 1 - (float(SS\_Residual))/SS\_Total

print('R Squared: \n',r\_squared)

**Result**



* R-squared is a statistical measure which indicates the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

**Action IX. Create set of imaginary values for the FEATURES, at least 3 values sets, and predict the CLOSE value for these imaginary FEATURES values. Report your predicted values.**

New\_predict\_x = X.iloc[7:16, :]

#print(New\_predict\_x)

New\_predict\_y = regr.predict(New\_predict\_x)

print(">>>>Value 1<<<<",New\_predict\_y)

New\_predict\_x = X.iloc[4:20, :]

#print(New\_predict\_x2)

New\_predict\_y = regr.predict(New\_predict\_x)

print(">>>>Value 2<<<<",New\_predict\_y)

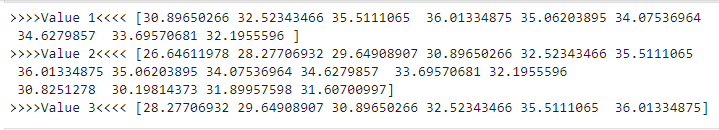
New\_predict\_x = X.iloc[5:11, :]

#print(New\_predict\_x3)

New\_predict\_y = regr.predict(New\_predict\_x)

print(">>>>Value 3<<<<",New\_predict\_y)

**Result**



**Action X. Re-instantiate your linear regression model with the parameter “fit\_intercept” set to “FALSE” and rerun you analysis on the entire features data (X). What are coefficients (𝜃1,2,…,𝜃𝑛) of the model?**

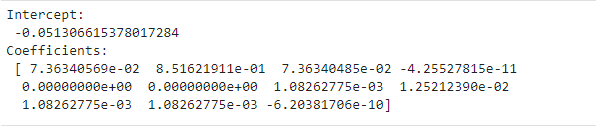
regr = linear\_model.LinearRegression()

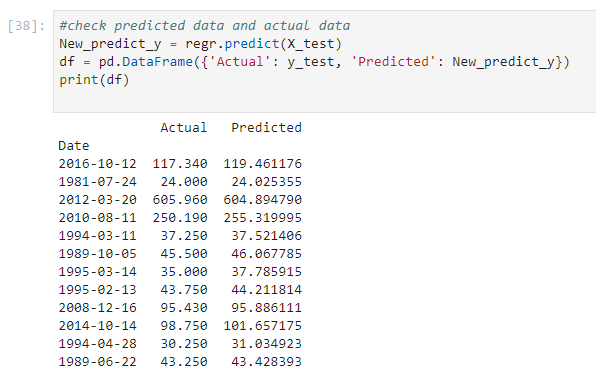
regr.fit(New\_predict\_x,New\_predict\_y)

print('Intercept: \n', regr.intercept\_)

print('Coefficients: \n', regr.coef\_)

**Result**





Therefore, after train model and applying new imaginary value, above code is shows that actual data and predicted data are approximately similar.

**Action XI. Calculate the coefficients (𝜃1,2,…,𝜃𝑛) analytical closed-form solution of linear regression. Make sure those estimates coincide with what you get in Action X to be certain you got them right. Use the matrix algebra functionality provided by library NUMPY to find the optimal vector 𝜃. Submit your code that you have used for verification.**

# Compute beta

Xt = np.transpose(X)

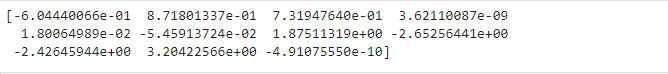
XtX = np.dot(Xt,X)

Xty = np.dot(Xt,Y)

beta = np.linalg.solve(XtX,Xty)

print(beta)

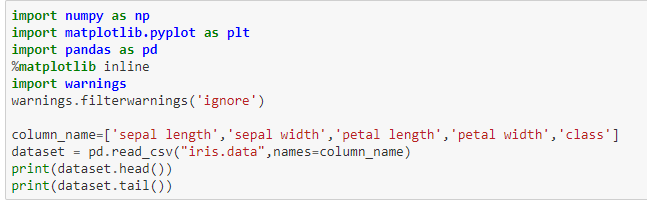
**Result**

****

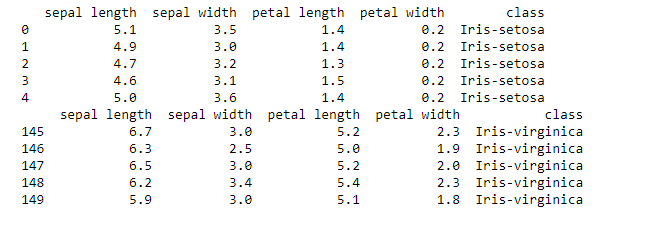
**Task 3. Data Clustering**

**Action I. View the dataset by using the information from the above link and briefly describe it.**

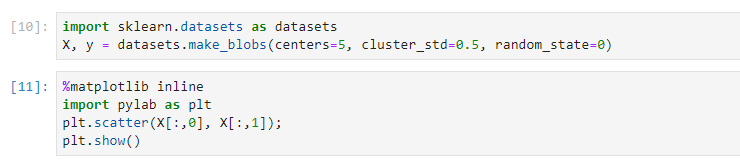
Clustering is the process of partitioning a group of data points into a small number of clusters (groups). For instance, the following dataset has different features of Iris plants such as their type, length, and width.



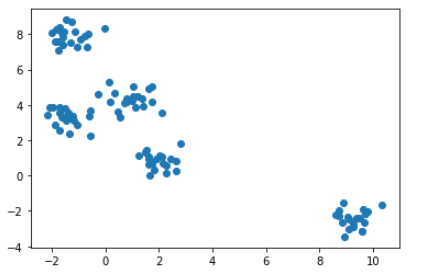
**Result**

****

**Action II. Read data and convert it into a suitable format for clustering.**

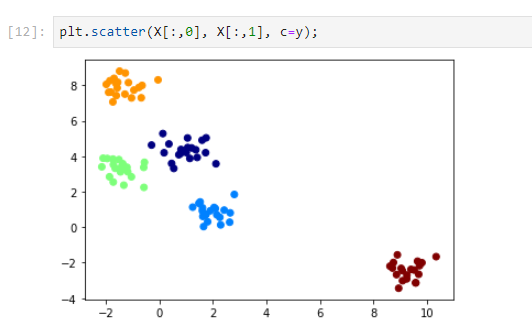
****

**Result**

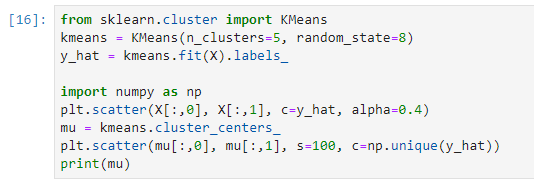
****

**Action III. For the k-means clustering algorithm, what will be value of “n\_clusters” parameter. Why? Perform the clustering using the K-means algorithm.**

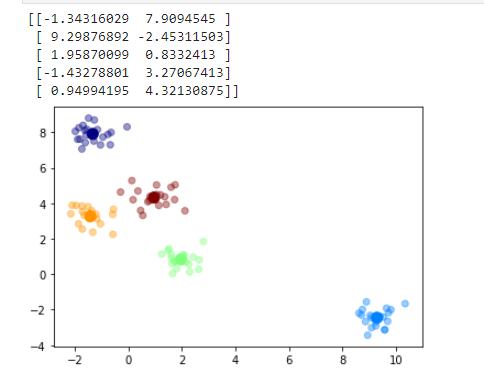
* The data looks like it may contain five different types of data point. In fact, this is how it was created above by using the centers parameter to the make\_blobs function equal to 5.
* We can plot the category information contained in vector Y as well, using the color parameter c.
* Noticed though, that this plot is just for visualization purposes because in the artificial data we know the right category of each point, in a real clustering unsupervised learning problem, you do not know the categories, **y**, of the data contained in **X**.

****

**Action IV. Plot the data for estimated clusters with different colour-coding.**

****

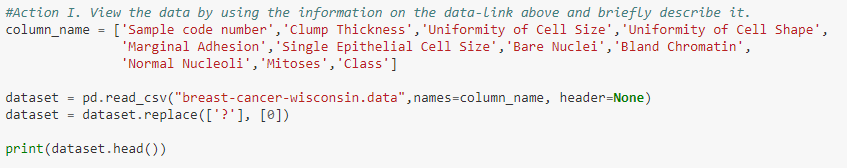
**Result**

****

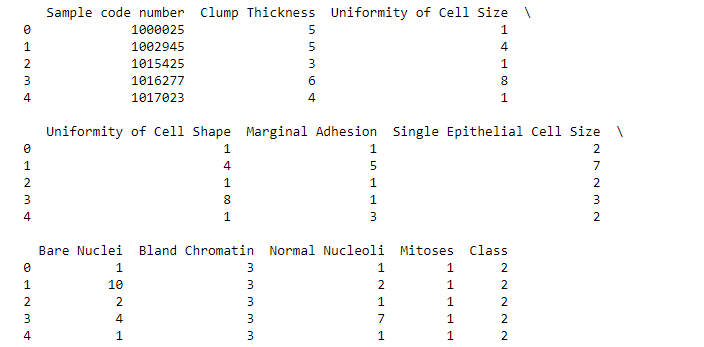
**Task 4. Data Classification**

**Action I. View the data by using the information on the data-link above and briefly describe it.**

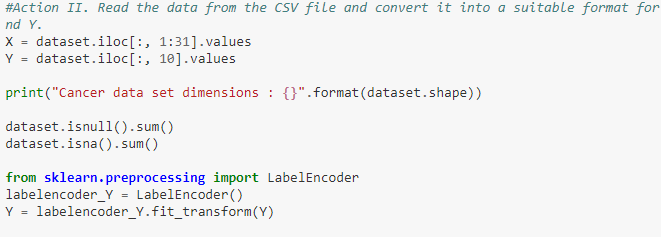
* After observing dataset, it can be said that which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyper parameter selection.
* The aim is to classify whether the breast cancer is benign or malignant.
* To achieve this I have used machine learning classification methods to fit a function that can predict the discrete class of new input.
* The given dataset is periodically therefore it reflects this chronological grouping of the data.



**Result**

****

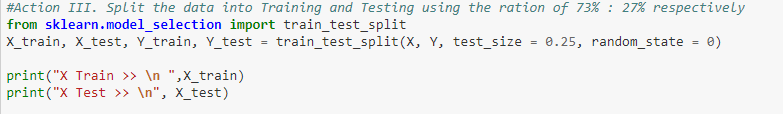
**Action II. Read the data from the CSV file and convert it into a suitable format for classification by separating them into X and Y.**

****

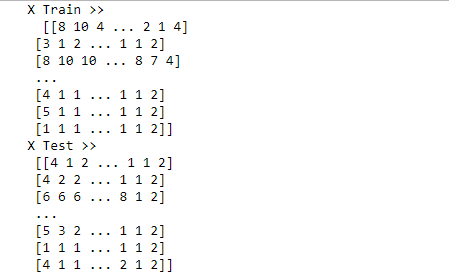
**Result**

****

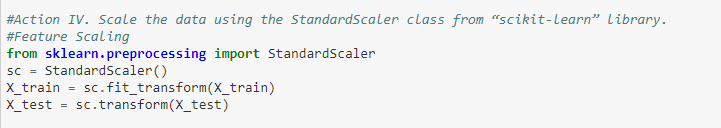
**Action III. Split the data into Training and Testing using the ration of 73% : 27% respectively.**

****

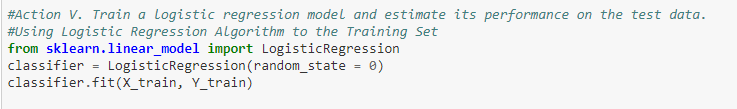
**Result**

****

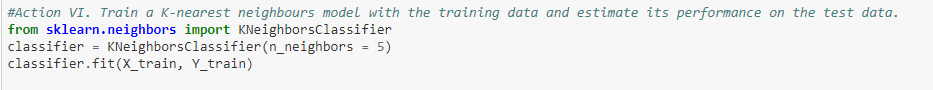
**Action IV. Scale the data using the StandardScaler class from “scikit-learn” library.**

****

**Action V. Train a logistic regression model and estimate its performance on the test data.**

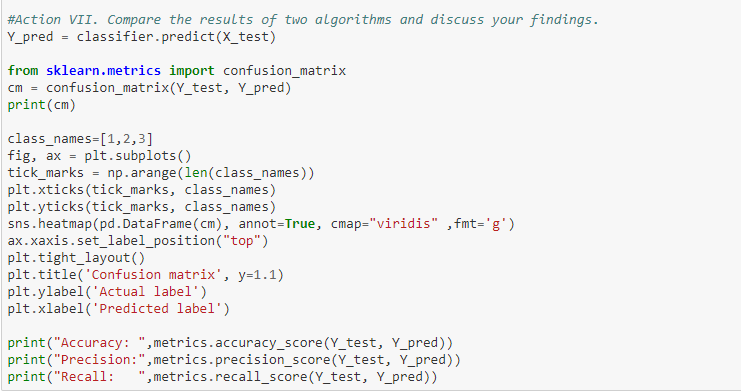
****

**Action VI. Train a K-nearest neighbours model with the training data and estimate its performance on the test data.**

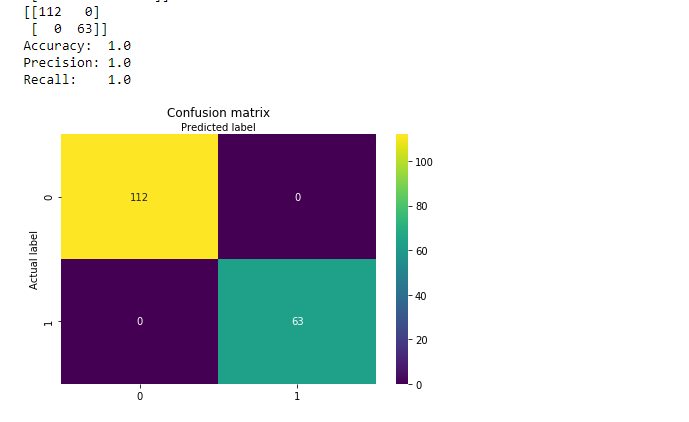
****

**Action VII. Compare the results of two algorithms and discuss your findings.**

* After applying Logistic regression model to train dataset, it gives 95.8% of accuracy, however K-nearest neighbor has also nearly same accuracy.
* To check the correct prediction we have to check confusion matrix object and add the predicted results diagonally which will be number of correct prediction and then divide by total number of predictions.

****

**Result**

****