Python for Deep Learning

LAB 1 Documentation

Lab ID: 7

Avinash Ganguri (6)

Nikhil Kantipudi (10)

Dileep Reddy Peddakam (19)

Introduction

Implemented several Machine learning techniques such as Classification, Regression, Natural Language Processing, dimensionality reduction and dealt with several python libraries such as Scikit learn and nltk, wordnet etc in the following 8 programs

Objective

To get optimized output for different programs based on ML techniques such as Classification, Natural Language Processing and Regression, etc...

Approaches/Methods

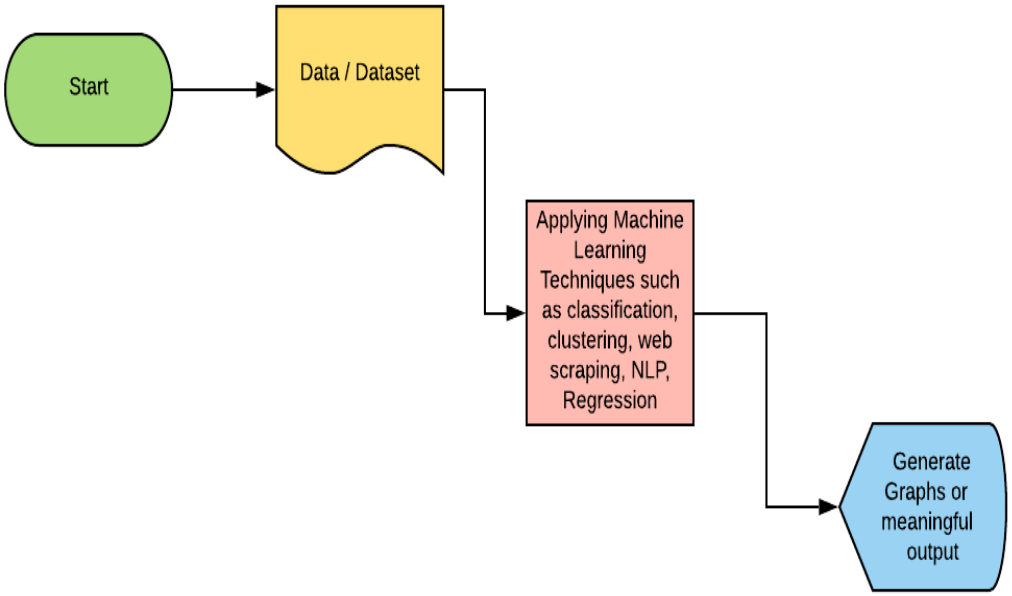
We used approaches or methods such as,

- Object oriented programming
- Web Scraping
- Data Analysis
- Classification algorithms such as Naïve Bayes, SVM and KNN
- Clustering algorithms like K-means, plotting Elbow graph
- Calculate silhouette score
- Applications of NLP
- Multiple Regression

Datasets

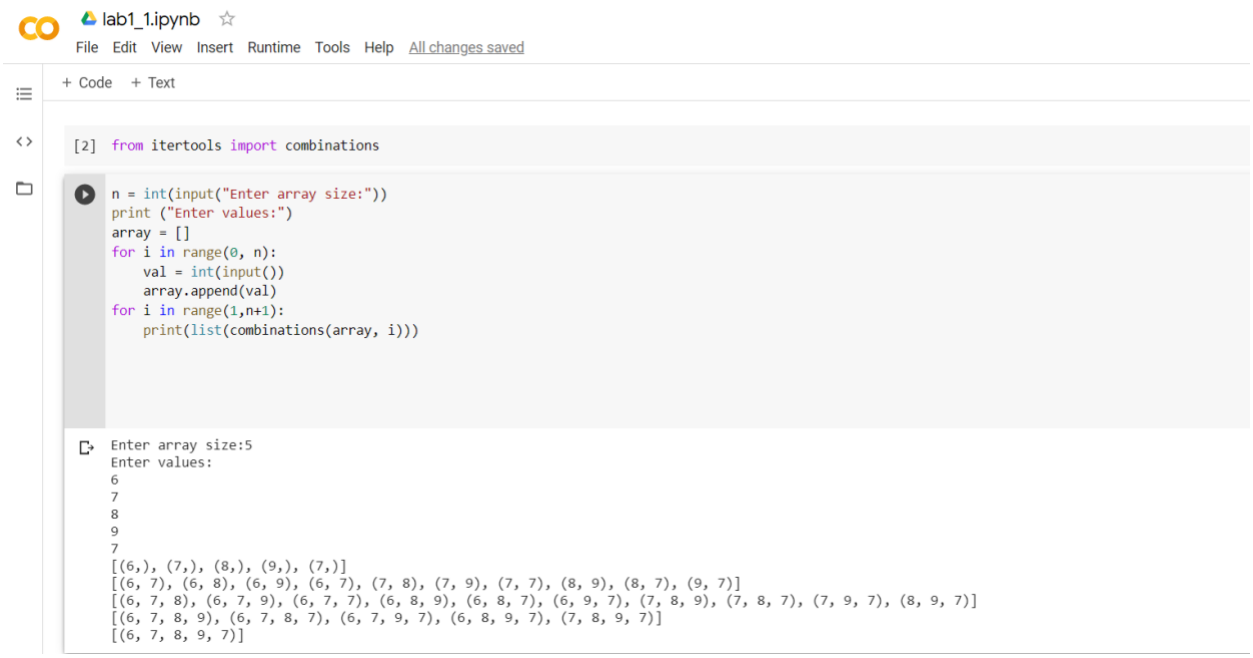
- For webscraping : https://catalog.umkc.edu/course-offerings/graduate/comp-sci/
- CC.CSV
- GLASS.CSV
- NPL_INPUT.TXT
- TRAIN.CSV
- WINEQUALITY-RED.CSV

Workflow


Start
Data / Dataset
Applying Machine Learning Techniques such as classification, clustering, web scraping, NLP, Regression
Generate Graphs or meaningful output

Evaluation of Programs

1)



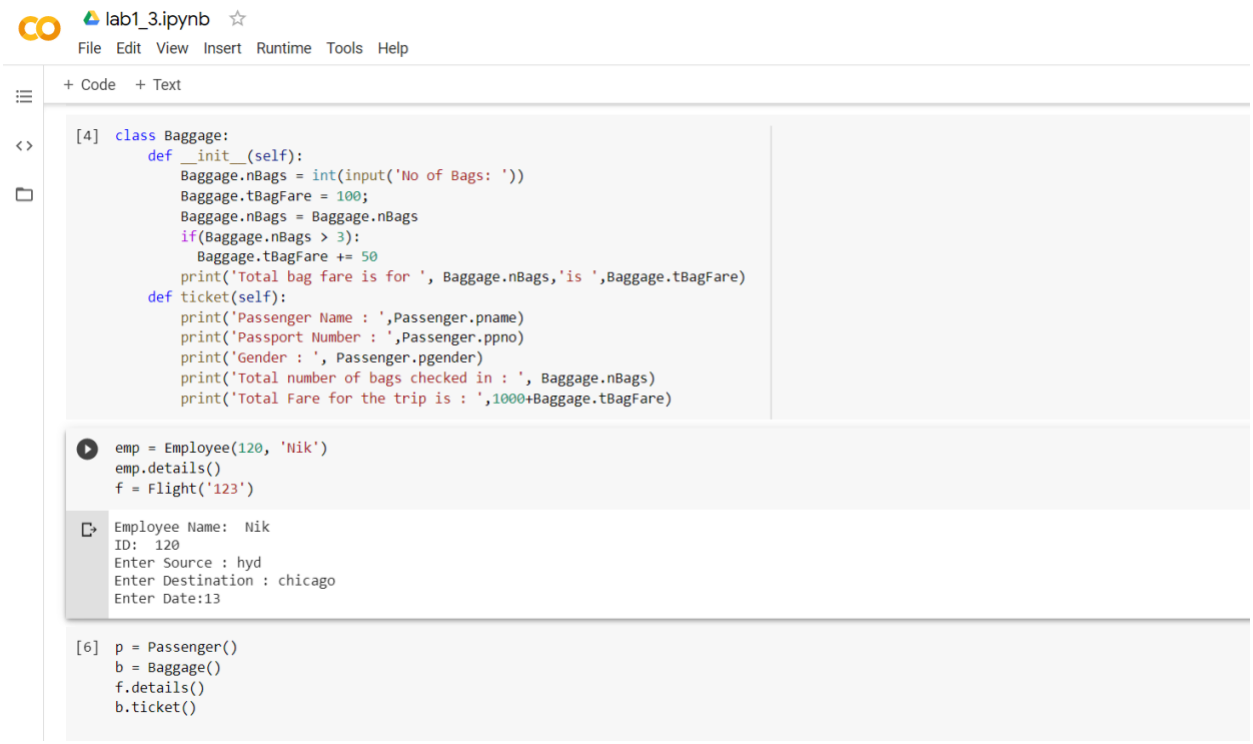
In this program we take values from the user which might contains the duplicate values. By using combinations on the arrays we displayed the subsets of the given numbers. Since we are using the combinations it doesn't display the null values.

2)

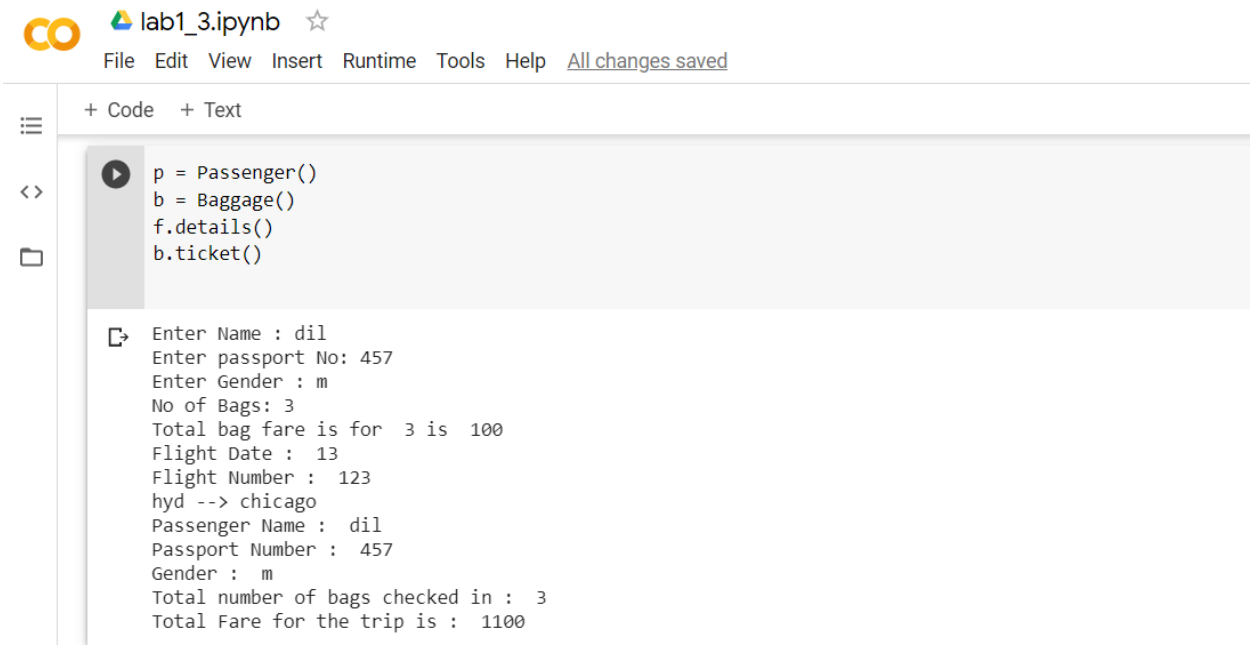
In this program we declared two dictionaries. Update method is used to merge the dictionaries. By using the lamda function and the sorted method we sorted these dictionaries based on the values.

3)

In this program, flight class accepts the input from the user and details method which displays the values and the second class which is employee will display the employee details by using the inheritance and method overloading. The passenger will take the input from the user.

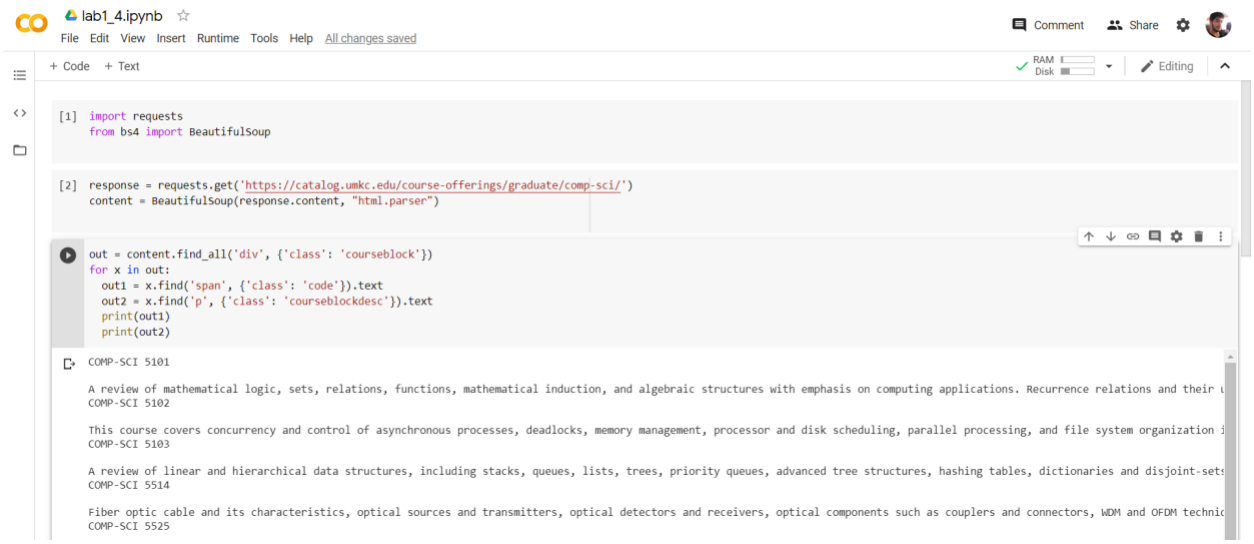


Here Baggage class take the input of number of bags so that it will calculate the fare



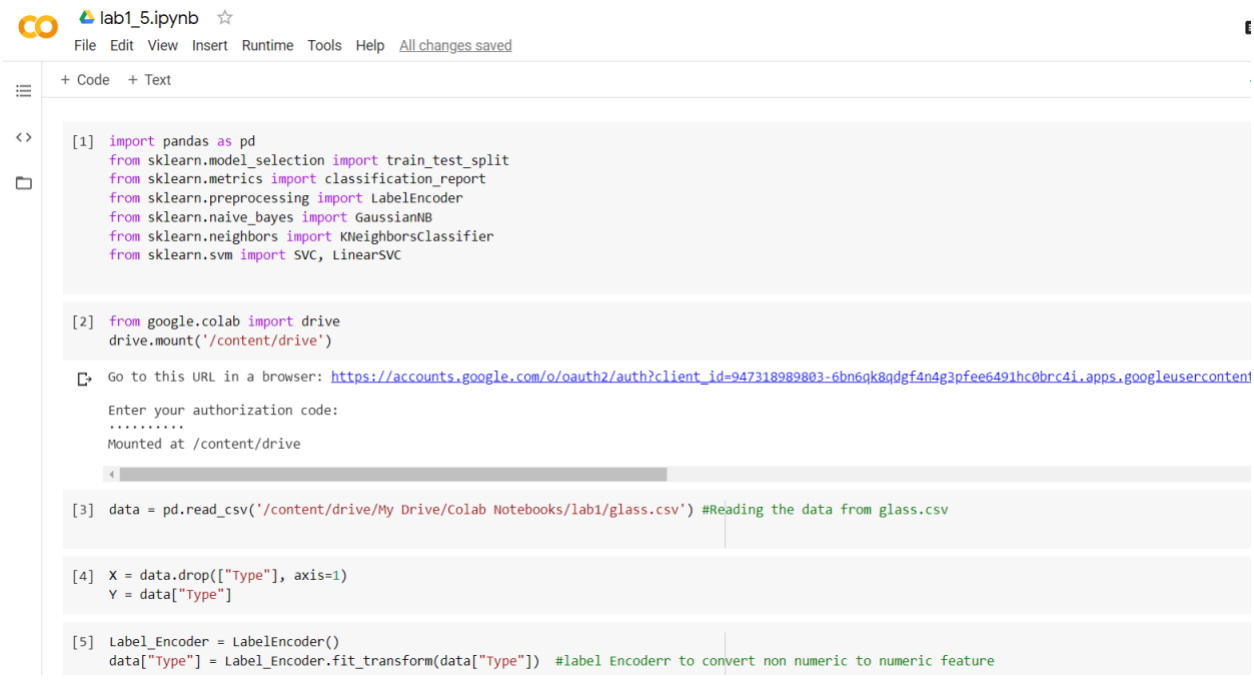
Finally ticket class takes all the details and it will give the fare of the trip.

4)



In this program we get the html link with the help of beautifulsoup package and convert that into text we give div tags with class courseblock then in loop to get course names and to display text without tags we use .text and the output is displayed.

5)



In this program we read the glass.csv from the drive and we drop the type column and we use the label encoder to convert the non numeric to numeric feature

```
[5] Label_Encoder = LabelEncoder()
    data["Type"] = Label_Encoder.fit_transform(data["Type"])  #label Encoderr to convert non numeric to numeric feature
```

```
data=data.apply(lambda x: x.fillna(x.mean()),axis=0)   #Filling null values with mean using lamda function
print(data.isnull().sum())
```

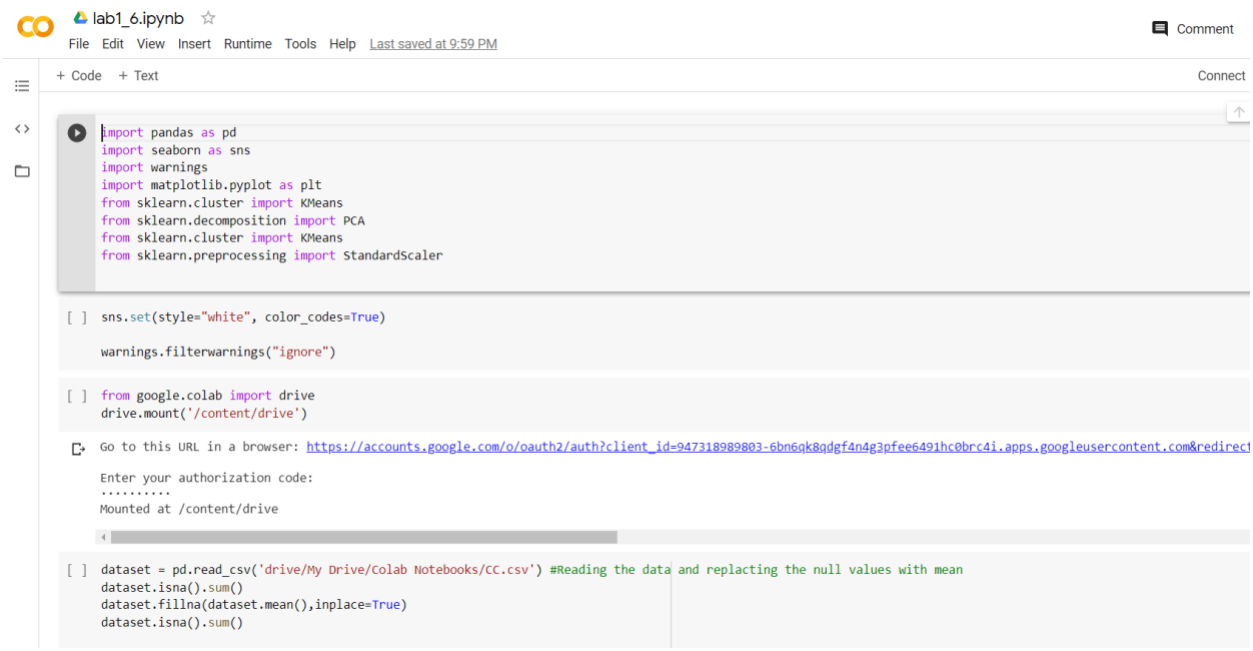
```
RI      0
Na      0
Mg      0
Al      0
Si      0
K       0
Ca      0
Ba      0
Fe      0
Type    0
dtype: int64
```

Here we print the data

Here we split the train data and test data. And we use fit function to fit the data and we use .predict to predict the data instances and print the outputs for naïve bayes, svm and knn algorithms.

So Here Knn algorithm gives the better result than remaining.

6)



lab1_6.ipynb

File Edit View Insert Runtime Tools Help Last saved at 9:59 PM

Comment

+ Code + Text

Connect

```
import pandas as pd
import seaborn as sns
import warnings
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

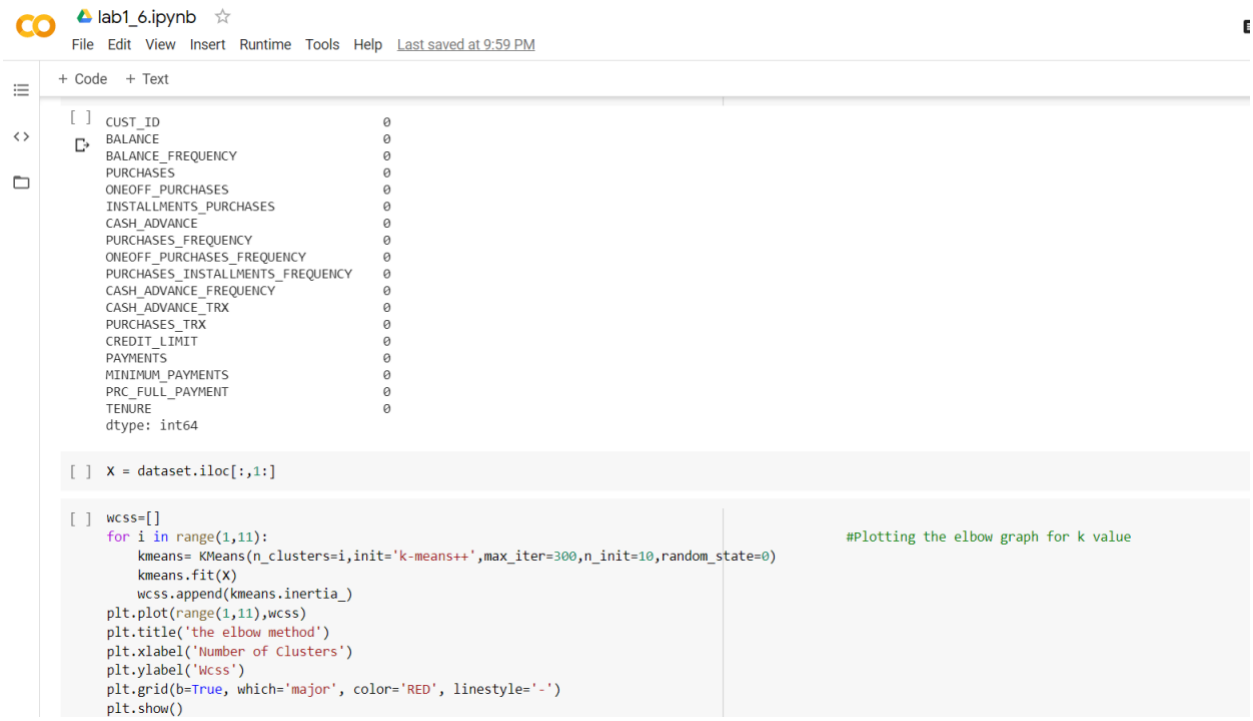
```
[ ] sns.set(style="white", color_codes=True)

    warnings.filterwarnings("ignore")
```

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect

Enter your authorization code:
··········
Mounted at /content/drive
```

```
[ ] dataset = pd.read_csv('drive/My Drive/Colab Notebooks/CC.csv') #Reading the data and replacting the null values with mean
    dataset.isna().sum()
    dataset.fillna(dataset.mean(),inplace=True)
    dataset.isna().sum()
```

In this program we read the cc.csv from the drive and replace the null the values with mean by using .mean().



lab1_6.ipynb

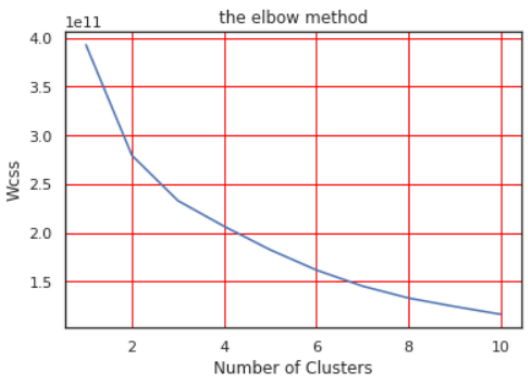
File Edit View Insert Runtime Tools Help Last saved at 9:59 PM

+ Code + Text

```
[ ] CUST_ID                          0
    BALANCE                          0
    BALANCE_FREQUENCY                0
    PURCHASES                        0
    ONEOFF_PURCHASES                 0
    INSTALLMENTS_PURCHASES           0
    CASH_ADVANCE                     0
    PURCHASES_FREQUENCY              0
    ONEOFF_PURCHASES_FREQUENCY       0
    PURCHASES_INSTALLMENTS_FREQUENCY 0
    CASH_ADVANCE_FREQUENCY           0
    CASH_ADVANCE_TRX                 0
    PURCHASES_TRX                    0
    CREDIT_LIMIT                     0
    PAYMENTS                         0
    MINIMUM_PAYMENTS                 0
    PRC_FULL_PAYMENT                 0
    TENURE                           0
    dtype: int64
```

```
[ ] X = dataset.iloc[:,1:]
```

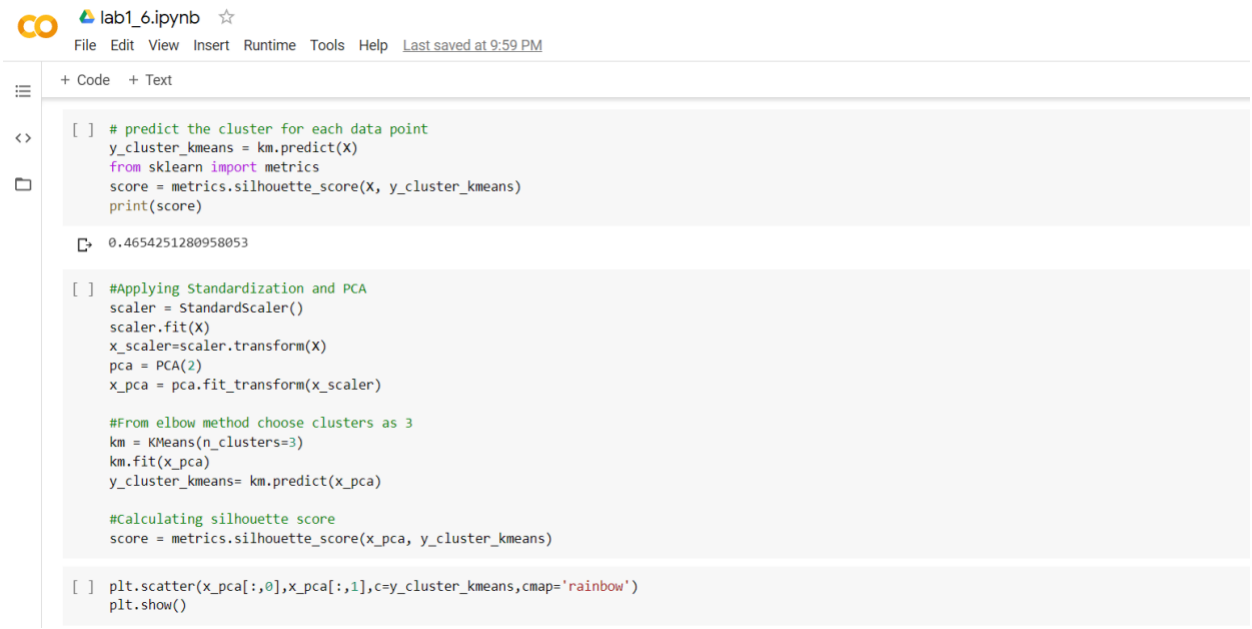
```
[ ] wcss=[]
    for i in range(1,11):                                                    #Plotting the elbow graph for k value
        kmeans= KMeans(n_clusters=i,init='k-means++',max_iter=300,n_init=10,random_state=0)
        kmeans.fit(X)
        wcss.append(kmeans.inertia_)
    plt.plot(range(1,11),wcss)
    plt.title('the elbow method')
    plt.xlabel('Number of Clusters')
    plt.ylabel('Wcss')
    plt.grid(b=True, which='major', color='RED', linestyle='-')
    plt.show()
```




```
[ ]  # K value is 3
     ##building the model
    from sklearn.cluster import KMeans
    nclusters = 3 # this is the k in kmeans
    km = KMeans(n_clusters=nclusters)
    km.fit(X)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

Here we plot the elbow graph for k value by using the .plot(). Once the value of k is known we build the model with the k value.




```
[ ] # predict the cluster for each data point
    y_cluster_kmeans = km.predict(X)
    from sklearn import metrics
    score = metrics.silhouette_score(X, y_cluster_kmeans)
    print(score)
```

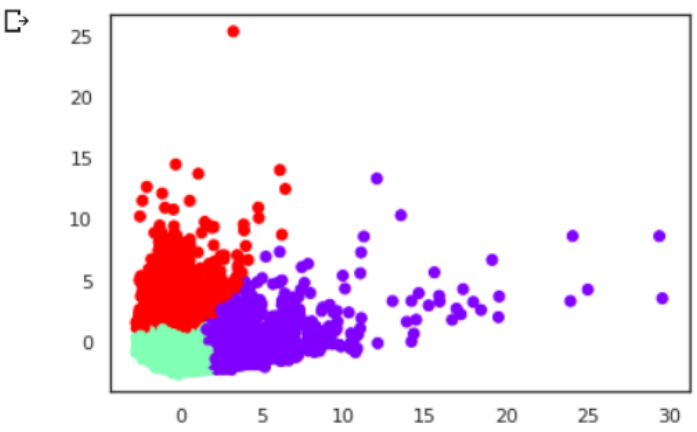
```
0.4654251280958053
```

```
[ ] #Applying Standardization and PCA
    scaler = StandardScaler()
    scaler.fit(X)
    x_scaler=scaler.transform(X)
    pca = PCA(2)
    x_pca = pca.fit_transform(x_scaler)

    #From elbow method choose clusters as 3
    km = KMeans(n_clusters=3)
    km.fit(x_pca)
    y_cluster_kmeans= km.predict(x_pca)

    #Calculating silhouette score
    score = metrics.silhouette_score(x_pca, y_cluster_kmeans)
```

```
[ ] plt.scatter(x_pca[:,0],x_pca[:,1],c=y_cluster_kmeans,cmap='rainbow')
    plt.show()
```

And we evaluate the silhouette score by using the silhouette_score(), and we print the score. After that we apply stardardization and PCA

```
[ ]  plt.scatter(x_pca[:,0],x_pca[:,1],c=y_cluster_kmeans,cmap='rainbow')
     plt.show()
```



7)

lab1_7.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
[ ]  import nltk
     from nltk.util import ngrams
     from nltk.stem import PorterStemmer
     from nltk.stem import WordNetLemmatizer
     from nltk import wordpunct_tokenize, pos_tag, ne_chunk
     import requests
     from bs4 import BeautifulSoup
```

```
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')
nltk.download('maxent_ne_chunker')
nltk.download('words')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Package words is already up-to-date!
True
```

```
[ ]  from google.colab import drive
     drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[ ]  f =open("/content/drive/My Drive/Colab Notebooks/lab1/nlp_input.txt","r", encoding="cp1252")
     text=f.read()
```

In this program we download nltk files and we read the nlp_input.txt file.

```
lemmatizer=WordNetLemmatizer()
for w in wtokens:
  print(lemmatizer.lemmatize(w))
```

```
and
lambda
pair
and
we
can
see
which
pair
ha
the
lowest
associated
error
.
We
can
see
```

We perform lemmatization which is similar to stemming but gives meaningful words

```
wtokens=nltk.word_tokenize(text)
stokens=nltk.sent_tokenize(text)
for w in wtokens:
  print(w)
```

```
alpha
and
lambda
pairs
and
we
can
see
which
pair
has
the
lowest
associated
error
.
We
can
see
that
the
R
mean-squared
values
```

Here wtokens gives each word as a token from the text.

```
[ ] trigrams=list(ngrams(wtokens,3))
    print(trigrams)
```

```
[('Regression', 'analysis', 'is'), ('analysis', 'is', 'a'), ('is', 'a', 'statistical'), ('a', 'statistical', 'technique'), ('statistical', 'technique', 'that'),
```

Here trigrams gives the consecutive sequence of 3 words from the sentence.

```
[ ] wfrequency = nltk.FreqDist(trigrams)

    Topten = wfrequency.most_common(10)

    print("Top 10 Trigrams:\n", Topten)
```

```
Top 10 Trigrams:
 [(('we', 'need', 'to'), 3), (('the', 'coefficients', '.'), 3), (('?', '?', '='), 3), (('to', 'find', 'the'), 2), (('find', 'the', 'optimal'), 2), (('over', 'a', 'number'), 2),
```

It displays most repeated trigrams with their count.

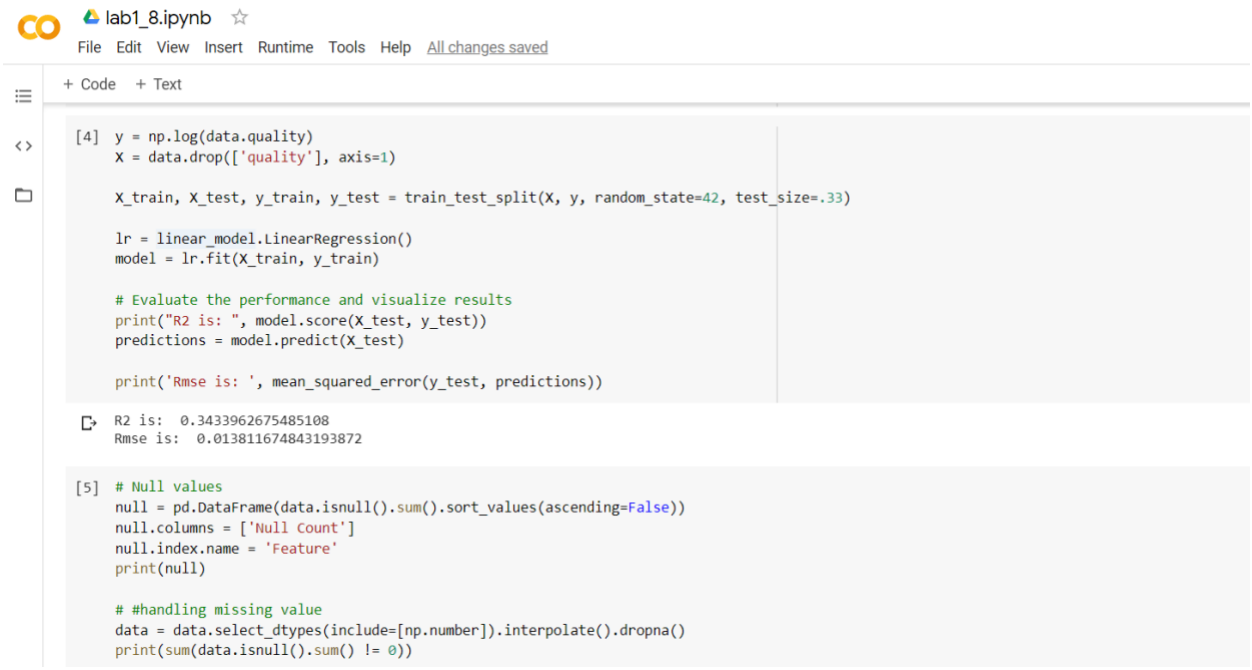
```
l=[]
for w in stokens:
  for a,b in Topten:
    if a in ngrams(w.split(),3):
      print(a,w)
      l.append(w)
      break
print("concatinated Text")
print(l)
```

```
('we', 'need', 'to') First we need to understand the basics of regression and what parameters of the equation are changed when using a specific model.
('to', 'find', 'the') Visualization of the squared error (from Setosa.io)
The equation for this model is referred to as the cost function and is a way to find the optimal error by minimizing and measuring it.
('to', 'find', 'the') The gradient descent algorithm is used to find the optimal cost function by going over a number of iterations.
('we', 'need', 'to') But the data we need to define and analyze is not always so easy to characterize with the base OLS model.
('to', 'each', 'other') Equation for least ordinary squares
One situation is the data showing multi-collinearity, this is when predictor variables are correlated to each other and to the response variable.
('penalty', 'term', 'to') To produce a more accurate model of complex data we can add a penalty term to the OLS equation.
('penalty', 'term', 'to') These are known as L1 regularization(Lasso regression) and L2 regularization(ridge regression).The best model we can hope to come up with minimizes both th
Ridge regression uses L2 regularization which adds the following penalty term to the OLS equation.
('we', 'need', 'to') = 1 denotes lasso)
Performing Elastic Net regression
Performing Elastic Net requires us to tune parameters to identify the best alpha and lambda values and for this we need to use the caret package.
('over', 'a', 'number') We will tune the model by iterating over a number of alpha and lambda pairs and we can see which pair has the lowest associated error.
concatinated Text
['First we need to understand the basics of regression and what parameters of the equation are changed when using a specific model.', 'Visualization of the squared error (from Setos
```

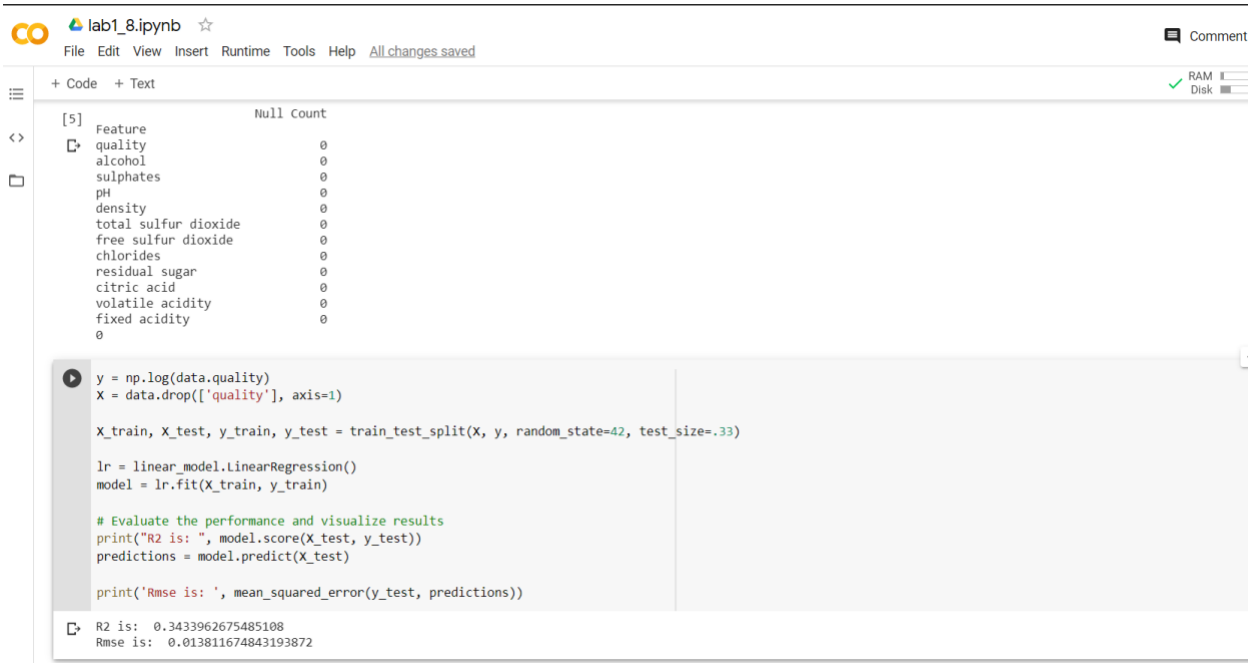
Here the above text file is taken and finding the repeated trigrams and append is used to concatenate. Finally we display the concatenated results.

8)

In this program we read the winequality-red.csv from the drive.



we drop the quality field. And we split the train data and test data. We use .score and .predict to evaluate the performance and visualize the results. Isnull() is used for null values.



In this field we evaluate the model using RMSE and R2 and the results are same

Conclusion

We have learned several methods and approaches while implementing lab programs such as Classification, Clustering, Web scraping, Data Visualization, application of Natural Language Processing and Multiple Regression.