CHAPTER 1

ORGANISATIONAL OVERVIEW

1.1 INTRODUCTION

Brillica Services Pvt. Ltd. is the Best Technology Provider in India. WE provide specialized courses in Data Science, Internet of things, Python, Machine Learning with Python, Java, CISCO, Artificial Intelligence, Android App and Development, Microsoft and many other technologies with Live Projects.

Here students don’t just complete the course with a single project, they complete their course with a deep practical knowledge and multiple projects.

Brillica Services has successfully executed projects in African continent and is still working with them towards providing emerging technology solutions for various corporates.

Our Associations

Brillica Services is proudly Associated with MICROSOFT, INTEL and IBM.
As we are the only associated Business and Education partner with Microsoft, Intel & IBM in Uttarakhand, our every student gets certification of MICROSOFT & IBM.

Healthy Environment is a key to success!

- Brillica Services is the Place where Trainer and the Students share a comfortable workplace which makes their efforts successful.
- We give every student and employee a fair time to share their views and thoughts so that we could make them doubt free.

CHAPTER 2

INTRODUCTION TO PROJECT

2.1 PURPOSE:

Spam e-mails can be not only annoying but also dangerous to consumers. It can be defined as first one Anonymity, second one Mass Mailings and third one Unsolicited.

Spam e-mail are message randomly sent to multiple addresses by all sorts of groups, but mostly lazy advertisers and criminals who wish to lead you to phishing sites.

2.2 OBJECTIVE:

The main objective is to give knowledge to the user about the fake e-mails and Relevant e-mails and also to classify that mail is spam or not.

2.3 MOTIVATION:

- Unwanted e-mails irritating internet connection
- Critical e-mail message is missed or delayed.
- Millions of compromised computers.
- Billions of dollars lost worldwide.
- Identity theft.
- Spam can crash mail servers and fill up hard drives.

2.4 DEFINITION:

This project is to classify whether the e- mail is ham or spam. I developed a machine learning model and trained it on the data of e-mail on two different criteria. I collected data from sites and grouped into two units for training and test in 7:3 ratios.

I moved to model designing using machine learning algorithms namely KNN Classifier.

CHAPTER 3

FEASIBILITY STUDY

Feasibility studies aim to objectively and rationally uncover the strengths and weakness of an existing business or proposed venture, opportunities and threats present in environment, the resources required to carry through and ultimately the prospects for success. In simple terms the likelihood that the system will be useful to the organization and it is an important outcome of the preliminary investigation. It is divided into following categories.

- Economic Feasibility
- Technical Feasibility
- Operational Feasibility

ECONOMICAL FEASIBILITY:

The economic feasibility step of business development is that period during which break-even financial model of the business venture is developed based on all costs associated with taking the product from idea to market and achieving sales sufficient to satisfy debt or investment requirements. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed System will provide. It includes qualification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

TECHNICAL FEASIBILITY:

It is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of proposed system. It is based on the outline design of system requirements in terms of input, process, outputs, fields, programs, procedures. This can be quantified in terms of volumes of data, trends and frequencies of updating etc. In order to estimate whether the new system will perform adequately or not. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

OPERATIONAL FEASIBILITY:

It is a measure of how well a proposed system solves the problems and takes advantage of opportunities identified during scope definition and how it satisfies the requirement identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.

CHAPTER 4

EXTERNAL INTERFACE REQIRMENTS:

4.1 HARDWARE INTERFACE:

Processor: Intel® Core™ i7-6500U CPU @ 2.50GHz 2.60GHz
Installed memory (RAM): 16.00 GB
System type: 64-bit operating system, x64-based processor
Graphics Video Card: 2.00GB

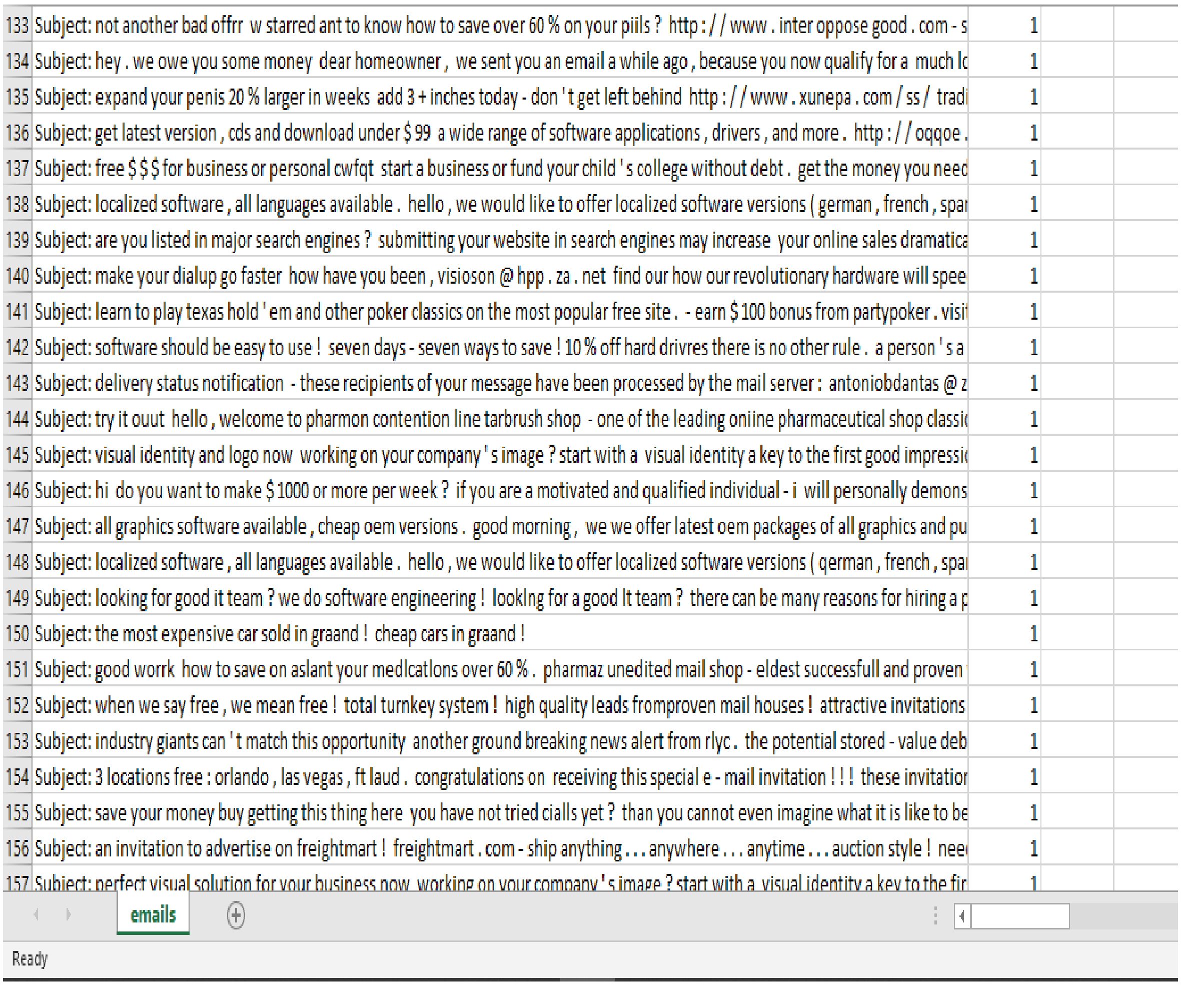
4.2 SOFTWARE INTERFACE:

Platform: OS - Windows 8 and above/Mac/Linux
Tools: Anaconda and Jupyter Notebooks

CHAPTER 5

MODULES OF THE PROJECT

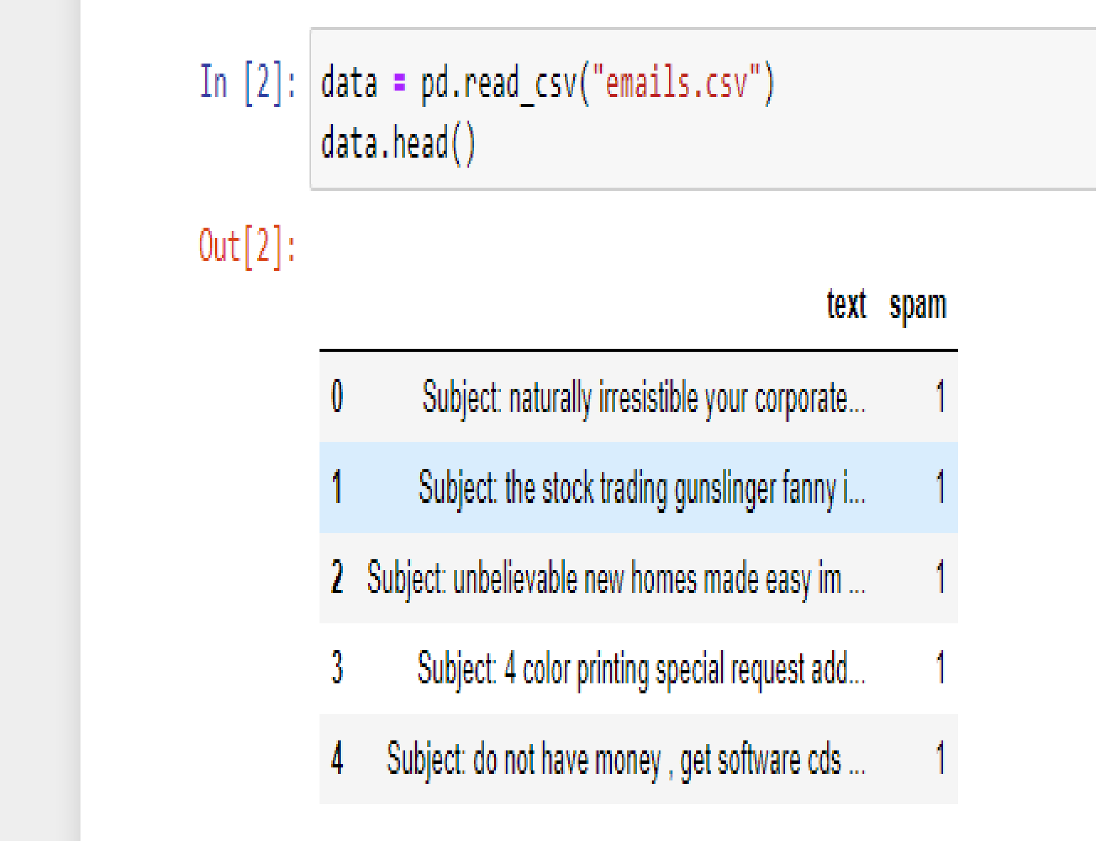
1. Collecting Data



133	Subject: not another bad offrr w starred ant to know how to save over 60 % on your piils ? http : / / www . inter oppose good . com - s	1
134	Subject: hey . we owe you some money dear homeowner , we sent you an email a while ago , because you now qualify for a much l	1
135	Subject: expand your penis 20 % larger in weeks add 3 + inches today - don ' t get left behind http : / / www . xunepa . com / ss / tradi	1
136	Subject: get latest version , cds and download under $ 99 a wide range of software applications , drivers , and more . http : / / oqqoe .	1
137	Subject: free $ $ $ for business or personal cwfqt start a business or fund your child ' s college without debt . get the money you need	1
138	Subject: localized software , all languages available . hello , we would like to offer localized software versions (german , french , spa	1
139	Subject: are you listed in major search engines ? submitting your website in search engines may increase your online sales dramatica	1
140	Subject: make your dialup go faster how have you been , visioson @ hpp . za . net find our how our revolutionary hardware will spee	1
141	Subject: learn to play texas hold ' em and other poker classics on the most popular free site . - earn $ 100 bonus from partypoker . visi	1
142	Subject: software should be easy to use ! seven days - seven ways to save ! 10 % off hard drivres there is no other rule . a person ' s a	1
143	Subject: delivery status notification - these recipients of your message have been processed by the mail server : antoniobdantas @ z	1
144	Subject: try it ouut hello , welcome to pharmon contention line tarbrush shop - one of the leading oniine pharmaceutical shop classi	1
145	Subject: visual identity and logo now working on your company ' s image ? start with a visual identity a key to the first good impressi	1
146	Subject: hi do you want to make $ 1000 or more per week ? if you are a motivated and qualified individual - i will personally demons	1
147	Subject: all graphics software available , cheap oem versions . good morning , we we offer latest oem packages of all graphics and pu	1
148	Subject: localized software , all languages available . hello , we would like to offer localized software versions (qerman , french , spa	1
149	Subject: looking for good it team ? we do software engineering ! looklng for a good It team ? there can be many reasons for hiring a p	1
150	Subject: the most expensive car sold in graand ! cheap cars in graand !	1
151	Subject: good worrk how to save on aslant your medlcatlons over 60 % . pharmaz unedited mail shop - eldest successfull and proven	1
152	Subject: when we say free , we mean free ! total turnkey system ! high quality leads fromproven mail houses ! attractive invitations	1
153	Subject: industry giants can ' t match this opportunity another ground breaking news alert from rlyc . the potential stored - value deb	1
154	Subject: 3 locations free : orlando , las vegas , ft laud . congratulations on receiving this special e - mail invitation ! ! ! these invitatio	1
155	Subject: save your money buy getting this thing here you have not tried cialls yet ? than you cannot even imagine what it is like to be	1
156	Subject: an invitation to advertise on freightmart ! freightmart . com - ship anything . . . anywhere . . . anytime . . . auction style ! nee	1
157	Subject: perfect visual solution for your business now working on your company ' s image ? start with a visual identity a key to the fir	1

Figure 5.1 Data Sample

2. Reading Data



In [2]:

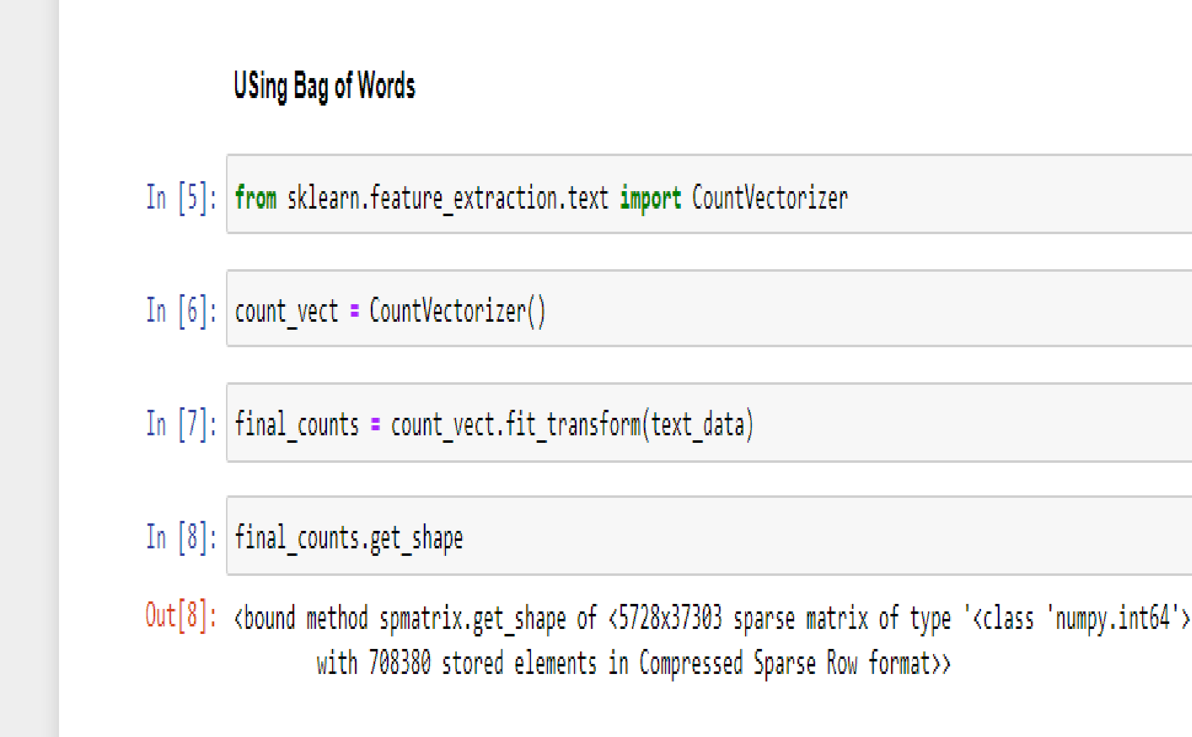
```
data = pd.read_csv("emails.csv")
data.head()
```

Out[2]:

	text	spam
0	Subject: naturally irresistible your corporate...	1
1	Subject: the stock trading gunslinger fanny i...	1
2	Subject: unbelievable new homes made easy im ...	1
3	Subject: 4 color printing special request add...	1
4	Subject: do not have money , get software cds ...	1

Figure 5.2 First 5 Rows of Data

3. Converting words into numerals



USing Bag of Words

```
In [5]: from sklearn.feature_extraction.text import CountVectorizer
```

```
In [6]: count_vect = CountVectorizer()
```

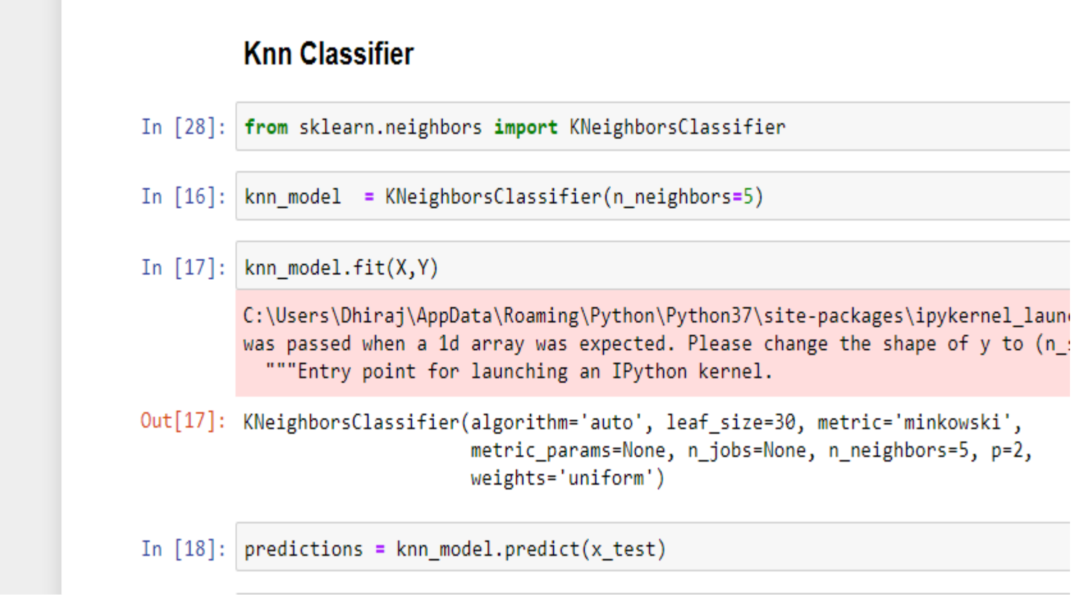
```
In [7]: final_counts = count_vect.fit_transform(text_data)
```

```
In [8]: final_counts.get_shape
```

```
Out[8]: <bound method spmatrix.get_shape of <5728x37303 sparse matrix of type '<class 'numpy.int64'>
            with 708380 stored elements in Compressed Sparse Row format>>
```

Figure 5.3 Bag of Words

4. Training Model on KNN Classifier

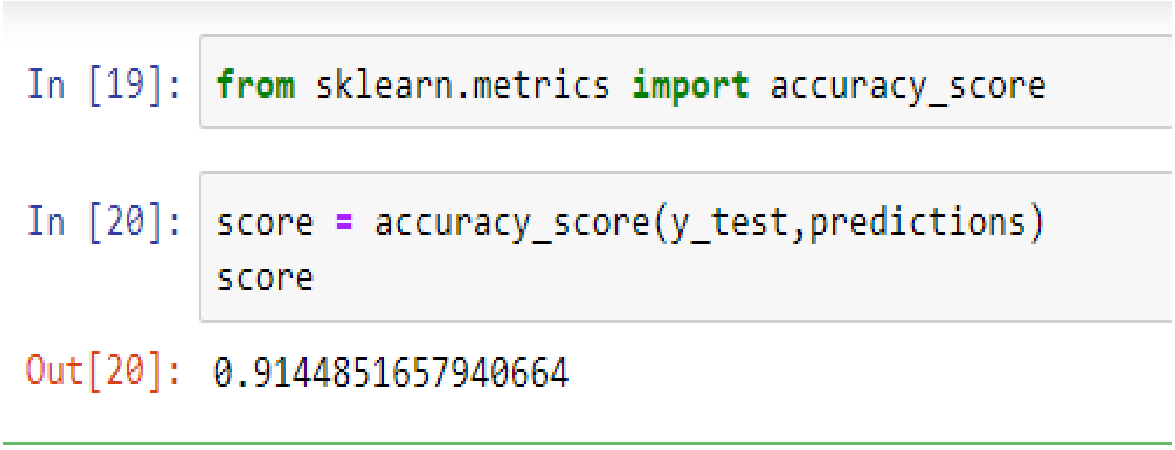


Knn Classifier

```
In [28]: from sklearn.neighbors import KNeighborsClassifier

In [16]: knn_model  = KNeighborsClassifier(n_neighbors=5)

In [17]: knn_model.fit(X,Y)

         C:\Users\Dhiraj\AppData\Roaming\Python\Python37\site-packages\ipykernel_laun
         was passed when a 1d array was expected. Please change the shape of y to (n_
           """Entry point for launching an IPython kernel.

Out[17]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                              metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                              weights='uniform')

In [18]: predictions = knn_model.predict(x_test)
```

Figure 5.4

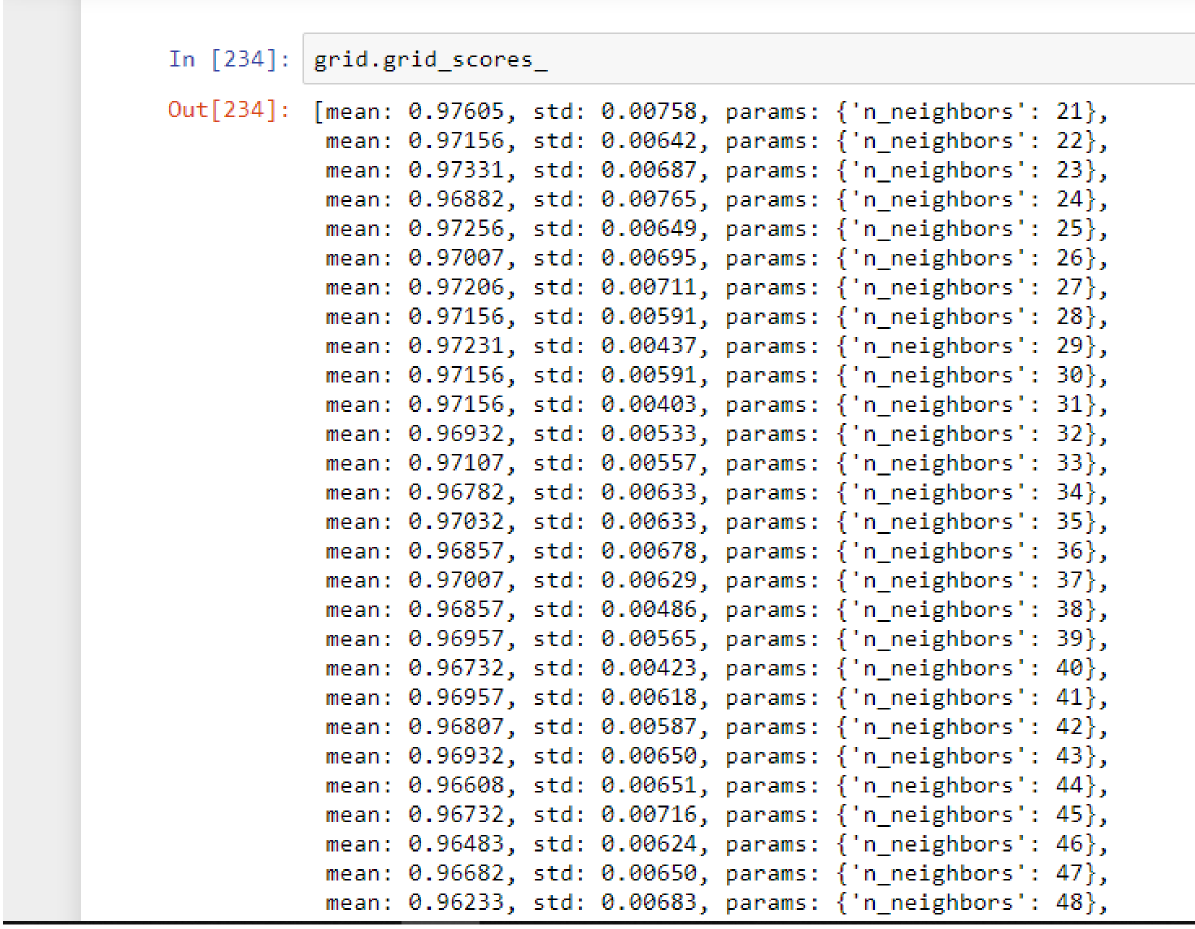
5. Checking Model Performance



```
In [19]: from sklearn.metrics import accuracy_score

In [20]: score = accuracy_score(y_test,predictions)
         score

Out[20]: 0.914485165794066 4
```

Figure 5.5

6. Improving the model performance



```
In [234]: grid.grid_scores_

Out[234]: [mean: 0.97605, std: 0.00758, params: {'n_neighbors': 21},
           mean: 0.97156, std: 0.00642, params: {'n_neighbors': 22},
           mean: 0.97331, std: 0.00687, params: {'n_neighbors': 23},
           mean: 0.96882, std: 0.00765, params: {'n_neighbors': 24},
           mean: 0.97256, std: 0.00649, params: {'n_neighbors': 25},
           mean: 0.97007, std: 0.00695, params: {'n_neighbors': 26},
           mean: 0.97206, std: 0.00711, params: {'n_neighbors': 27},
           mean: 0.97156, std: 0.00591, params: {'n_neighbors': 28},
           mean: 0.97231, std: 0.00437, params: {'n_neighbors': 29},
           mean: 0.97156, std: 0.00591, params: {'n_neighbors': 30},
           mean: 0.97156, std: 0.00403, params: {'n_neighbors': 31},
           mean: 0.96932, std: 0.00533, params: {'n_neighbors': 32},
           mean: 0.97107, std: 0.00557, params: {'n_neighbors': 33},
           mean: 0.96782, std: 0.00633, params: {'n_neighbors': 34},
           mean: 0.97032, std: 0.00633, params: {'n_neighbors': 35},
           mean: 0.96857, std: 0.00678, params: {'n_neighbors': 36},
           mean: 0.97007, std: 0.00629, params: {'n_neighbors': 37},
           mean: 0.96857, std: 0.00486, params: {'n_neighbors': 38},
           mean: 0.96957, std: 0.00565, params: {'n_neighbors': 39},
           mean: 0.96732, std: 0.00423, params: {'n_neighbors': 40},
           mean: 0.96957, std: 0.00618, params: {'n_neighbors': 41},
           mean: 0.96807, std: 0.00587, params: {'n_neighbors': 42},
           mean: 0.96932, std: 0.00650, params: {'n_neighbors': 43},
           mean: 0.96608, std: 0.00651, params: {'n_neighbors': 44},
           mean: 0.96732, std: 0.00716, params: {'n_neighbors': 45},
           mean: 0.96483, std: 0.00624, params: {'n_neighbors': 46},
           mean: 0.96682, std: 0.00650, params: {'n_neighbors': 47},
           mean: 0.96233, std: 0.00683, params: {'n_neighbors': 48},
```

Figure 5.6 Using Grid to find optimal value of K.

CHAPTER 6

IMPLEMENTATION DETAILS

METHODOLOGY

- Collecting data.
- Exploring and preparing the data.
- Training a model on the data.
- Evaluating model performance.
- Improving model performance.

KNN Algorithm

In pattern recognition, the k-nearest neighbors' algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification.

Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data.

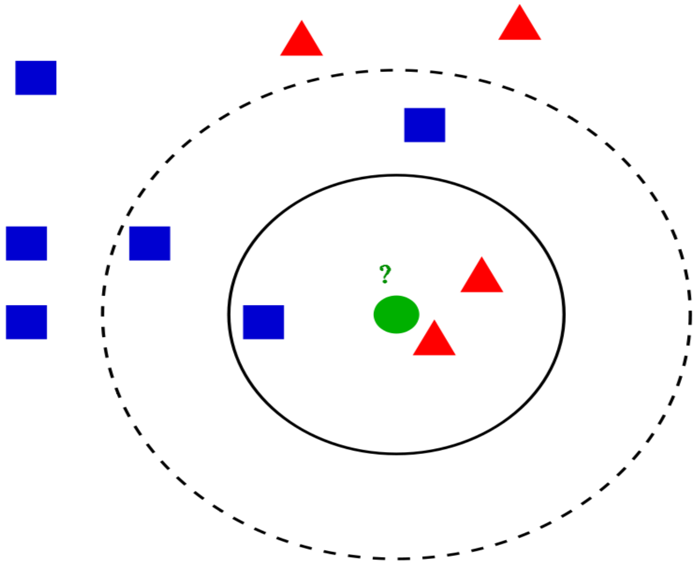


Fig 6.1 KNN Classifier

Collecting Data: We collected data from Kaggle where spam classifier data was already presented for a data science competition.

Exploring and preparing the data: We then used bag of words technique to convert words into numerals and then divided into two parts Training Set and Test Set.

Training a model on the data: We used KNN Classifier to train our model using training set**.**

Evaluating model performance: We used Test data to evaluate performance of the model that is 91.44 percent.

Improving model performance: We used grid search for optimal value of K to improve our model performance.

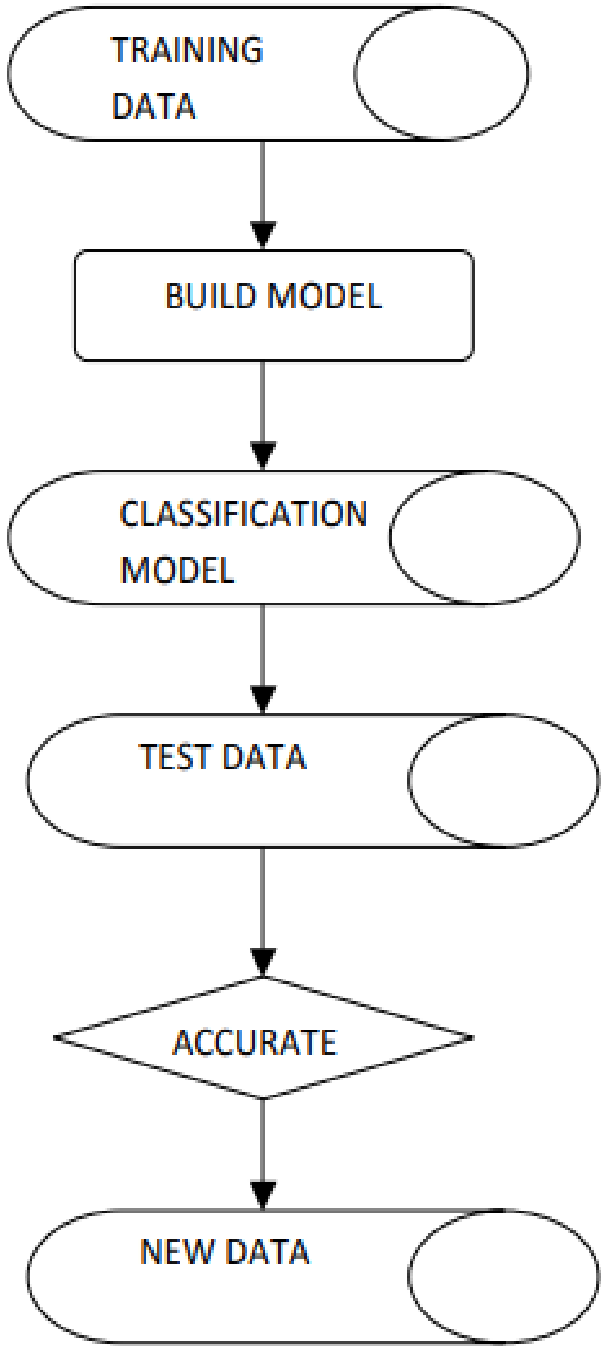


Figure 6.2 Flow Diagram

CHAPTER 7

OVERALL DESCRIPTION

7.1 Project Perspective:

7.1.1 Developers View:

As per developers' points of view we design a system which can predict whether the e-mail is spam or ham and can be customized easily according to data set entered. It is very easy to implement and is fast and accurate.

7.1.2 Users View:

As per user point of view, this system is easy to find whether an e-mail is ham or spam meanwhile enhancing the accuracy.

7.2 Project Functions:

7.2.1 Classifying e-mail into ham or spam:

Our project basic functionality is to predict the smart city given pre- trained model it can achieve accuracy up to 91.44%.

7.3 DFD Diagram:

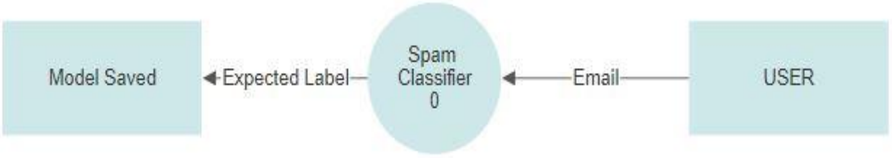


Figure 7.1 Level 0

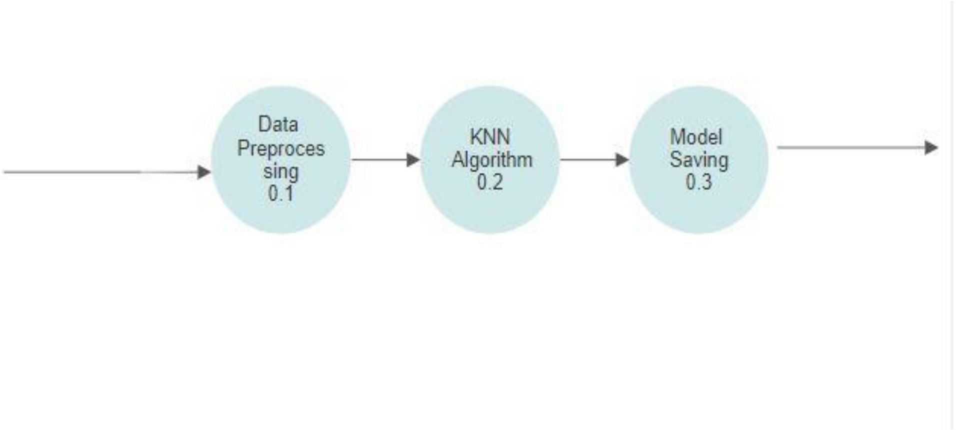


Figure 7.2 Level 1

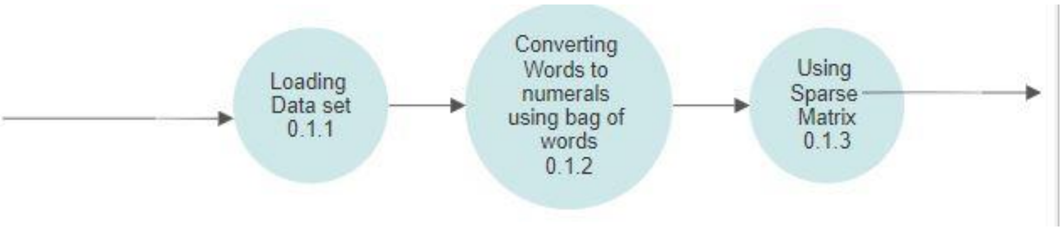

Loading Data set 0.1.1
Converting Words to numerals using bag of words 0.1.2
Using Sparse Matrix 0.1.3

Figure 7.3 Level 2

CHAPTER 8

SYSTEM FEATURES

8.1 System Basic Feature:

- **Platform Independent**

  Our program is platform independent; it can run on any platform having Python libraries.

- **Stable**

  Our code is stable can be run on any Python compiler or can be run online without any difficulties.

- **Adaptable**

  Our program can adapt itself with any dataset loaded or trained and can work according to it.

- **Simple design**

  Our code is simple in design can be understand or use by anyone, who have basic knowledge of Python.

- **Faster**

  Our system is faster as compared to other models with a greater efficiency.

- **Accurate**

  Our program has 91.44% accuracy with great operating power.

CHAPTER 9

OTHER NON-FUNCTIONAL REQUIREMENTS

9.1 Performance Requirements:

Our system should have enough memory and speed to process the data and operations between them and able to store data sets and should be able to process dataset.

9.2 Software Quality Attributes:

9.2.1 Interoperability:

My program works in any operating system having Anaconda environments.

9.2.2 Maintainability:

My model can be change according to different data sets and can be trained accordingly as per required.

9.2.3 Usability:

My project main purpose is to classify e-mail into Ham or Spam.

9.2.4 Efficiency:

My algorithm gives 91%, it is good and process it faster.

9.2.5 Classifications:

My model can predict smart cities effectively and accurately consuming less time.

9.3 TESTING REQUIREMENTS:

9.3.1 DEBUGGING:

- **Using debuggers:**

We used in built debuggers in Python console to identify major errors like syntax and semantics errors. After all the syntax and semantics errors were removed, we setup check points in our program which is divided into following segments**.**

- **Data Preprocessing:**

In this segment I checked whether our data is processed correctly or not, by running each command.

- **Model Defining:**

In model defining, I defined KNN Algorithm and checked whether it is compiled correctly or not by using confusion matrix, which provides accuracy of my model

- **Training Model:**

In this segment my model was trained using different values of k so that it can achieve good accuracy.

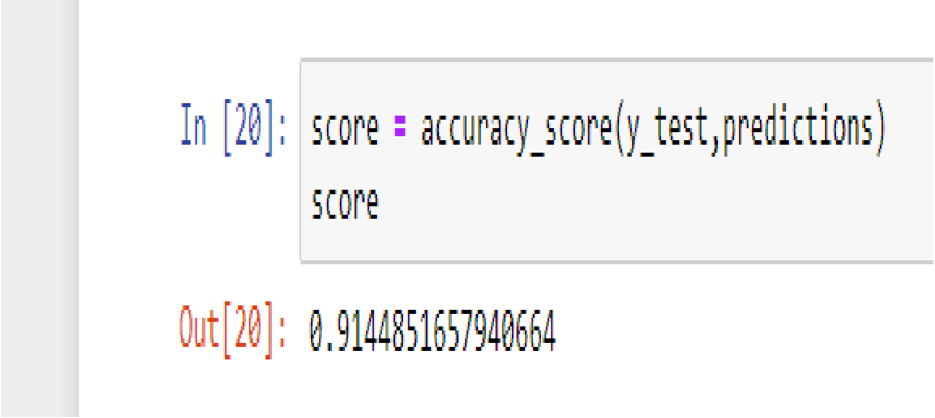
- **Backtracking:**

We start from the point where problem occurred and go back through the code in sequence to rectify where error has been occurred and then to add corrective measures in the segment, so that the part can run properly and the proceed the same process further, until the whole program is checked and corrected.

9.3.2 Performance Testing:

- The following figure shows accuracy of our model

.



```
In [20]: score = accuracy_score(y_test,predictions)
         score

Out[20]: 0.9144851657940664
```

Fig 9.1 Accuracy of Model

CHAPTER 10

CONCLUSION

Given a set of words, we used Bag of words to obtain words which allow us to distinguish between spam and ham emails. We used k-NN method that gives highest classification accuracy no matter how many attributes are used and which method is used. We also see that on average the accuracy improves as the number of attributes increase. It is possible that accuracy may increase more than 91.44 % if we further increase the number of attributes. By using other techniques like tf-idf we can get accuracy up to 97% but it uses too much space. Other than that, we implemented n-gram, word2vec but these features take more time to train model but can give significant increase in accuracy on cost of space. Techniques like word2vec can take up to space 32 GB so we used Bag of words to convert words into vectors.

CHAPTER 11

SCOPE FOR FUTURE WORK

- **Using GUI:** I want to add GUI so that it is easier to use and read results for non-machine learning background users.
- **No. of datasets:** I used only 500 datasets. It will increase processing time since data is large.
- **Processing Time:** processing time can be reduced by trying different algorithms.
- **EMBEDDING IT AS API:** We can use it as an API with our messaging app, mailing app like its currently used in g-mail, true caller.

BIBLIOGRAPHY

- About Company: http://www.brillicaservices.com/about-us/
- Dataset: https://www.kaggle.com/uciml/sms-spam-collection-dataset
- Testing: http://softwaretestingfundamentals.com/system-testing/