CALIFORNIA STATE UNIVERSITY
FULLERTON

# Department of Computer Science

This project has been satisfactorily demonstrated and is of suitable form.

This project report is acceptable in partial completion of the requirements for the Master of Science degree in Computer Science.

Machine Learning to detect Fake News

**Project Title**

NAVKIRAN KAUR

**Student Name**

DR. SHAWN WANG

**Advisor's Name**

**Advisor's signature** **Date**

DR. KENNETH KUNG

**Reviewer's name**

**Reviewer's signature** **Date**

# MACHINE LEARNING

# TO DETECT FAKE NEWS

## PROJECT REPORT

SUBMITTED BY:

NAVKIRAN KAUR

CWID: 802974436

EMAIL ID: kaur.navkiran92@csu.fullerton.edu

SUBMITTED TO:

ADVISOR: Dr. Shawn Wang

REVIEWER: Dr. Kenneth Kung

CPSC 597

DEPARTMENT OF COMPUTER SCIENCE

CALIFORNIA STATE UNIVERSITY, FULLERTON

DATE: May 11th, 2017

# ABSTRACT

The purpose of this project is to work on the fake news dataset and to detect the fake news. This project gives an overview of some of the most popular machine learning algorithms- Naive Bayes, SVM, K-NN and Logistic Regression. The news is categorized as fake, real, satire, bias or hate based on two different datasets. The term ‘Fake News’ is used here to define the unlike problems from fake news to satire news or the hate news. Online fake news has been a topic of interest these days and has been used in multitude of ways. I will be using machine learning algorithms in Python with Scikit-Learn to train and test the data to find out the accuracies of each algorithm. Also, the description of the algorithms is presented and comparison of their performance to find out which algorithm is suitable for this kind of text dataset.

*Keywords: Machine Learning, Naïve Bayes, SVM, K-NN, Logistic Regression, Python, Scikit-Learn*

# TABLE OF CONTENTS

# TABLE OF FIGURES

# TABLE OF Tables

# 1. INTRODUCTION

**1.1 Definition of Problem:** Fake news is kind of yellow journalism that comprises of thoughtful half-truth spread through the broadcasting news media or via internet-based social media. Fake news is becoming a treacherous movement that's fast becoming a global problem. Many opponent's troupes charges guilt on technology firms like Facebook, Twitter and Google signifying that they have an obligation to address the fake news sweeping because their algorithms affects who sees which stories. But systematizing fake news detection with machine learning might be a clever idea as machine learning is already widely used and thrived in detecting spam. Fake news, news manipulation and the lack of trust in the media are growing problems with huge consequences in our society. This term "fake news" is basically targeting and weaponizing for political purposes, Also, the American democracy has been constantly pounded by changes in media technology. During the presidential candidate election, the fake news was both broadly and deeply slanted in favor of Donald Trump. According to this research Paper about Fake News, it says that there were around 115 pro-Trump fake stories that were shared on Facebook a total of 30 million times and 41 pro-Clinton fake stories shared a total of 7.6 million times. So, by coming across these stories and data, I find the need to check whether the news is genuine or fake without coming to any conclusion.

**1.2 Objective:** The objective of this project is to examine the problems and possible significances related with the spread of fake news. We will be working on different fake news data set in which we will apply different machine learning algorithms to train the data and test it to find which news is the real news or which one is the fake news. As the fake news is a problem that is heavily affecting society and our perception of not only the media but also facts and opinions themselves. By using the artificial intelligence and the machine learning, the problem can be solved as we will be able to mine the patterns from the data to maximize well defined objectives. So, our focus is to find which machine learning algorithm is best suitable for what kind of text dataset. Also, which dataset is better for finding the accuracies as the accuracies directly depends on the type of data and the amount of data. The more the data, more are your chances of getting correct accuracy as you can test and train more data to find out your results.

## 1.3 Assumptions and Limitations:

### Assumptions

- The data set we are using is effective and efficient.
- The data collected from different online sites is reliable for this project.
- The machine learning algorithm’s selected for this project are suitable for this kind of dataset.
- As the machine learning algorithms are more effective on the numerical datasets, it will effectively work on the text based data set as well.
- The data we are training is being properly trained for future predictions.

**Limitations**

- As the data is collected online, most of the data is non-readable
- As the features keep updating or deprecating, it sometimes created trouble in writing the commands.
- We will be able to train and test the data but because being it text data, it is sometimes hard to convert the string data to float data to make calculations.

## 1.4 Development Environment

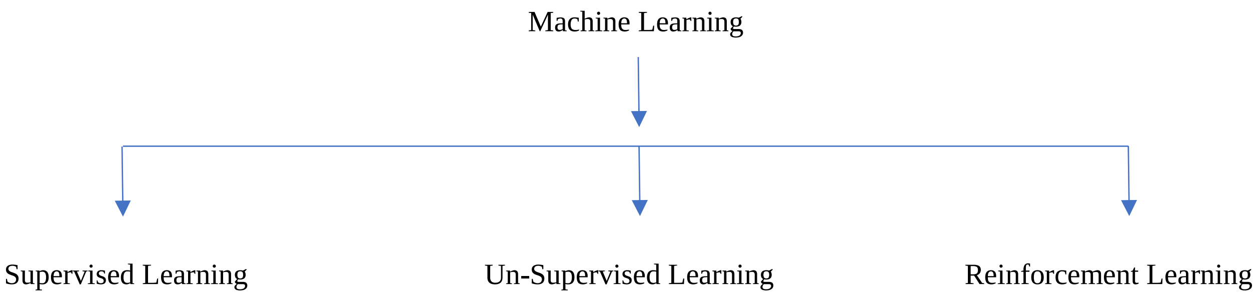
### Hardware Requirement

- Windows 7 and advance.
- Processor Intel Pentium P-4 1ghz
- At least 4-GB RAM preferred
- Hard disk of size 160GB or above
- Monitor mono VGA/ Color SVGA(Proffered)
- Keyboard & Mouse

### Software Requirement

- Python
- Anaconda
- Jupyter- IPython Notebook
- Mat Plot Library
- Natural Language Processing

**2. MACHINE LEARNING:** We are possibly living in the most crucial period of human history where managing the big data and mining out the valuable information is the focus. Machine Learning is a core sub area and type of artificial intelligence that offers the computer the capability to learn and change when they are exposed to newly generated data. Learning here means understanding, observing and representing information about statistical phenomenon. Nowadays, machine learning algorithms are used in different domains for different applications. For some time, it has been used in decision making systems such as spam filters, virus detection or for detecting network intrusion systems. They can be used in such environment because they can be trained to detect abnormal behaviors.



**SUPERVISED LEARNING**: The data we input is called the training data and it is called the label or result. In this we train a model by using the training process which makes the different predictions. The trained model is corrected if it gives the wrong predictions. The training process lasts till the model attains the desired level of accuracy on the training data. In the supervised learning we have some idea about how the results will look like.

For Example: Logistic Regression

**UNSUPERVISED LEARNING**: In the unsupervised learning, the model is developed by inferring structures existing in the input data. It can be through a mathematical process to systematically reduce redundancy, or it may be organizing data by similarity.

For Example: Apriori Algorithm

**REINFORCEMENT LEARNING**: By using this algorithm, the can be trained to make specific decisions. The machine is exposed to an environment where it trains itself continually using trial and error. It learns from the experience to make accurate business decisions.

For Example: Markov Decision Process.

Machine learning system can be trained to classify news by Training and Testing (Classification).

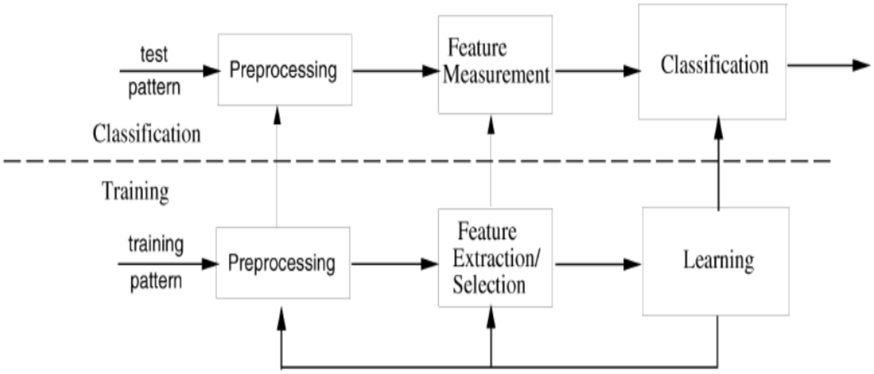


Figure 1: Machine Learning classified as training and testing

## 3. ARCHITECTURE:

The architecture of this project includes working on the raw data and cleaning the null and nan values to clean the data that includes data pre-processing. The next step includes converting the data into vectors and building the models to work on to. Then the next step includes applying machine learning algorithm on the models to predict the accuracy of the applied algorithms.

## Detailed Architecture:

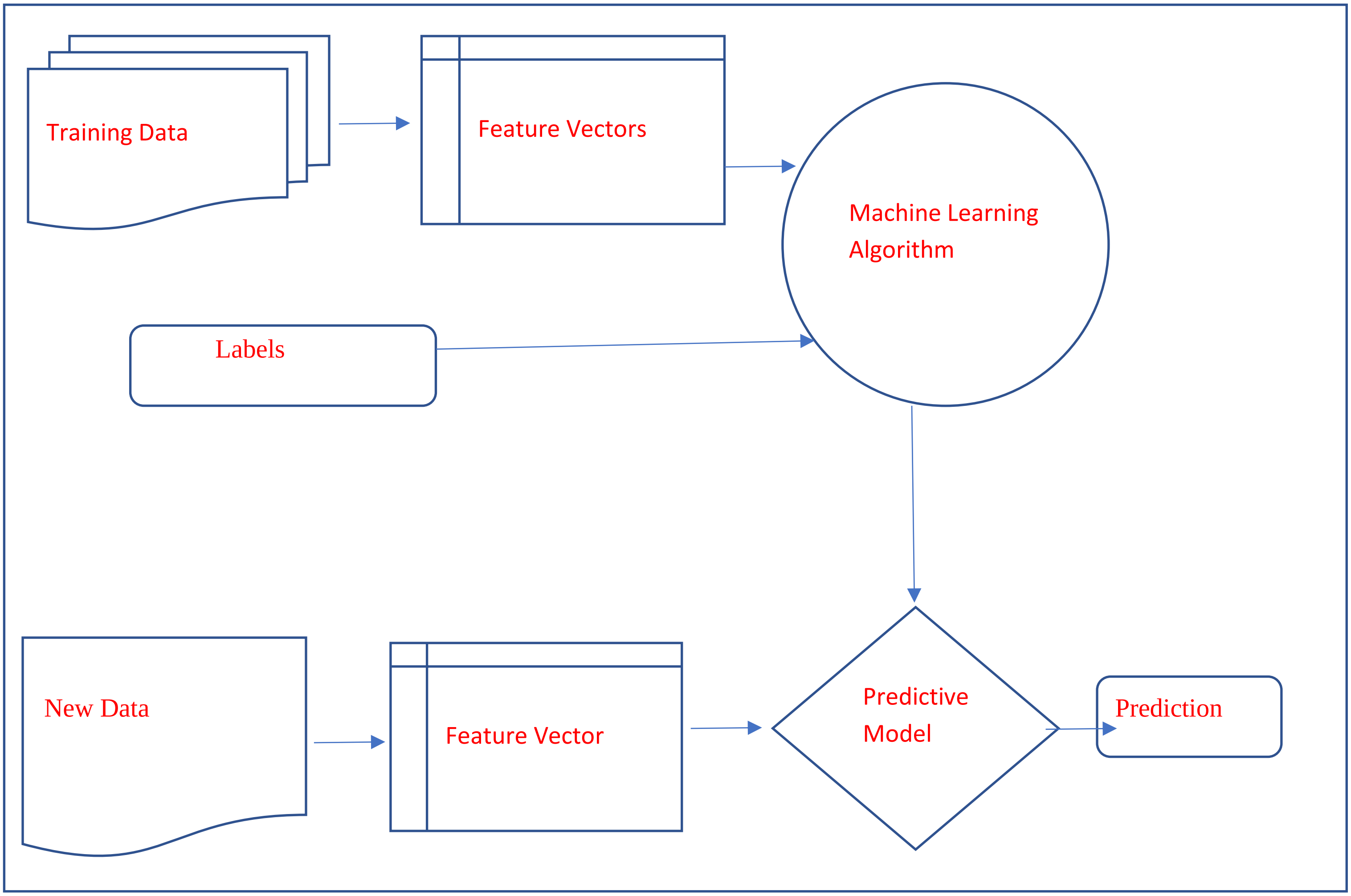


Figure 2: Architecture of this project

The architecture of this project is divided into two phases:

The Build Phase

The Operational Phase.

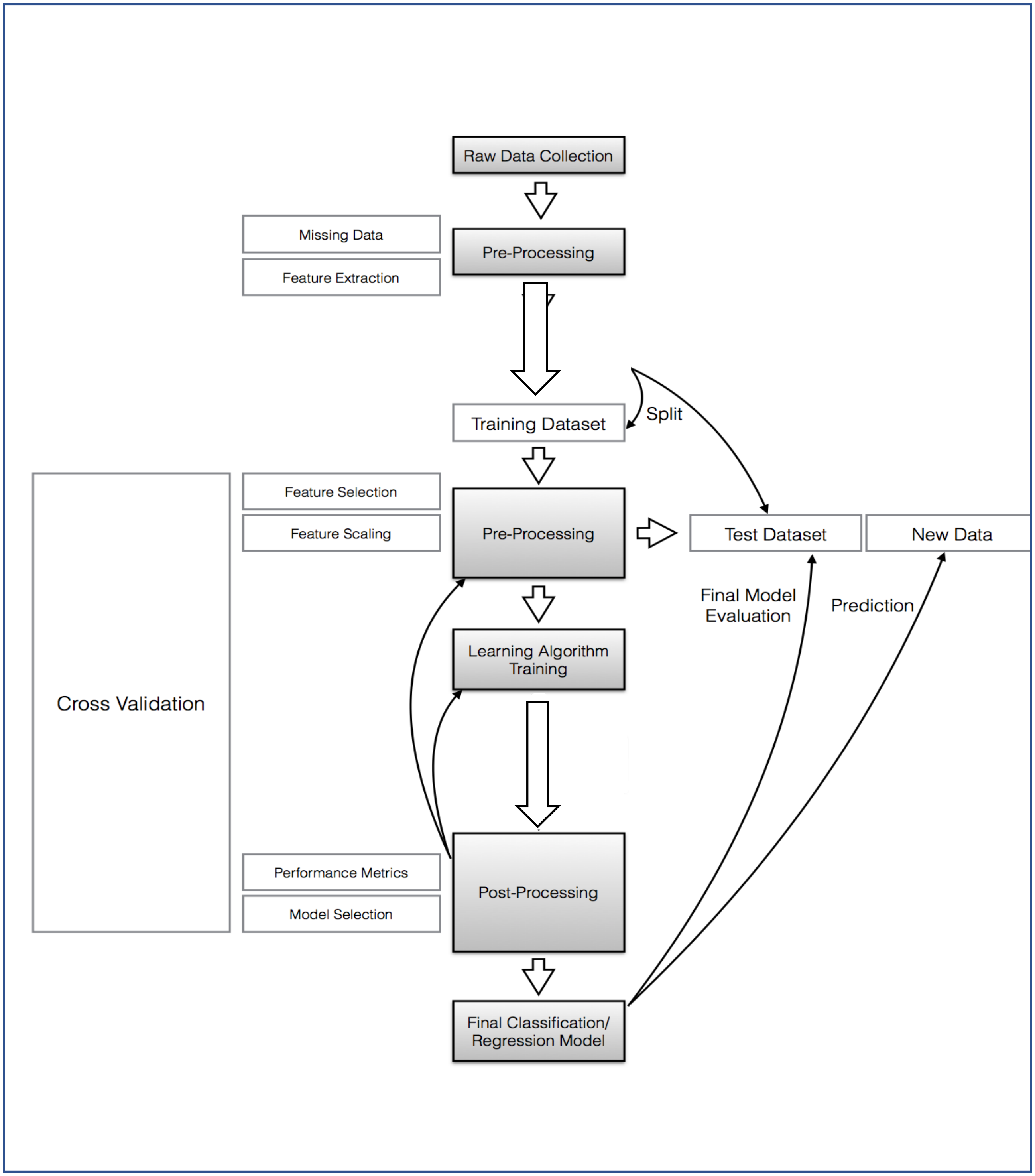
In this build phase, we basically train the data by labeling them and extract the feature vectors and perform machine learning algorithms (also known as estimation algorithm) on them. In the operational phase, we take new data and get the feature vectors from them and then apply the predictive model from the machine learning algorithms that have been used to find out the prediction of the new data entered the machine.

In this project, we are focused on supervised learning. It is the task of inferring a function from labeled training data. The training data consists of a set of training examples. A supervised learning algorithm analyzes the training data and produces an inferred function which can be used for mapping new examples.

**3.1Implementation**: In a superior picture, this is how the architecture is basically implemented. It has various steps that we need to follow step wise to finally achieve the results and predictions of the algorithms such as the estimation algorithm and predictive models. Predictive model is a concept of building a model that can make predictions. It is divided into two sub areas: regression and pattern classification. We will be focusing on the classification model, the general approach of assigning predefined class labels to instances to group them into discrete categories. It basically follows these steps:

- Raw data collection and feature extraction
- Sampling
- Cross validation
- Normalization
- Machine Learning Algorithm
- Prediction

Figure 3: Implementation



# 4. METHODS

**4.1Data Set:** A data set is a collection of data. It consists of all the information gathered during a survey which needs to be analyzed. Learning how to interpret the results is a key component to the survey process

**4.1.1 Data Set Description:** In this project, there are two kinds of news based data sets: one which is a big data set which includes the author, title, language, country, site URL, domain rank, spam score and type of the news such as fake, bias, conspiracy, hate or satire. It has around 17000 news from different online sites. This dataset was available on Kaggle's. This data set contains text and metadata from more than 100 websites using the webhose.io API.

Text and metadata from fake news sites:

- author: author of story
- title: title of story
- language: data from webhose.io
- site_url: site URL
- country: data from webhose.io
- domain_rank: data from webhose.io
- spam_score: data from webhose.io
- type

Fake: 7429, Conspiracy: 430, Satire: 267, Bias: 239 and hate: 199

The other data set is a simple data set of 7000 news which is categorized as news and labels: i.e. fake or real. This data set was also available on Kaggle's website.

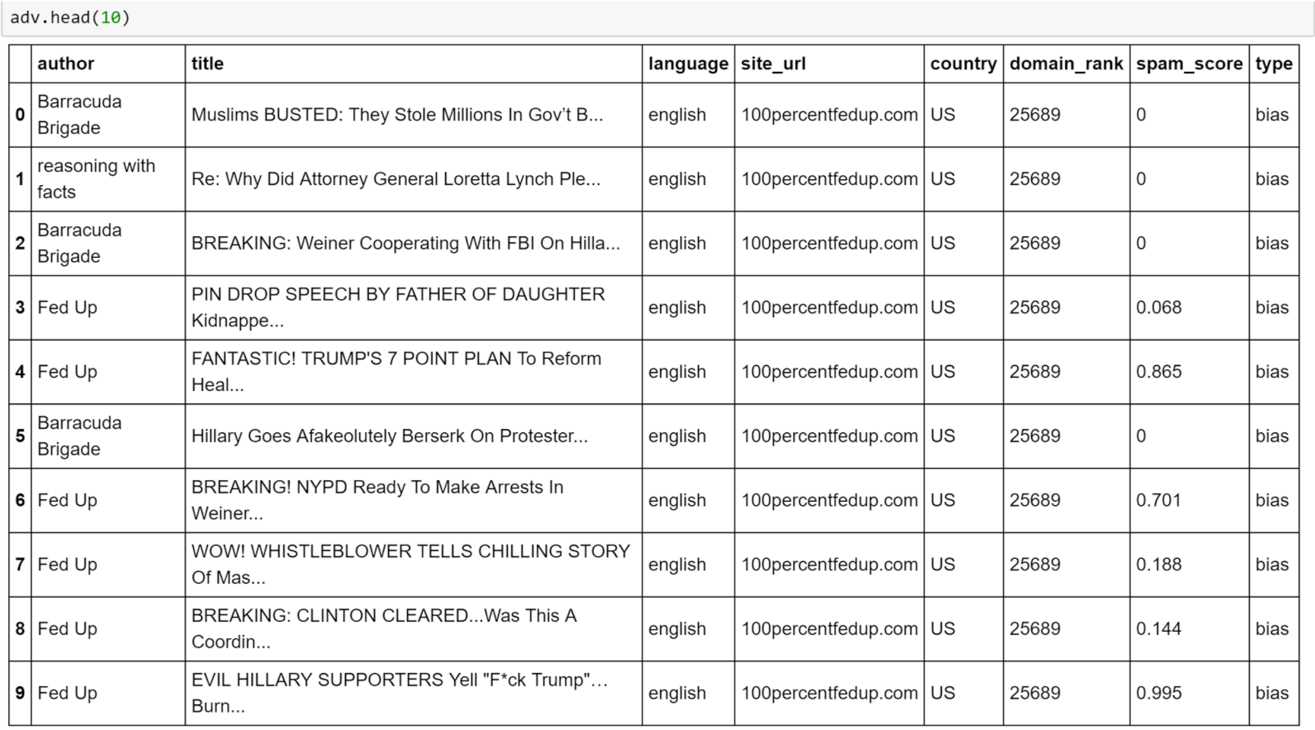
The type of data set is .csv file.

Text and metadata from fake news sites:

- news: text of story
- label: type of story

Fake: 1206 and Real: 1147

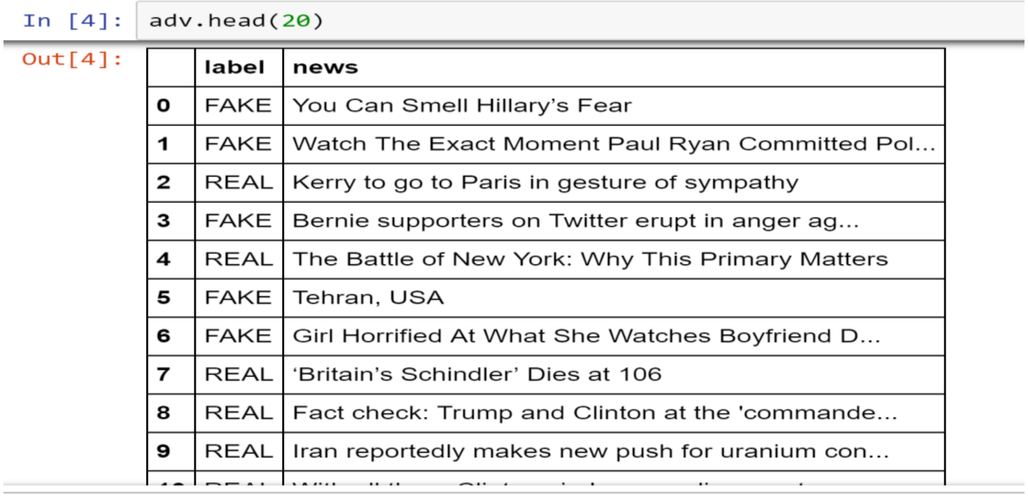
**DataSet1:** Figure4: Screen shot of dataset 1



```
adv.head(10)
```

| | author | title | language | site_url | country | domain_rank | spam_score | type |
|---|---|---|---|---|---|---|---|---|
| 0 | Barracuda Brigade | Muslims BUSTED: They Stole Millions In Gov't B... | english | 100percentfedup.com | US | 25689 | 0 | bias |
| 1 | reasoning with facts | Re: Why Did Attorney General Loretta Lynch Ple... | english | 100percentfedup.com | US | 25689 | 0 | bias |
| 2 | Barracuda Brigade | BREAKING: Weiner Cooperating With FBI On Hilla... | english | 100percentfedup.com | US | 25689 | 0 | bias |
| 3 | Fed Up | PIN DROP SPEECH BY FATHER OF DAUGHTER Kidnappe... | english | 100percentfedup.com | US | 25689 | 0.068 | bias |
| 4 | Fed Up | FANTASTIC! TRUMP'S 7 POINT PLAN To Reform Heal... | english | 100percentfedup.com | US | 25689 | 0.865 | bias |
| 5 | Barracuda Brigade | Hillary Goes Afakeolutely Berserk On Protester... | english | 100percentfedup.com | US | 25689 | 0 | bias |
| 6 | Fed Up | BREAKING! NYPD Ready To Make Arrests In Weiner... | english | 100percentfedup.com | US | 25689 | 0.701 | bias |
| 7 | Fed Up | WOW! WHISTLEBLOWER TELLS CHILLING STORY Of Mas... | english | 100percentfedup.com | US | 25689 | 0.188 | bias |
| 8 | Fed Up | BREAKING: CLINTON CLEARED...Was This A Coordin... | english | 100percentfedup.com | US | 25689 | 0.144 | bias |
| 9 | Fed Up | EVIL HILLARY SUPPORTERS Yell "F*ck Trump"... Burn... | english | 100percentfedup.com | US | 25689 | 0.995 | bias |

**DataSet2:** Figure5: Screen shot of dataset 2



```
In [4]: adv.head(20)
Out[4]:
```

| | label | news |
|---|---|---|
| 0 | FAKE | You Can Smell Hillary's Fear |
| 1 | FAKE | Watch The Exact Moment Paul Ryan Committed Pol... |
| 2 | REAL | Kerry to go to Paris in gesture of sympathy |
| 3 | FAKE | Bernie supporters on Twitter erupt in anger ag... |
| 4 | REAL | The Battle of New York: Why This Primary Matters |
| 5 | FAKE | Tehran, USA |
| 6 | FAKE | Girl Horrified At What She Watches Boyfriend D... |
| 7 | REAL | 'Britain's Schindler' Dies at 106 |
| 8 | REAL | Fact check: Trump and Clinton at the 'commande... |
| 9 | REAL | Iran reportedly makes new push for uranium con... |

### 4.1.2 Data Use:

So, in this project we are using different packages and to load and read the data set we are using pandas. By using pandas, we can read the .csv file and then we can display the shape of the dataset with that we can also display the dataset in the correct form. We will be training and testing the data, when we use supervised learning it means we are labeling the data. By getting the testing and training data and labels we can perform different machine learning algorithms but before performing the predictions and accuracies, the data is need to be preprocessing i.e. the null values which are not readable are required to be removed from the data set and the data is required to be converted into vectors by normalizing and tokening the data so that it could be understood by the machine. Next step is by using this data, getting the visual reports, which we will get by using the Mat Plot Library of Python and Sickit Learn. This library helps us in getting the results in the form of histograms, pie charts or bar charts.

## 4.2 Preprocessing:

The data set used is split into a training set and a testing set containing in Dataset 1 -3256 training data and 814 testing data and in Dataset II- 1882 training data and 471 testing data respectively. Cleaning the data is always the first step. In this, those words are removed from the dataset. That helps in mining the useful information. Whenever we collect data online, it sometimes contains the undesirable characters like stop words, digits etc. which creates hindrance while spam detection. It helps in removing the texts which are language independent entities and integrate the logic which can improve the accuracy of the identification task.

### 4.2.1 Natural Language Processing:

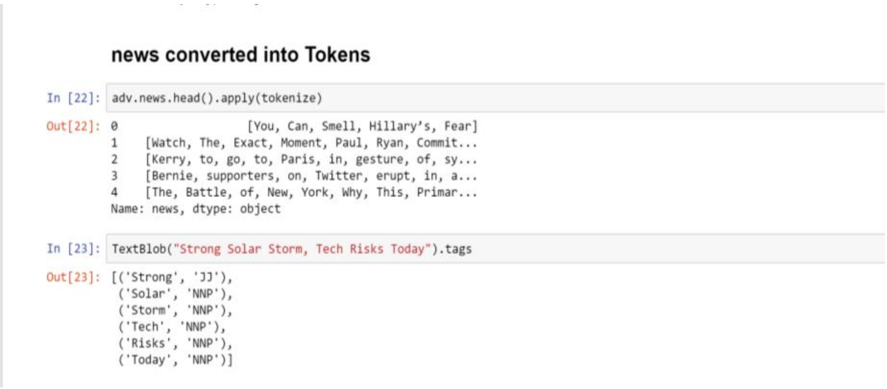
Natural language toolkit basically known as NLTK. NLTK is a collection of libraries and programs for symbolic and statistical natural language processing. It is mainly used for tokenization, POS Tagging or parsing. It is focused on gathering and classifying unstructured texts.

**4.2.1.1** Removal of Stop words: Stop words like and, the, of are very common in all the English sentences and are not very meaningful in deciding the Legitimate status, so these words are removed.

**4.2.1.2** Removal of Nan Values: Also, while working on the datasets we need to remove the nan and null values that are not useful, this makes the dataset more effective in working as there are no un-desired values.

**4.2.1.3** Tokenization: It is the process of braking a stream of text up into words or the other meaningful elements called tokens. Also, we normalize the dataset, it involves replacing normal features, so that eac og them would range from 0 to 1.

Figure6: Screenshot of tokenization



news converted into Tokens

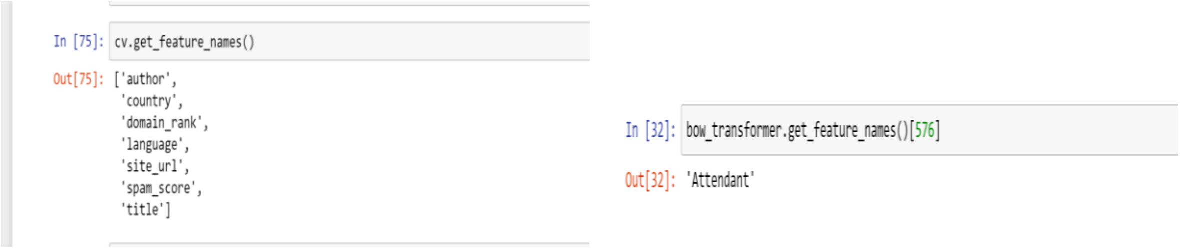
```
In [22]: adv.news.head().apply(tokenize)

Out[22]: 0                  [You, Can, Smell, Hillary’s, Fear]
         1    [Watch, The, Exact, Moment, Paul, Ryan, Commit...
         2    [Kerry, to, go, to, Paris, in, gesture, of, sy...
         3    [Bernie, supporters, on, Twitter, erupt, in, a...
         4    [The, Battle, of, New, York, Why, This, Primar...
         Name: news, dtype: object

In [23]: TextBlob("Strong Solar Storm, Tech Risks Today").tags

Out[23]: [('Strong', 'JJ'),
          ('Solar', 'NNP'),
          ('Storm', 'NNP'),
          ('Tech', 'NNP'),
          ('Risks', 'NNP'),
          ('Today', 'NNP')]
```

**4.2.1.4** Lemmatization: It is the process of alliancing together the different inflected forms of a word so they can be analyzed as a single item. For example, “include”, “includes” and “included” would all be represented as “include”. The context of the sentence is preserved in lemmatization.

## 4.2.2 Feature Extraction: Feature extraction s the process of selecting a subset of relevant features for use in model construction. Feature extraction methods helps in to create an accurate predictive model. They help in selecting features that will give better accuracy. When the input data to an algorithm is too large to be handled and its supposed to be redundant then the input data will be transformed into a reduced illustration set of features also named feature vectors. Altering the input data to perform the desired task using this reduced representation

instead of the full-size input. Feature extraction is performed on raw data prior to applying any machine learning algorithm, on the transformed data in feature space.

Figure7: Screenshot of feature Extraction

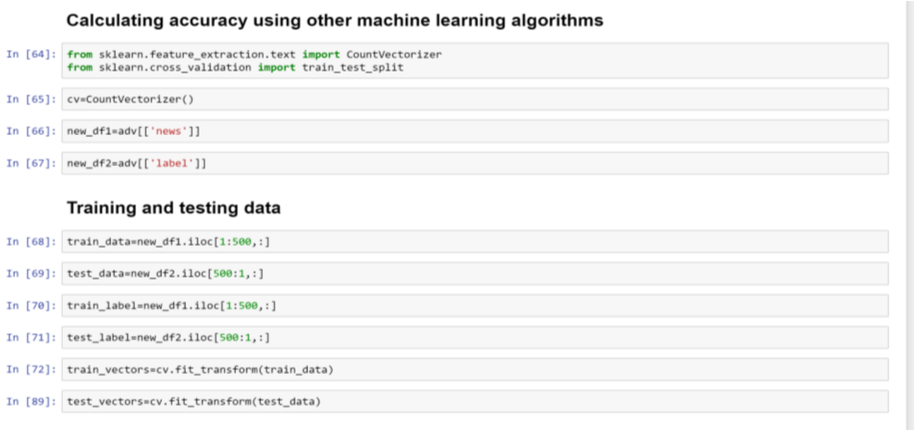


```
In [75]: cv.get_feature_names()

Out[75]: ['author',
          'country',
          'domain_rank',
          'language',
          'site_url',
          'spam_score',
          'title']
```

```
In [32]: bow_transformer.get_feature_names()[576]

Out[32]: 'Attendant'
```

### 4.2.3 Training the Classifier:

As In this project I am using Scikit-Learn Machine leanring ibrary for implementing the architecture.Scikit Learn is an open source python Machine Learning library which comes bundled in 3$^{rd}$ distribution anaconda. This just needs importing the packages and you can compile the command as soon as you write it. If the command doesn't run, we can get the error at the same time. I am using 4 different algorithms and I have trained these 4 models i.e. Naïve Bayes, Support Vector Machine, K Nearest Neighbors and Logistic Regression wic are very popular methods for document classification problem. Once the classifiers are traned, we can c heck the performance of the models on test-set. We can extract the word count vector for each mail in test-set and predict it class with the trained models.



**Calculating accuracy using other machine learning algorithms**

```
In [64]: from sklearn.feature_extraction.text import CountVectorizer
         from sklearn.cross_validation import train_test_split

In [65]: cv=CountVectorizer()

In [66]: new_df1=adv[['news']]

In [67]: new_df2=adv[['label']]
```

**Training and testing data**

```
In [68]: train_data=new_df1.iloc[1:500,:]

In [69]: test_data=new_df2.iloc[500:1,:]

In [70]: train_label=new_df1.iloc[1:500,:]

In [71]: test_label=new_df2.iloc[500:1,:]

In [72]: train_vectors=cv.fit_transform(train_data)

In [89]: test_vectors=cv.fit_transform(test_data)
```

Figure8: Screenshot of training and testing data

## 4.3 Classifier:

### 4.3.1 Naïve Bayes:

It is a very simple classification algorithm that makes robust assumptions about the individuality of each input variable. It is mostly applicable and effective in many problem domains. It is an instinctive method that uses the possibilities of each attribute fitting to each class to make a prediction. It simplifies the calculation of probabilities by if the probability of each attribute belonging to a given class value is independent of all other attributes. According to the scikit learn organization naïve Bayes classifiers have worked quite well in many real-world situations and spam filtering. Naïve Bayes classifiers can be extremely fast compared to more sophisticated methods. In this project, we will be using Multinomial Nave Bayes which implements the naïve Bayes algorithm for multinomially distributed data and is one of the two classic naïve Bayes variants used n text classification.

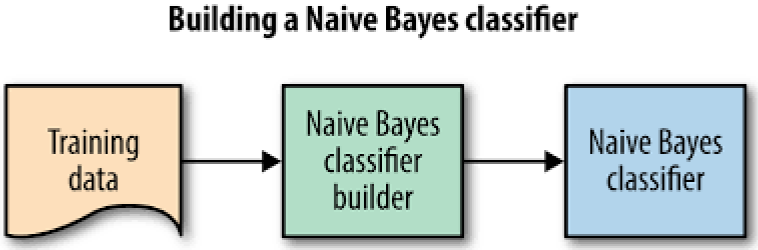

Building a Naive Bayes classifier
Training data
Naive Bayes classifier builder
Naive Bayes classifier

Figure9: Naïve Bayes Classifier

### 4.3.2 Logistic Regression:

It is a classification algorithm used to estimate the discrete values based on given set of independent variables. It predicts the probability of occurrence of an event by fitting data to a logit function. It predicts the probability; its output values lie between 0 and 1. It helps in predictive analysis. It is used to describe data. When selecting the model for the logistic regression, another important consideration is the model fit adding independent variables to a logistic regression model will always increase tis statistical validity, because it will always explain a bit more variance of the log odds.

### 4.3.3 Support Vector Machine:

It is also a classification method. In machine learning, support vector machine algorithm comes under the supervised learning models that analyses the data used for classification and regression analyses. SVM is primarily a classifier method that performs classification task by constructing.

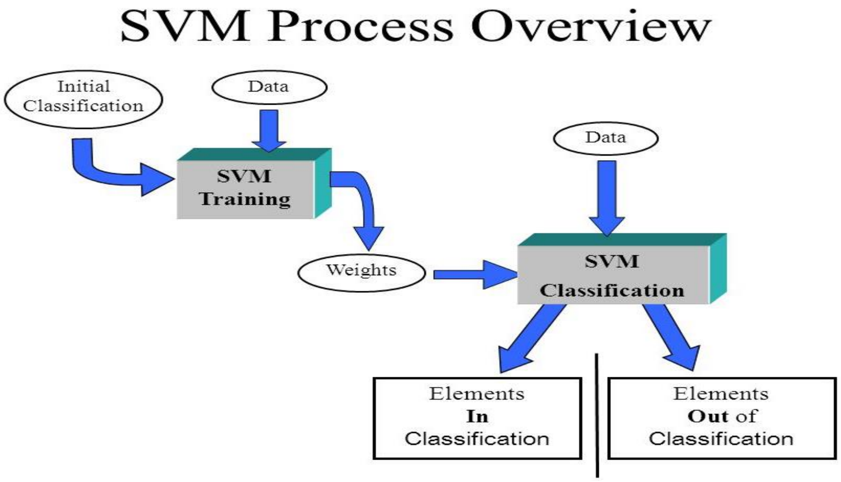
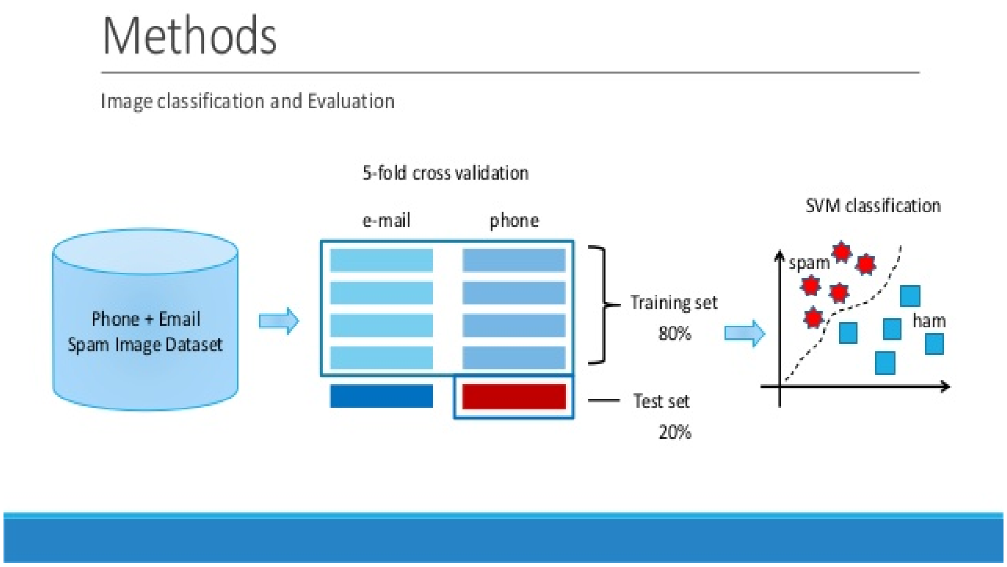


Figure 10: SVM Process View

It tries to find an optimum separating hyperplane between members of the two initial classifications. When the training examples consist of very diverse expression patters, then finding an optimal hyperplane can be impossible. The expression data can be transformed to a higher dimensional space by applying a kernel function. This transformation can have the effect of allowing a separating hyperplane to be found. Two classes are produced as SVM results one is the positive class: that contains elements with expression patterns like those in the positive examples in the training set. And other is the negative class that contains all other members of the input.



Methods

Image classification and Evaluation

Figure 11: Example of SVM classification

**4.3.4 K Nearest Neighbors:** KNN 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. In k-NN classification, the output is a class membership. An object is classified by a majority 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. 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. The k-NN algorithm is among the simplest of all machine learning algorithms. Both for classification and regression, it can be useful to assign weight 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.

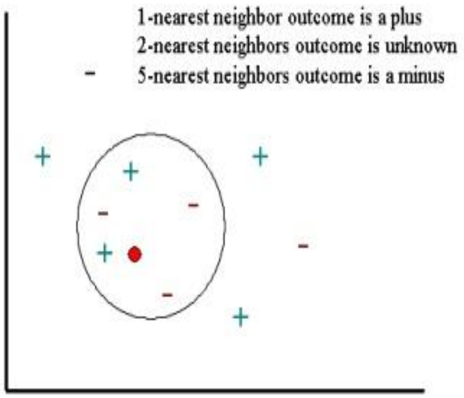
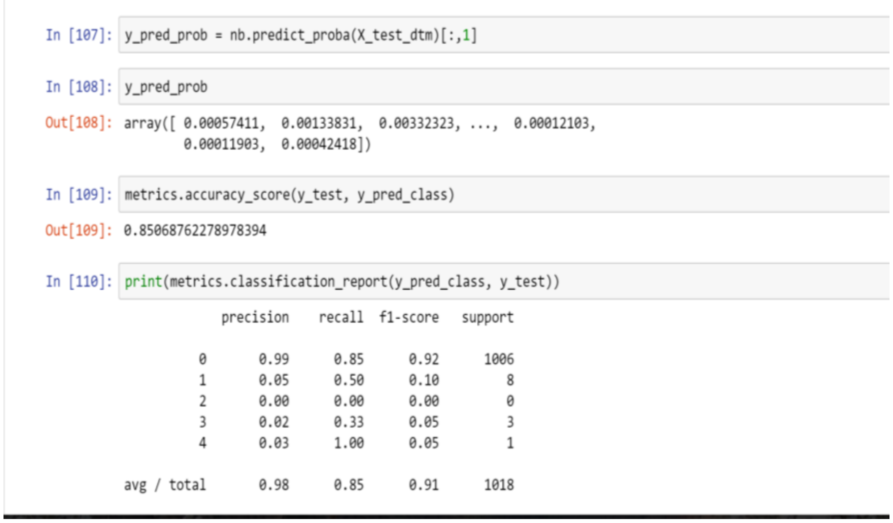

1-nearest neighbor outcome is a plus
2-nearest neighbors outcome is unknown
5-nearest neighbors outcome is a minus
-
+
+
+
-
-
+
-
-
+
-

Figure 12: K-NN algorithm

**4.4Prediction:** A suitable means for the evaluation of the performance is the confusion matrix which is a square matrix that consists of columns and rows that list the number of instances as actual class vs predicted class ratios. Often the prediction, accuracy or error are used to report classification performance. It is defined as the faction of correct classification out of the total number of samples, it is often used synonymous to precision although it is calculated differently.



```
In [107]: y_pred_prob = nb.predict_proba(X_test_dtm)[:,1]

In [108]: y_pred_prob

Out[108]: array([ 0.00057411,  0.00133831,  0.00332323, ...,  0.00012103,
                  0.00011903,  0.00042418])

In [109]: metrics.accuracy_score(y_test, y_pred_class)

Out[109]: 0.85068762278978394

In [110]: print(metrics.classification_report(y_pred_class, y_test))
                       precision    recall  f1-score   support

                    0       0.99      0.85      0.92      1006
                    1       0.05      0.50      0.10         8
                    2       0.00      0.00      0.00         0
                    3       0.02      0.33      0.05         3
                    4       0.03      1.00      0.05         1

          avg / total       0.98      0.85      0.91      1018
```

Figure 13: Screenshot of Prediction

# 5. RESULTS AND ANALYSIS

So, we cleaned the data set, converted it into tokens and then into vectors so that the machine learning algorithms could be applied on them. The results we observed are based on both the data sets and are different from both the data sets.

1. **Length Observation:** Starting from the length of the news we got some histogram plots. It tells us how the length of news varies and we can get some idea about the length of the type of news. This length calculation also helps in calculating the count and mean of the length. In this project, we have used the mat plot library to get the visual results. As visual results help in better understanding of the results.

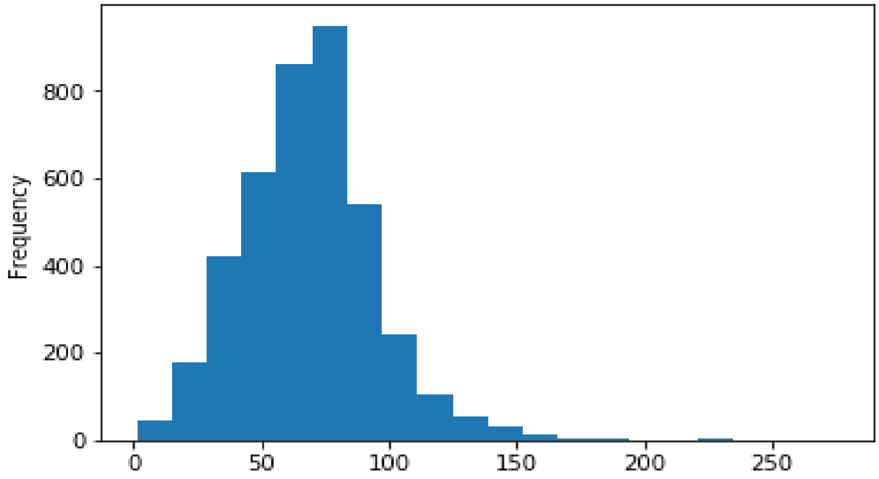


Figure 14: Data Set 1: Length of news

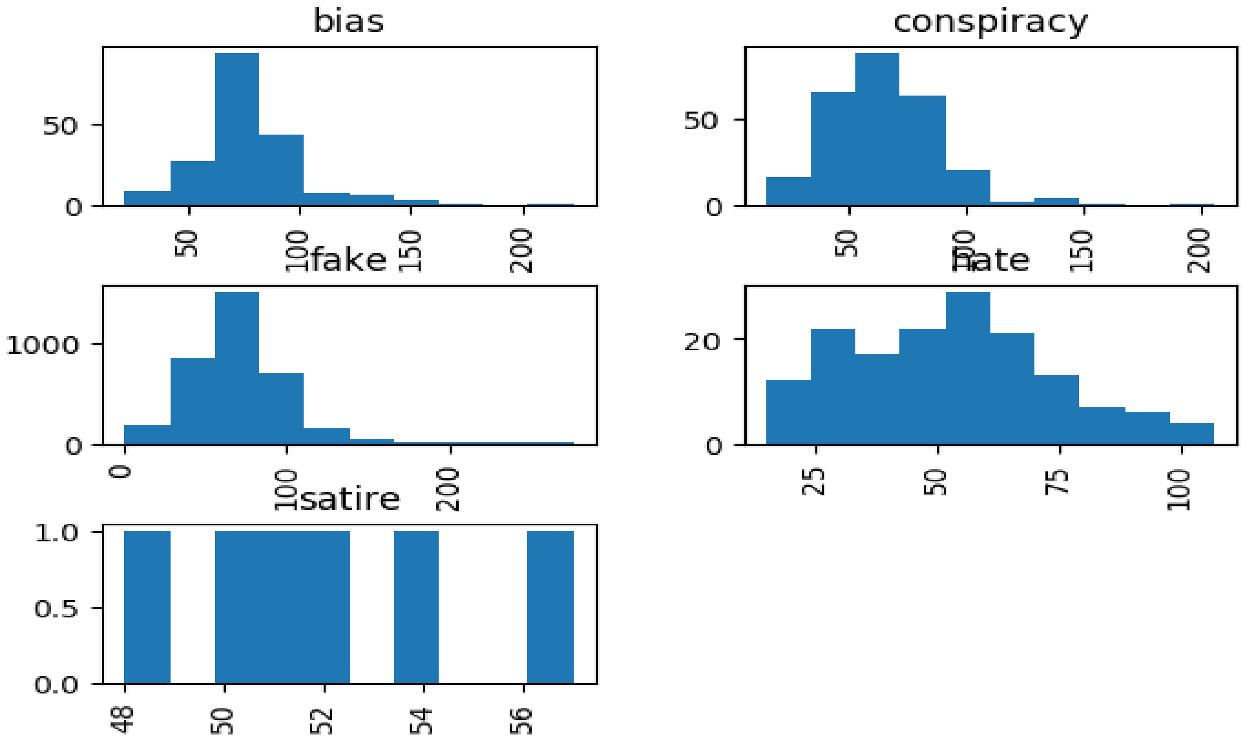


Figure 15: Data Set 1: Length of news according to their type

As we can observe from here that the maximum frequency of the length of news expanded to more than 800 words and the density of the messages stays 0 and 150. In the second graphs, we can get idea how the length of news varies for each kind of news as we have fake news, conspiracy news, fake news, hate news and satire news.

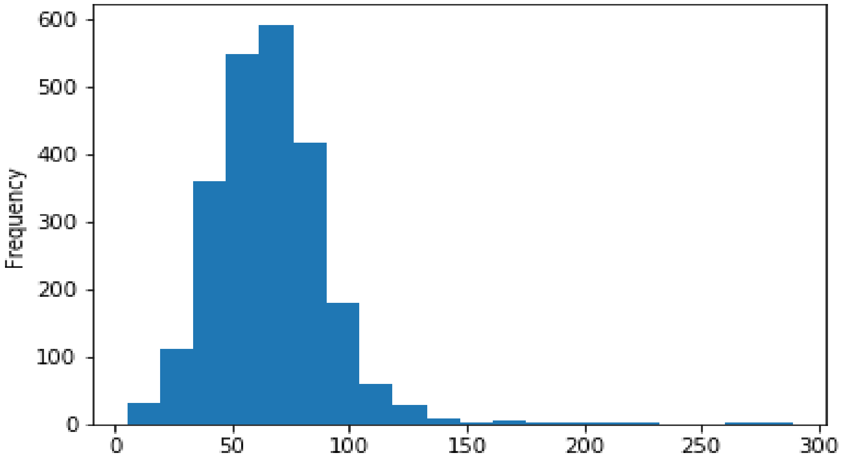


Figure 16: Data Set 2: Length of news

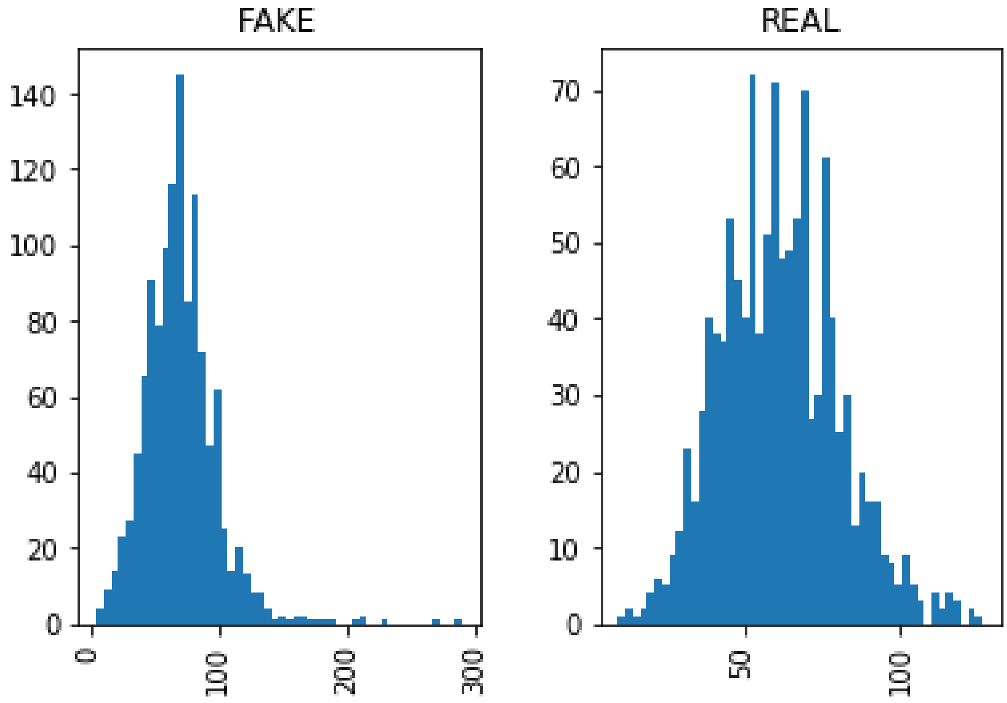


Figure 17: Data Set 2: Length of news according to their type- fake or real

In the second dataset, we can observe that the maximum frequency goes to 600 and the density of the news remains same as the dataset 1. But when we observe the length of news for specific type such as fake or real, we can observe that the length for real news varies just to 70 but the fake news varies to 140. So, we from these graphs we can get some idea about the how the length varies which helps in prediction.

2. **Sparse Matrix:** Sparse matrix or sparse is a matrix in which most of the elements are zero. In contrary if most of the elements are non-zero then the matrix is considered dense. We have calculated the sparsity for both the datasets:

| **Data Set** | **Sparsity** | **Sparsity matrix shape** | **Number of non-zeros** |
|---|---|---|---|
| **Data set 1** | **0.09%** | **4070,12080** | **43816** |
| **Data set 2** | **0.13%** | **2353,7934** | **24537** |

From this we can observe that the data set 2 has more sparsity and the data set 1 is more dense data set as it has more number of non – zero elements.

3. **Confusion Matrix:** Confusion matrix is table that is used to define the performance/prediction of a classification model or classifier on a set of test data for which the true values are known. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class. Classification accuracy alone can be misleading if there is an unequal number of observations in each class or if there are more than two classes in the dataset. Calculating confusion matrix gives a better idea of what your classification model is getting rift and types of error it is making.

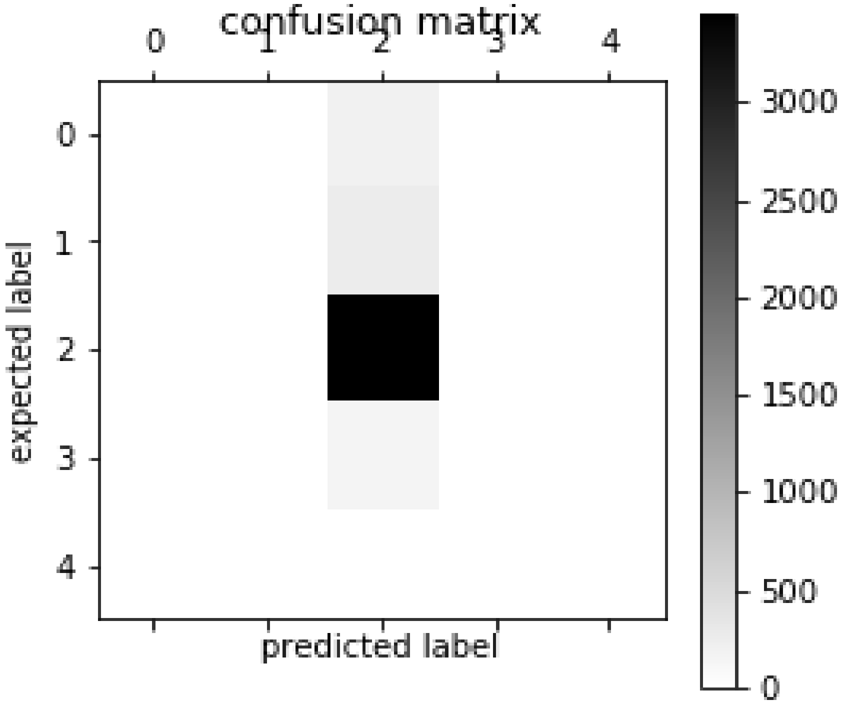


Figure 18: Confusion matrix of Dataset 1

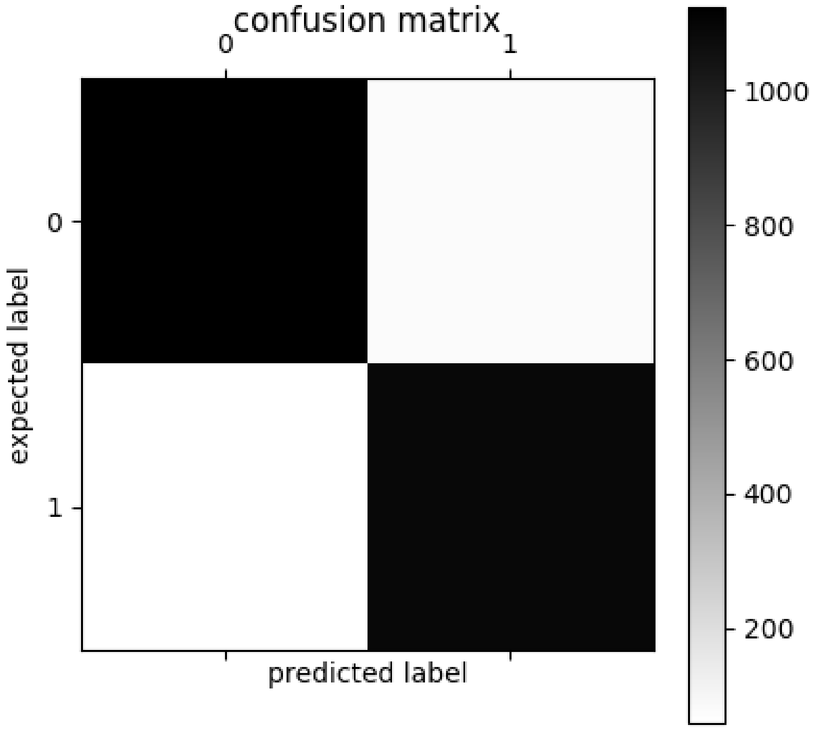


Figure 19: Confusion matrix of Dataset 2

4. **Cross Validation**: Cross validation is one of the most useful techniques to evaluate different combinations of feature selection and learning algorithms. There are multiple types of cross-validation and the most common one would probably be k-fold cross-validation, in which the original training dataset s split into k different subsets. In this we can calculate the cross-validation score and mean score. A major step when working with machine learning is checking the performance of your model. One method of assessing a machine learning algorithm’s performance is cross-validation. This technique has the algorithm make predictions using data not used during the training stage. Cross-validation partitions a dataset and uses a subset to train the algorithm and the remaining data for testing. Because cross-validation does not use all the data to build a model, it is a commonly used method to prevent overfitting during training.

From here we can see that, the data set 1 has stable values. All the values and algorithms depends on the type of dataset you have.

| scores | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Score mean | Score std |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset1 | 84.14 | 84.45 | 84.45 | 84.14 | 84.70 | 84.87 | 84.49 | 85.75 | 85.58 | 85.75 | 84.95 | 0.64 |
| Dataset2 | 61.90 | 65.60 | 66.13 | 64.36 | 65.43 | 57.44 | 64.36 | 63.82 | 65.42 | 62.03 | 63.65 | 2.49 |

Table 1: Cross Validation Score

## 5. Accuracy vs. Training set size

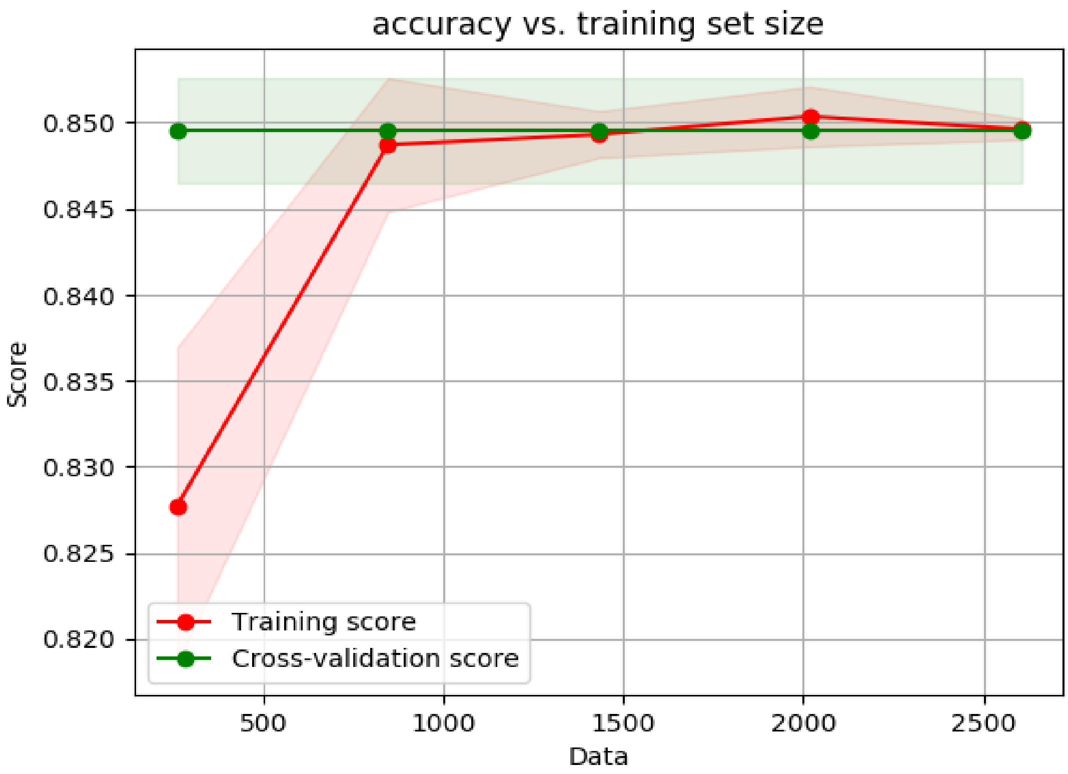


Figure 20: Accuracy vs Training Set Size of Dataset 1

As we can observe the accuracy vs training set size graph is based on the scores calculated by training the data and the scores calculated by cross validation score. In the data set 1 we can see that the training score rises with the number of attempts made on the data set whereas the cross-validation score remains constant which tells us that the way we are training and testing our data is stable and we are going on the right direction.

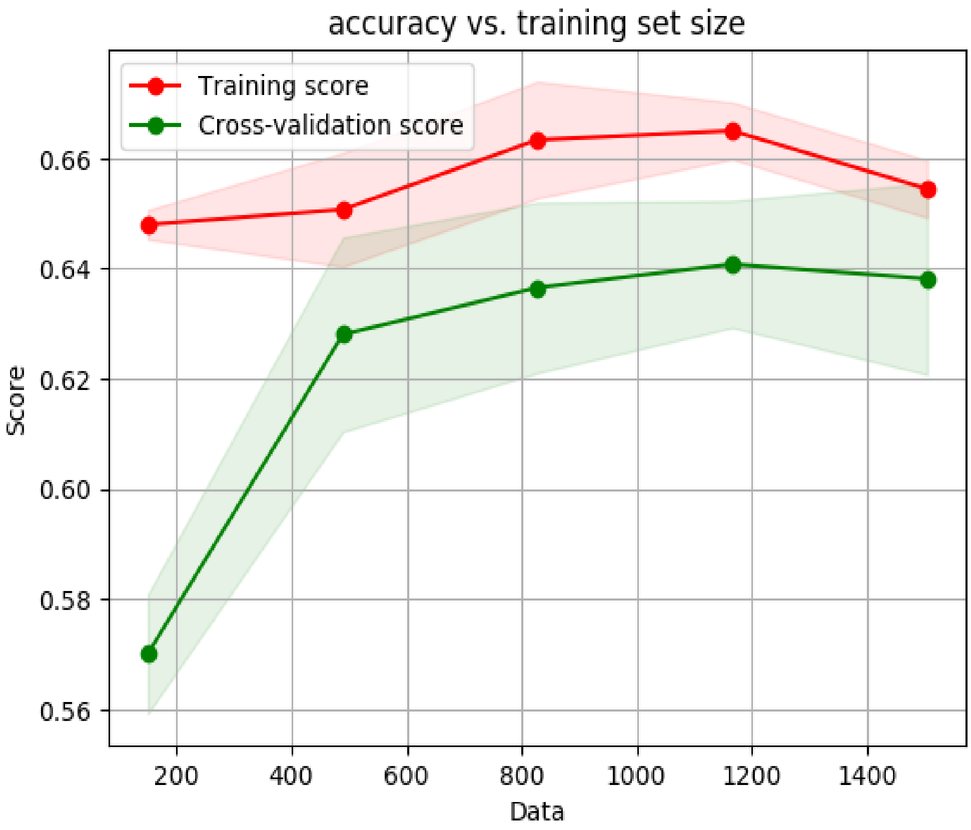


Figure 21: Accuracy vs Training Set Size of Dataset 2

In the data set 2 we can observe that there is huge variation in the validation scores. These scores depend on the data set. That is why machine learning is always applied on to the big amount of data as the when you train more amount of data you are likely to get more correct values for the test data.

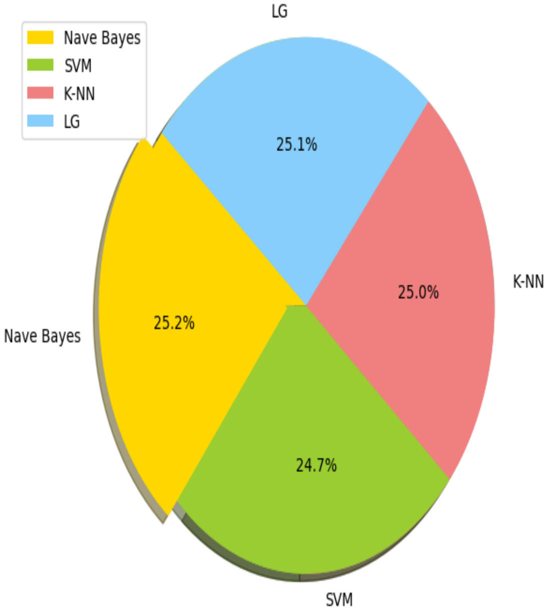
6. **Accuracies of different machine learning algorithms applied on these two data sets:**

| Data Set | Naive Bayes<br>On training and testing data | SVM<br>On training and testing data | KNN<br>On training and testing data | Logistic Regression<br>On training and testing data |
|---|---|---|---|---|
| Dataset 1 | 85.06 | 85.026 | 85.00 | 85.36 |
| Dataset 2 | 78.94 | 63 | 52.29 | 78.77 |

Table 2: Accuracies

After applying the algorithms and getting the prediction/ accuracy scores we can make analysis on which algorithm will be best suitable for the which dataset.

More visual results of these algorithms are represented in the form of pie charts which helps in better understanding and are more user interactive.



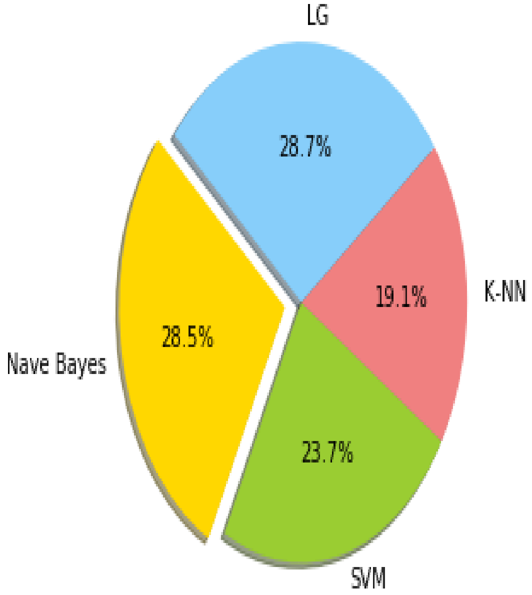


Figure 22: Accuracy Comparison Dataset 1

Figure 23: Accuracy Comparison Dataset 2

As from Dataset 1 we can analyze that, the most suitable algorithm for this kind of text data set is the SVM algorithm and then is the Naïve Bayes. The accuracies for naïve Bayes algorithms are slightly similar as there is very slight difference in their accuracies.

And from data set 2 we can analyze that logistic gives the better accuracy among other algorithms and then it’s Naïve Bayes and then SVM.

So, from here question arises isn’t the algorithms have same format but why these algorithms have different accuracies. The answer is, the accuracy and type of algorithm depends on your dataset. If the data set is clean and you have more amount of information then there are chances of getting better accuracies.

**Spam Detector Time by each algorithm for computation on these two datasets -**

| Data Set | Naive Bayes | SVM | KNN | Logistic Regression |
|---|---|---|---|---|
| Dataset 1 | 4 ms | 6.88 s | 5.5 ms | 65.2 ms |
| Dataset 2 | 8.65 ms | 6.13 s | 3.01 ms | 43 ms |

Table 3: Spam Detector Time 33

From here we can say KNN takes least time in spam detection whereas logistic regression takes most time.

SVM and Naïve Bayes takes almost same time for the computation. Also, the spam detection time depends on the type of your data.

When we look for algorithm, there are number of dimensions we need keep in consideration.

- Type of dataset- text data or numerical data
- Training examples, what kind of data are you training and testing for making calculations.
- What are you trying to accomplish?
- How important is it to visualize the process?

For example, in my project I calculated applied the Naïve Bayes Classifier in two ways on data set 2.

Firstly, I applied it on the data training and then calculated it without taking in consideration the testing data and the accuracy I got for the data set was 94% which seems accurate, but when we use some data, we have some idea that what type of data you have.

But when I applied the same algorithm on both the training data and testing data, and when I calculated the prediction results, the results varied between 60’s and from that score I could tell the accuracy is more real now. Because the data set I used was not very fine data and I had just 5000 rows of information about the news which is very less for making predictions in the real world. This is how your algorithms depends on the data set you use.

## Also, talking about the algorithm, each algorithm has its features that are taken into consideration.

**Logistic Regression:** Logistic regression is a well-behaved classification algorithm that can be trained if you expect your features to be roughly linear and the problem to be linearly separable. It is also robust to noise and you can avoid overfitting and even do feature selection easily. It can also be used in Big data scenarios since it is efficient.

It was best suitable for second kind of dataset because it has nice probabilistic interpretation unlike naive Bayes or SVM’s and you can easily update your model to take in new data. Also, if you expect t receive more training data in the future that you want to be able to quickly incorporate into your model then it is the best algorithm.

**Naïve Bayes:** If the training set is small, high bias/low variance classifier e.g. Naïve Bayes have an advantage over low bias/high variance classifiers like KNN or Logistic regression, since the latter will over lift. It computes the multiplication of independent distributions, suffer multicollinearity. It works best on the unbalanced classes and nominal attributes only. So, these are sometimes counted as disadvantages for this kind of algorithm. The other advantages of this algorithm are these algorithms are very simple and quickie to evaluate and their performance is high.

**SVM's:** Support vector machine learning algorithm are considerate moderate for all kinds of data set. As they have high accuracy and nice theoretical guarantees regarding overfitting and with an appropriate kernel they can work well even if the data isn't linearly separable in the base feature space. It is more efficient in using when you have two classes. It classifies the data by finding th best hyperplane that separates all the data points from once class from those of the other class.

It is extremely accurate and tends not to over fit. Over fitting means that the model is so closely aligned to training data sets that it does not know how to respond to new situations.

Once trained, it is a fast option because its jut deciding between one of the two classes.

It tends to handle complex, non-linear classification very well.

**Comparison of algorithms:**

| | |
|---|---|
| **Naïve Bayes** | It should be used when we have a large training set<br>If the instances have several attributes<br>The attributes which describes the instances are conditionally independent.<br>It is easier to predict class of the test data set<br>It is faster than other algorithms |
| **Logistic Regression** | It is one of the most interpretable machine learning algorithm<br>Requires minimal tuning<br>Robust for noisy data set |
| **Support Vector Machine** | It offers best classification performance I.e. the accuracy on training data and results out to be average algorithm for all kinds of dataset.<br>It renders more efficiency for correct classification of the future data.<br>It does not make strong assumptions<br>It does not over fit the data |
| **K Nearest Neighbors** | Categorizing data points on their distance to other points in a training dataset can be simple yet effective way of classifying data.<br>The training time of kNN is short. |

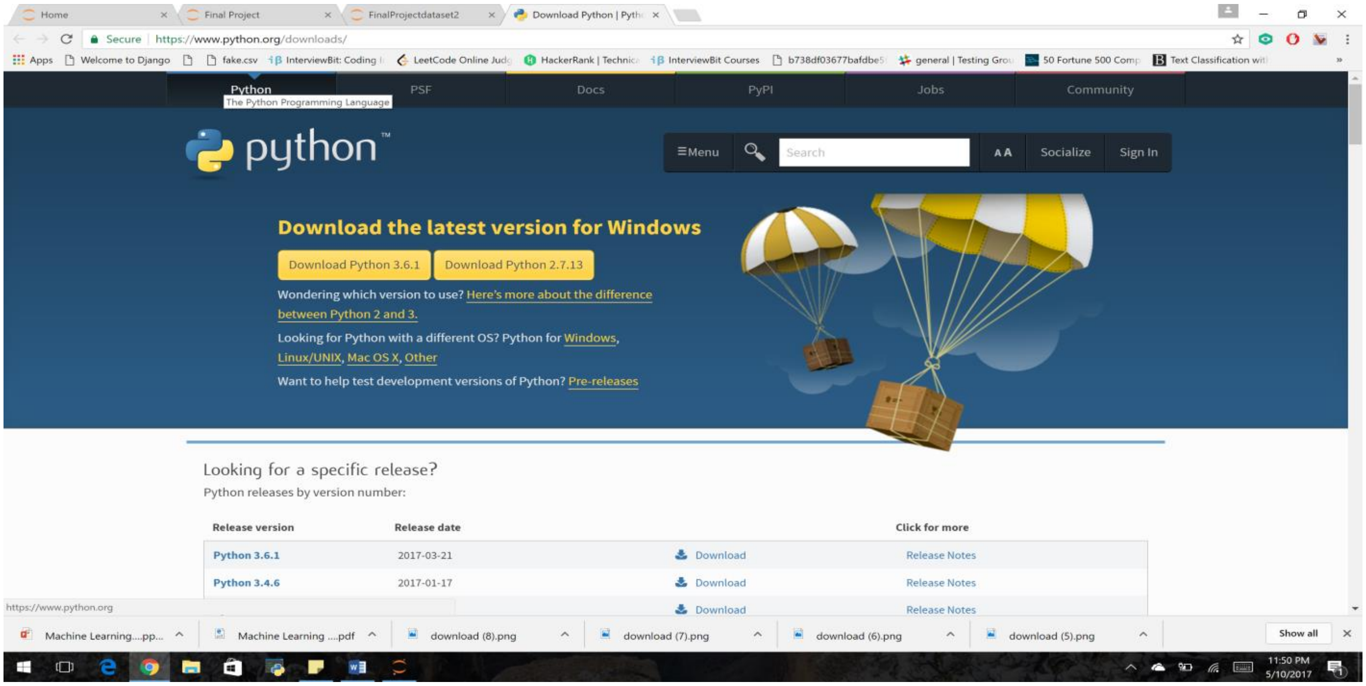
Table 4: Brief Algorithm Comparison 36

**6. CONCLUSION:** In the end after all the research and after calculating the results, I would conclude that the better data often beats the better algorithm. Designing good features goes a long way. If you have huge data set, which ever classification algorithm you use might not matter so much in terms of classification performance as we can observe that from data set 1. We have around 17000X6 matrix which contains huge amount of data, and the accuracies of each algorithm for this data set is almost similar. Also, what I have concluded from this research is that the more you use training data, you can overcome the model complexity.

**7. INSTALLATION INSTRUCTIONS:** We need to Install Python, Anaconda, I Jupyter i.e. the I-Python Notebook and Scikit Learn Library. The installation instructions are mentioned below.

## Python: Version 3+ Version

The Python download requires about 24 Mb of disk space-



Download the latest version for Windows

Wondering which version to use? Here's more about the difference between Python 2 and 3.

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases

Looking for a specific release?

Python releases by version number:

| Release version | Release date | | Click for more |
|---|---|---|---|
| Python 3.6.1 | 2017-03-21 | Download | Release Notes |
| Python 3.4.6 | 2017-01-17 | Download | Release Notes |
| | | Download | Release Notes |

**Downloading**

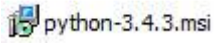
1. Click Python Download.

The following page will appear in your browser.

1. Click the Download Python 3.4.3 button.

The file named python-3.4.3.msi should start downloading into your standard download folder. This file is about 24 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 10 seconds over a cable modem).

The file should appear as



python-3.4.3.msi

2. Move this file to a more permanent location, so that you can install Python (and reinstall it later, if necessary).
3. Start the Installing instructions directly below.

**Installing**

1. Double-click the icon labeling the file python-3.4.3.msi.

An Open File - Security Warning pop-up window will appear.

**Click Run.**

A Python 3.4.3 Setup pop-up window will appear.

1. Ensure that the Install for all users radio button is pressed.
2. Click Next > button.

A new Python 3.4.3 Setup pop-up window will appear (Select Destination Directory).

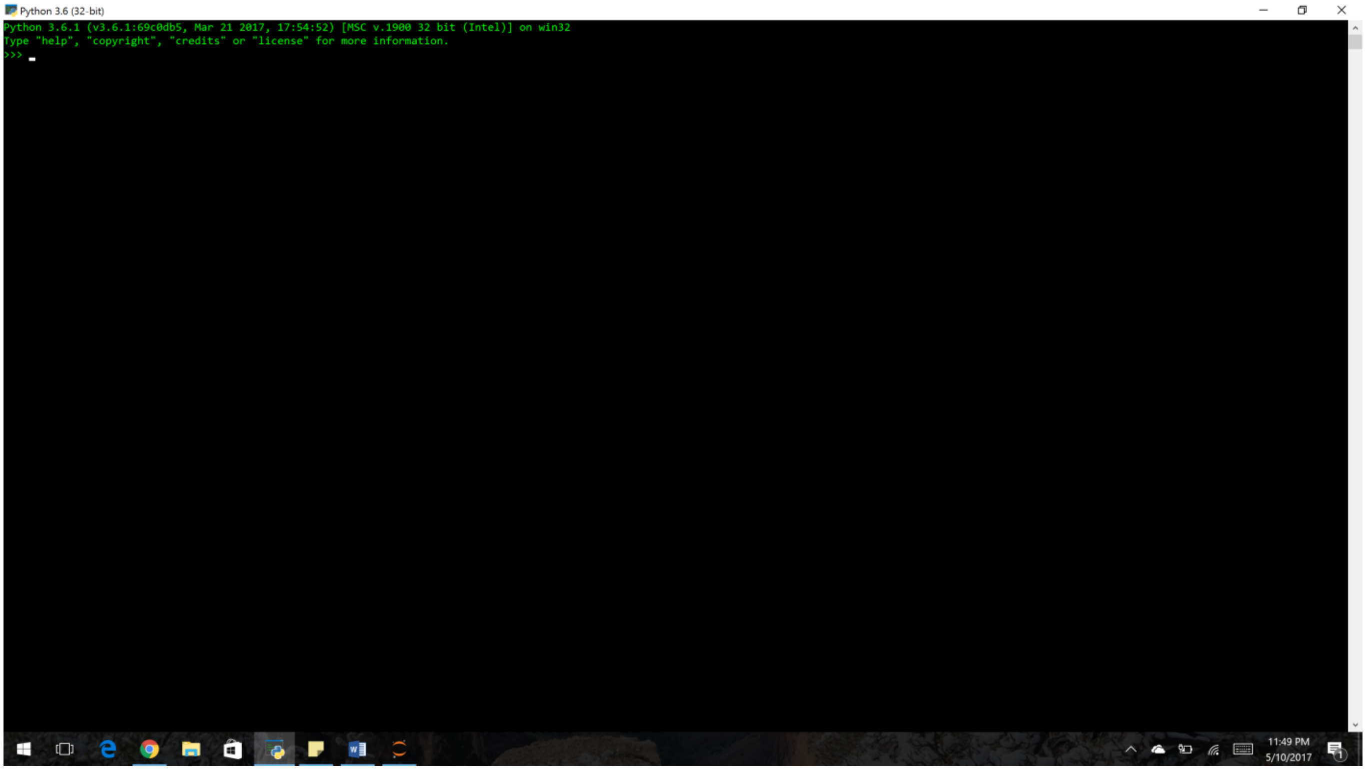
1. Click the Next > button.

A new Python 3.4.3 Setup pop-up window will appear (Customize Python 3.4.3).

A new Python 3.4.3 Setup pop-up window will appear (Install Python 3.4.3).

1. Click the Finish button.

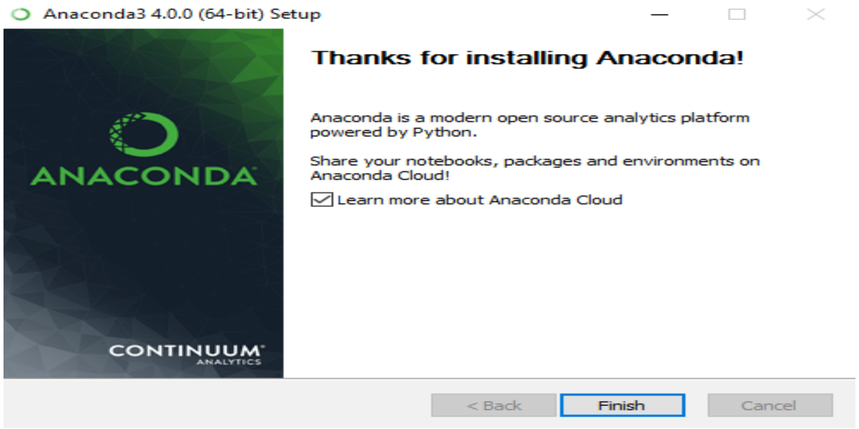
Python should now be installed. To try to verify installation, navigate to the directory C:\Python34 (or to whatever directory on which you installed Python) double-click the icon/file python.exe. The following pop-up window will appear.



```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

## Anaconda Installation

1. Download the Anaconda installer.
2. Click Run to launch the installer.
3. Click Next.
4. Read the licensing terms and click I Agree.
5. Select an install for “Just Me” unless you’re installing for all users (which requires Windows Administrator privileges).
6. Select a destination folder to install Anaconda and click Next.
7. Choose whether to add Anaconda to your PATH environment variable. Unless you plan on installing and running multiple versions of Anaconda, or if you want to limit the length of your PATH variable, you should accept the default and leave this box checked.
8. Choose whether to register Anaconda as your default Python 3.6. Unless you plan on installing and running multiple versions of Anaconda, or multiple versions of Python, you should accept the default and leave this box checked.
9. Click Install. You can click Show Details if you want to see all the packages Anaconda is installing.
10. After a successful installation, you will see the “Thanks for installing Anaconda” image:



**Thanks for installing Anaconda!**

Anaconda is a modern open source analytics platform powered by Python.

Share your notebooks, packages and environments on Anaconda Cloud!

☑ Learn more about Anaconda Cloud

**Scikit Learn-** If we have already installed python and anaconda we can use these two commands to install scikit learn-

```
pip install -U scikit-learn
```

or

```
conda install scikit-learn
```

## Jupyter- Ipython Notebook

Installing Jupyter using Anaconda and conda.

Installation steps:

1. Download Anaconda. Recommendation is downloading Anaconda's latest Python 3 version (currently Python 3.5).
2. Install the version of Anaconda which you downloaded, following the instructions on the download page.
3. To run the notebook:

```
jupyter notebook
```

or just use this command

```
pip3 install jupyter
```

## 8. RECOMMENDATION FOR ENHANCEMENT:

There can be few enhancements that can be made such as- allowing this machine to train itself using the data which has been annotated previously. Using bag of words approach is deeper manner. Implementing the flask- web development micro framework for python. Microframework is a term used to refer to minimalistic web application frameworks. It is contrasted with full-stack frameworks. Flask is recommended because we can create an API that can provide predictions based on a set of input variables using a pickled model. Also, scikit learn is an intuitive and powerful python machine learning library that makes training and validating many models easy. It can be persisted to avoid retraining the model every time they are used. That is why flask will be best suitable for this kind of application. I am still working on this part so as to make an API for this project.

# BIBLIOGRAPY

- http://approximatelycorrect.com/2017/01/23/is-fake-news-a-machine-learning-problem/
- https://miguelmalvarez.com/2017/03/23/how-can-machine-learning-and-ai-help-solving-the-fake-news-problem/
- https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/
- http://oliviaklose.com/machine-learning-11-algorithms-explained/
- http://sebastianraschka.com/Articles/2014_intro_supervised_learning.html
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.190.1441&rep=rep1&type=pdf
- http://scikit-learn.org/stable/supervised_learning.html#supervised-learning
- https://en.wikipedia.org/wiki/Machine_learning
- http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html
- http://www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html

- http://scikit-learn.org/stable/tutorial/basic/tutorial.html
- http://scikit-learn.org/stable/user_guide.html
- http://scikit-learn.org/stable/auto_examples/index.html
- https://web.stanford.edu/~gentzkow/research/fakenews.pdf
- Shrawan Kumar Trivedi. A Study of machine learning classifiers for spam detection
- Francisco Villegas Alejandre. Feature selection to detect botnets using machine learning algorithms
- Abbasi, A. and Chen, H. "A Comparison of Tools for Detecting Fake Websites," IEEE Computer