

# **CHAPTER I**

## **INTRODUCTION**

## 1.1 INTRODUCTION:

Data or information is the valuable asset in this century. The most important problem to be solved is to evaluate whether the data is relevant or irrelevant. Fake data has a huge impact on lot of people and organizations that may even lead to the end of the organization or panic the people.

Machine learning researchers believe that this problem can be solved using the machine learning algorithms and there is lot of on-going research in this field which lead to the new branch called Natural Language Processing.

This classification is not that simple there are lot of challenges to go through in order to succeed. Let's start with few of them machine learning works with the data if you are having huge and clean data then there was a great chance of creating great classifier. In order to create a real time application, the algorithm should be fed with the most recent data. Data is of different sizes so that should be properly cleaned to get better results.

The typical work flow of the classification task consists of collecting data, then cleaning, removing stop words, using term frequency inverse document frequency vectorizer (IDFV) or count vectorizer and then train the algorithm by using the above vectorized data. Else people will jump directly to the Deep learning which is time consuming and resource consuming.

Where as in hybrid algorithm five algorithms have been taken and then they are trained separately. Then to classify the new input data hybrid algorithm will use all the algorithms which have been trained already and then it will predict based on their predictions.

The goal of this research is to analyse and determine the use of different algorithms at the same time by creating a complex algorithm is more accurate. Then various scores like precision, recall, and f1\_score has been validated. To prove that this is better than the Deep Learning.

## 1.2 ABOUT:

The Hybrid algorithm implemented is a combination of five algorithms using all of them at a same time has increased the accuracy of the prediction system. This model can be used in the areas where accuracy is more important than any other factor like Fake News Detection, Deciphering, and so on. A fake news detection application is created to verify and validate the implementation of the algorithm. The website is equipped with lot of functionalities like account management, data handling, storing and retrieving data, explaining algorithm.

The list of algorithms that has been used here are as follows *Naïve Bayes*, *Random Forest*, *Decision Tree*, *Support Vector Machine*, *K Nearest Neighbours*. Based on them the combined hybrid algorithm has been created.

## 1.3 SUMMARY:

A group is always a better option compared with the single. When making any decision if a proper group survey is help then the decision will be proper. So, in place of single algorithm five algorithms will answer the same question which was posted then mode operation will be applied on all the results based on which the most repeated will be considered as the final result. A fake news detection website will be designed to verify and validate the algorithm.

## **CHAPTER II**

### **LITERATURE REVIEW**

## 2.1 LITERATURE REVIEW:

This is a data driven world anything can be solved if proper data has been identified internet made our lives lot easier than we imagine. But the information is being manipulated by lot of people for their own reasons [2]. In order to make a clear idea lets consider data as news which is very important if there is a manipulation in news that may lead to huge asset and human loss [6]. So, in order to solve the problem reader should be able to understand the news they read was real or fake. In order to get into a statistical conclusion a manual testing was conducted and it resulted in an accuracy of 65% for humans [1]. This result is pretty low so to create a better way researcher started using machine learning algorithms to complete the task, surprisingly 75% of accuracy achieved with a simple Naïve Bayes algorithm [1]. This gave the conclusion that machine learning algorithms are saviours in this field. So, people started researching various machine learning algorithms to find the best the algorithm for the above explained scenario [8]. It all started with Naïve Bayes [1][2] which results in 85% accuracy, then it followed by SVM [4] which results in 84% accuracy, then by the Random Forest algorithm [15] which gave an accuracy score of 82%. Similar kind of researches are carried out on all the algorithms like K Nearest Neighbours [10], Decision Tree [15], Logistic Regression [14] and so on. The data required for this analysis is collected from online sources like social media like twitter, Facebook, [3][13]and also from data sharing websites like Kaggle.

Though its not a proven factor all the time everyone knows that combined power is always better compared to the singular. So, to get the better results developer should combine the algorithms [7]. The best way is to conduct a voting mechanism before taking decisions which will save the complete project from one wrong prediction algorithm [7][9]. This will definitely increase the accuracy score of the fake data prediction system. Because this is not developed for one requirement, it can be used for multiple cases based on the data availability.

## **2.2 SUMMARY:**

The use of internet and services are increasing everyday similarly the internet gambling's are simultaneously increasing by communicating fake news with the users. This article clearly explains that a very simple machine learning algorithm can outperform the humans in the area of identifying the fake news. As numbers humans are 65% accurate simple algorithm is 75% accurate. This will definitely help users to take the better decisions.

# **CHAPTER III**

## **PROJECT OVERVIEW**

### **3.1 PROJECT OVERVIEW:**

The project is about creating a fake news detection website which is powered by an ensemble hybrid algorithm which is a combination of multiple independent algorithms.

Fake data is the most important problem to be addressed because a single fake news may lead to a great asset and human loss. The algorithm used in this project is 3% more accurate than any other algorithm that has witnessed. By this most of the data will get classified correctly which will save lot of people from loss.

### **3.2 HARDWARE REQUIREMENTS:**

- Processor: Intel I5 or more with 1.5 GHz or more
- RAM: 10 GB or More
- Hard Disk: 500 GB or More based on the training data

### **3.3 SOFTWARE REQUIREMENTS:**

- Programming Language: Python 3.5 or Higher
- Web Framework: Django 2.0 or Higher
- Database: Sqlite3
- Supporting Application: Anaconda
- Analysing software: Tableau
- Operating System: windows OS 10
- Supporting software's: SMTP, FTP .....



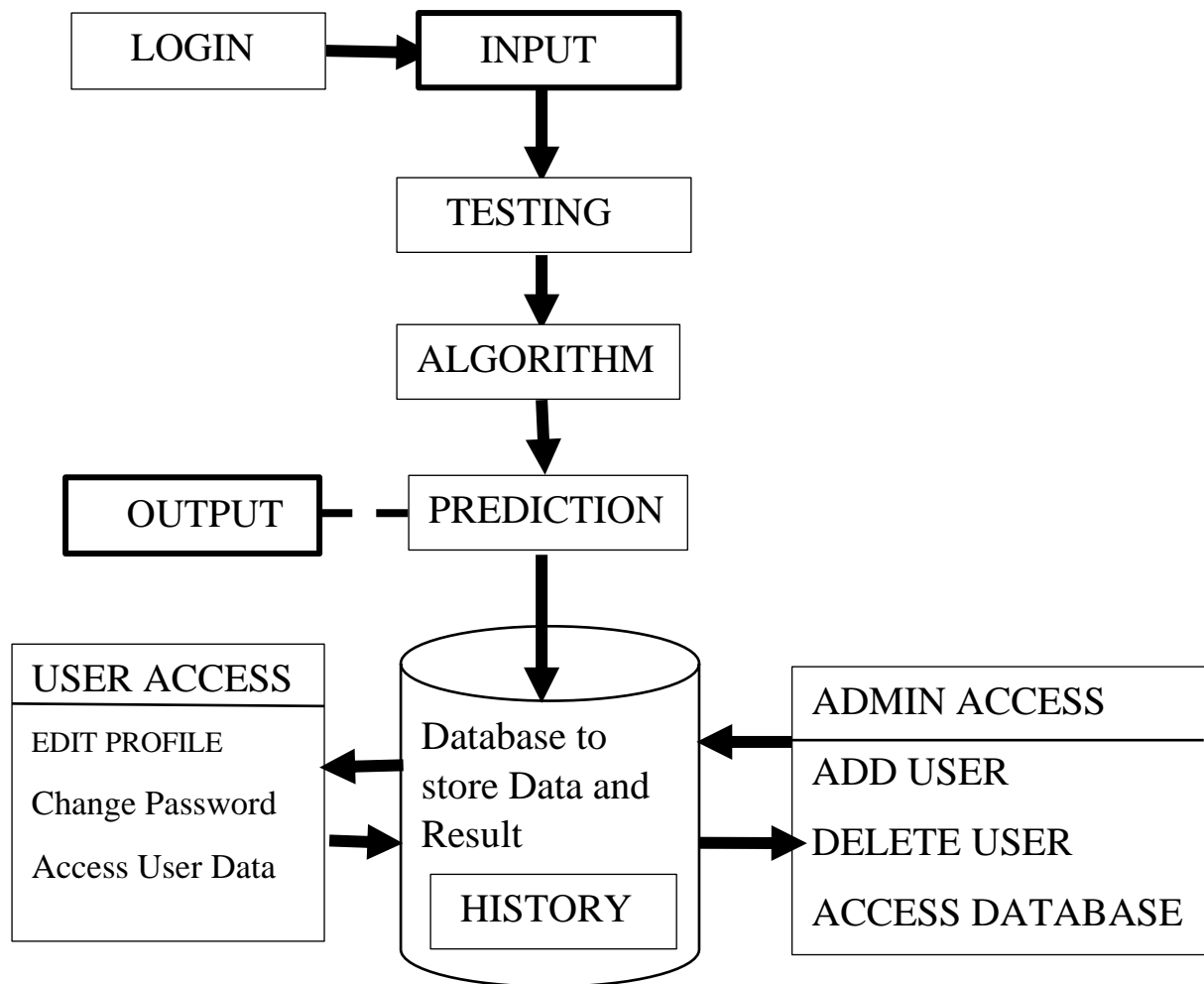


Fig 3.4: Block diagram of Fake Data Analysis website

### 3.4 BLOCK DIAGRAM DESCRIPTION:

The block diagram of the fake data detection application consists the following steps and modules.

#### 3.4.1 Website:

In order make people access the program from anywhere website is the best way. So, website is designed using Django a python web framework which is an MVC framework so its easy to maintain the webpage and make any changes. Website is created with different modules they are clearly explained in the coming modules.

### 3.4.2 Login:

In order to keep track of the user data, to provide service as per the user request like creating and deleting the account and forget password this makes the project complete and private so this module is embedded to the site as a first and foremost step.

### 3.4.3 Testing:

The data that's provided by the user for predicting is not in the format that algorithm need. So, the data will get passed through the testing phase. Its actually a regular expression validator which will remove all the unnecessary content like spaces, quotes, numbers etc. As a final result completely, processed data will be provided which can be used as a direct input for the Hybrid algorithm created.

### 3.4.4 Algorithm:

This is the most important step of the project. This is a combination of different algorithms in one place.

The list of algorithms that has been used here are as follows *Naïve Bayes, Random Forest, Decision Tree, Support Vector Machine, K Nearest Neighbours*. Based on them the combined hybrid algorithm has been created.

Every algorithm will be trained with the data provided and pickled and stored as a pickle file by which prediction time will be greatly reduced because

The results of the algorithm are compared with different datasets under different conditions. As a result, it can be concluded that this algorithm process works in the same way for most of the cases.

Complete analysis of the algorithm will be done in the below sections. Follow that for more information on the algorithm.

### 3.4.5 Prediction:

When the data is being processed by the algorithm. It will return the output of all the six algorithms including the hybrid algorithm all these results will get passed to the output page and the database.

### 3.4.6 Output:

Output page will read data from the prediction function and then it will display the data to the user in a card view. Where each card view represents one algorithm. Each card has data as follows algorithm name, result of algorithm for particular data, date and time of search.

### 3.4.7 Database:

The database used here to save data is sqlite3. This store the result data, user access data, search history. Each will get maintained for every user separately to retrieve data for future references.

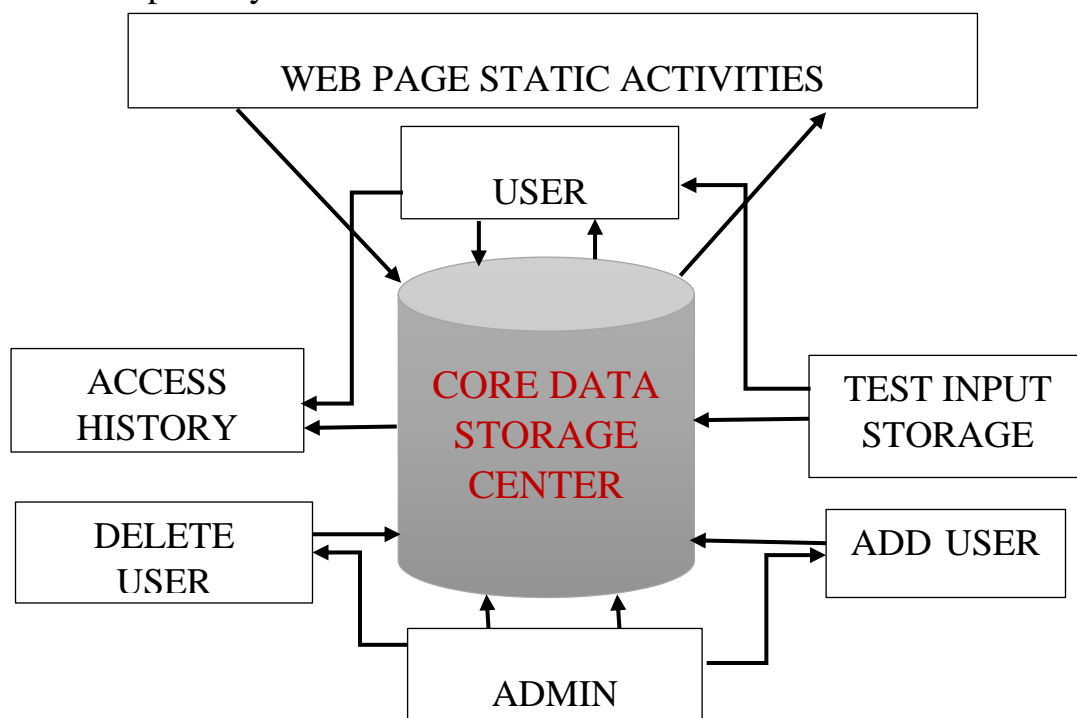


Fig 3.1.7: Database architecture diagram of the project

### **3.4.8 History:**

This is a page in the website dynamic for every user when the user wants to check his personal search history, they can open this page. It has cards where each card refers to the article they have searched for. In order to have complete insight they can open the card and analyse the data they have searched.

### **3.4.9 User Access:**

There are two kinds of requesters for the website. They are user and admin each have different rights reserved in order to reduce ambiguity between the users and to solve the situations of problem.

- A user can edit his profile if he/she want to change anything in the data that they have provided.
- Then can request for a change in their password for security reasons.
- They can access the previous history they have searched in the website.
- They can see the about us page where every algorithms performance was explained along with the hybrid algorithm.

### **3.4.10 Admin Access:**

An admin directly inherits all the user functions he can perform all the actions a typical user can perform along with that they have extra rights as follows.

- He/she can add or delete a user based on the situation.
- They can review all the user searches and these searches can be used to further improvement in the algorithm

# **CHAPTER IV**

## **ALGORITHM ANALYSIS**

## **4.1 INTRODUCTION:**

This section is about analysing accuracy, precision, recall and F1 scores of various algorithms and introducing a combined hybrid algorithm which is the combination of all the algorithms.

Let's see how a machine learning algorithm is capable of predicting whether a data is fake or real. It should definitely have some properties let's analyse them and get to the statement.

## **4.2 FAKE DATA PROPERTIES**

In order to declare any data as fake data there should be some checks if the data didn't pass those checks then that data will be labelled as the fake data.[1][2]

- They have lot of similar words in their repository
- They are having some emotional or offer words
- They are mostly opposing the already known facts
- Then content is not acceptable because of their contradictory behaviour

Based on the above statements there are lot of relations between different spam and fake messages. If people can classify a huge set of messages as fake or not then by studying all those articles, using that we can classify the other articles as fake or real by using already classified data as training set for the machine learning model.

In the above scenario a human is being replaced by a machine which is perfectly crafted to find the relation between the items. Along with the above statements it may also find lot of other relations which are not understandable by a human being.

## **4.3 TRAINING TESTING DATA:**

Data has been collected Kaggle data world. It has four columns and they are index, title, text and label of the various news articles of various journals.

Among the four text is used as an independent variable and label is used as a dependent variable. There is no change in the accuracy scores even with the use of titles because the title words will get repeated in the text column.

Multiple datasets have been collected from various sources among them three datasets were used in order to get the conclusion that the algorithm works for all types of data and didn't biased for some kind of data.

The train and test data split were in the ratio of 80 – 20 using random function where the train set is used for the training purpose and test set is used for the testing purpose.

#### **4.4 ANALYSER DEFINITIONS:**

The f1-score, precision, and recall are showcased below. The formulas for calculating them are as below

Precision is defined as the fraction of the relevant instances over the retrieved instances as true.

$$\diamond \text{ **Precision** } = \text{True Positive} / (\text{True positive} + \text{False Positive})$$

Recall is defined as the fraction of the relevant instances over the total relevant instances.

$$\diamond \text{ **Recall** } = \text{True Positive} / (\text{True positive} + \text{False Negative})$$

F1-score is based on both the precision and the recall values it is defined as the harmonic mean of the precision and recall

$$\diamond \text{ **F1-score** } = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

Both precision and recall are very important for a model because of which F1-score is used as a standard measure for the model comparison.

The list of algorithms analysed and used in this project is as follows

- Naïve Bayes algorithm
- Support Vector Machine
- Random Forest algorithm
- Decision Tree algorithm
- K Nearest Neighbours algorithm.

Let's know about each of them and clearly visualize each of their performances and come to the conclusion.

#### **4.5 NAÏVE BAYES ALGORITHM:**

Naïve Bayes classifier is a simple probability-based classifier based on the Bayes theorem with great (naive) independence assumption between the data features, where class labels picked from some finite set. It is not a one single algorithm to train such classifiers, but a collection of multiple algorithms based on one common principle: every naive Bayes classifier assumes that the value of a given feature is purely independent to the value of any other given feature, given the class variable.

Naïve Bayes is the most opted statistical technique for the models like email filtering, spam filtering and so on.

Naïve Bayes works on the bag of words features where the data of different articles collected is converted into encoded format by using various vectorization techniques based on the requirement some of them are count vectorizer (CV), term frequency and inverse document frequency vectorizer (TFIDF)

The Bag of words will be passed to the Naïve Bayes model as a training data and based on the data the model will learn



Then when any article is passed to classify vectorizer will create sparse matrix and then model will predict based on the word distribution in the sparse matrix.

$$\diamond \Pr(F|W) = \Pr(W|F) \cdot \Pr(F) / (\Pr(W|F) \cdot \Pr(F) + \Pr(W|T) \cdot \Pr(T)),$$

where:

- $\Pr(F|W)$  – conditional probability, fake data when the word present in the article;
- $\Pr(W|F)$  – conditional probability of finding the word W in fake data articles;
- $\Pr(F)$  – overall probability that the given data is fake data;
- $\Pr(W|T)$  – conditional probability of finding the word W in real data articles;
- $\Pr(T)$  – overall probability that given data is true data. This formula is based on the Bayes' theorem.

$$P(c/x) = \frac{P(x|c) P(c)}{P(x)}$$

Likelihood Class Prior probability  
 Posterior Probability Predictor Prior Probability

$$P(c/X) = P(x_1/c) \times P(x_2/c) \times \dots \times P(x_n/c) \times P(c)$$

**Accuracy score achieved by Naïve Bayes algorithm is 0.8966**

#### 4.5.1 Confusion Matrix Visualization:

- Confusion matrix is a great way of analysing a machine learning model. This has the data of *True positive*, *False positive*, *False negative*, *True Negative*.

- The confusion matrix of Naïve Bayes is as follows [526, 87, 44, 610]

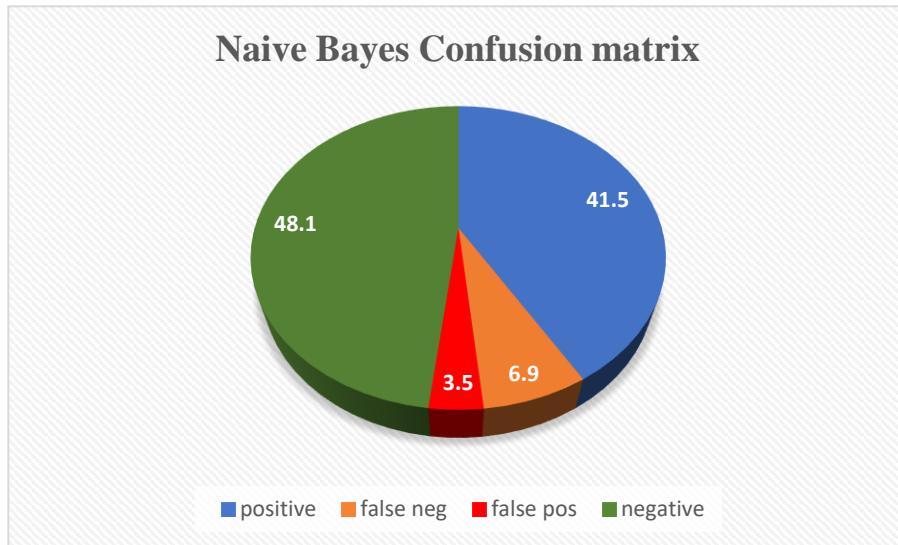


Fig 4.5.1.1: - pie chart representation of Naïve Bayes confusion matrix

- **Precision – 0.92**
- **Recall – 0.85**
- **F1\_score – 0.89**
- **Accuracy score - 0.8966**

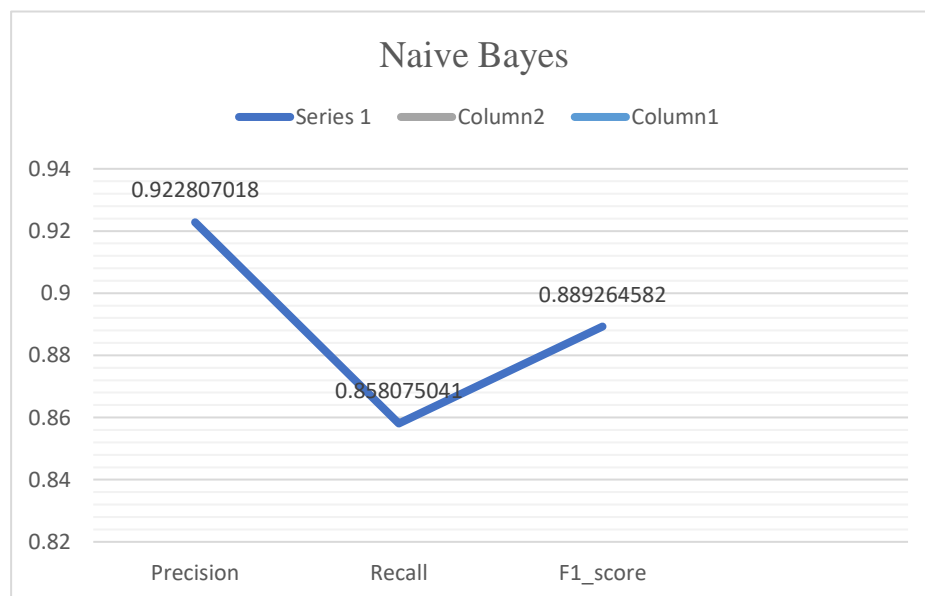


Fig 4.5.1.2: - Naïve Bayes analysis line chart

## 4.6 SVM FOR FAKE DATA ANALYSIS

A Support Vector Machine (SVM) is a discriminative labelled classifier where discrimination is achieved by creating a separating hyperplane. When labelled training data (*supervised learning*) is provided, the algorithm will output an optimized hyperplane which can categorize new examples. The dimensions of the plane are truly based on the count of independent variables in the training data.

Classifying the data is most common task in the machine learning field. Suppose some data points are provided each belong to the one of given two classes, and the primary goal is to predict in which class a new data point will be entered. In case of support-vector machines (SVM'S), a data point is represented as a  $p$ -dimensional vector, and our sole interest lies in finding a plane which can separate all the given points with the help of  $(p-1)$  dimensional hyperplane. This was called as a linear classifier. There are wide range of hyperplanes that might classify the same data. One reasonable choice is to find the best fitting hyperplane which can separate most of the data points in two classes. So, such a plane has to choose whose distance is maximum to the nearest point on each side of the plane. If there is such hyperplane, it is called as the maximum margin (MMH) hyperplane and the linear classifier it defines is called as a maximum-margin (MMC) classifier.

SVM is more robust and accurate algorithm compared with the other algorithms over there. This is based on the kernel trick; which is widely used in the distance-based classification tasks.

Support Vector Machine works on the bag of words features where the data of different articles collected is converted into encoded format by using various vectorization techniques based on the requirement some of them are count

vectorizer (CV), term frequency and inverse document frequency vectorizer (TFIDF)

**Accuracy score achieved by Support Vector Machine (SVM) algorithm is 0.8918**

#### **4.6.1 Confusion Matrix Visualization:**

Confusion matrix is a great way of analysing a machine learning model. This has the data of *True positive*, *False positive*, *False negative*, *True Negative*.

The confusion matrix of SVM is as follows [575, 38, 99, 555]

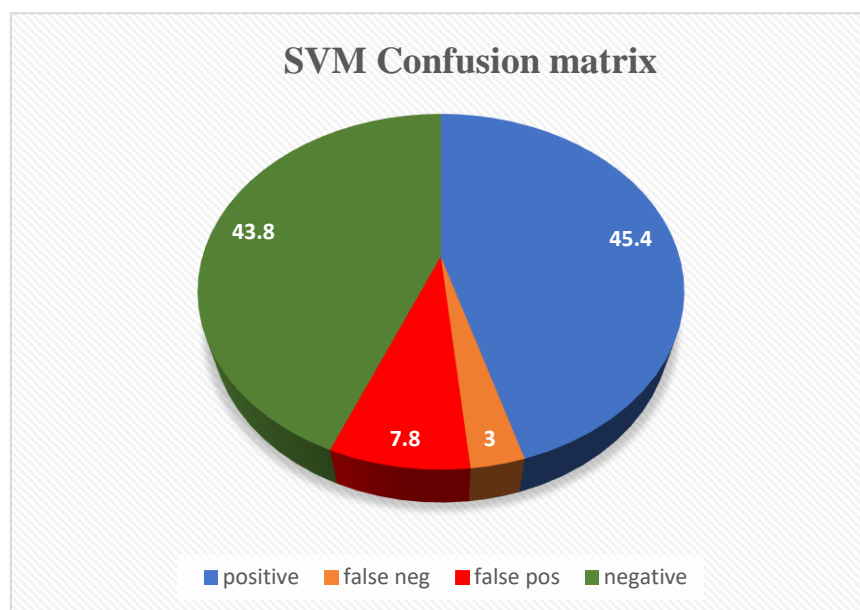


Fig 4.6.1.1: Pie chart representation of SVM confusion matrix

- **Precision – 0.85**
- **Recall – 0.93**
- **F1\_score – 0.89**
- **Accuracy score - 0.8918**

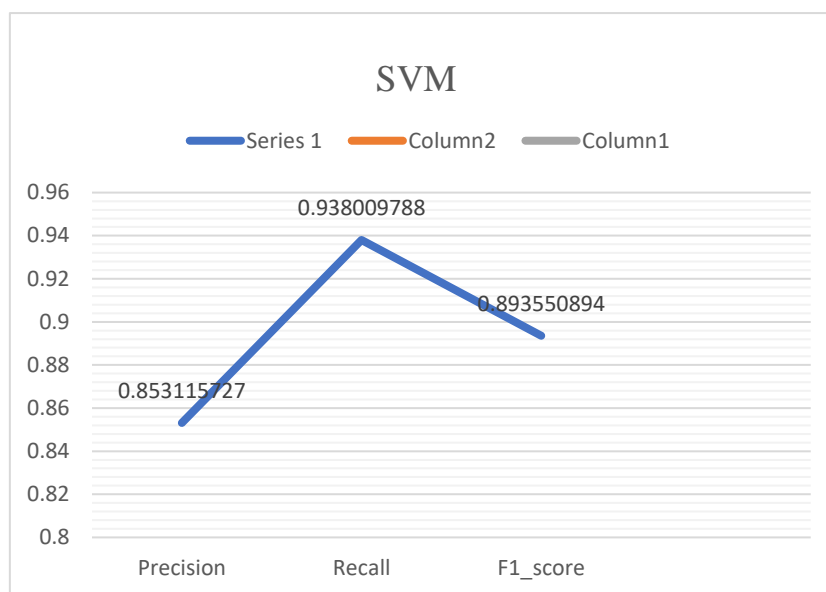


Fig 4.6.1.2: SVM analysis line chart

#### 4.7 RANDOM FOREST FOR FAKE DATA ANALYSIS:

Random forests are ensembled learning method for the classification, regression and some other tasks that are operated by constructing a multitude of the decision trees at the time of training and outputting the class that is one among the training data(classification) or mean predicted value (regression) of the individual trees calculated. Random decision forests are best to solve the overfitting problem of the training data using the decision tree classifier.

Random Forest algorithm is a supervised learning algorithm. You can already get from the name itself; the algorithm will create a forest and will makes it random some way or the other. The forest that it builds, is an ensemble implementation of Decision Trees, mostly it is trained using the bagging method. The core idea of bagging method is that a combined usage of learning models will increases the final result.

One major advantage of the random forest is, it can be used for both the regression and classification algorithms. This is the case for all the current machine learning algorithms.

Random Forest works on the bag of words features where the data of different articles collected is converted into encoded format by using various vectorization techniques based on the requirement some of them are count vectorizer (CV), term frequency and inverse document frequency vectorizer (TFIDF)<sup>4</sup>

**Accuracy score achieved by Random Forest algorithm is 0.8626**

#### 4.7.1 Confusion Matrix Visualization:

Confusion matrix is a great way of analysing a machine learning model. This has the data of *True positive*, *False positive*, *False negative*, *True Negative*.

The confusion matrix of Random Forest is as follows [547, 66, 108, 546]

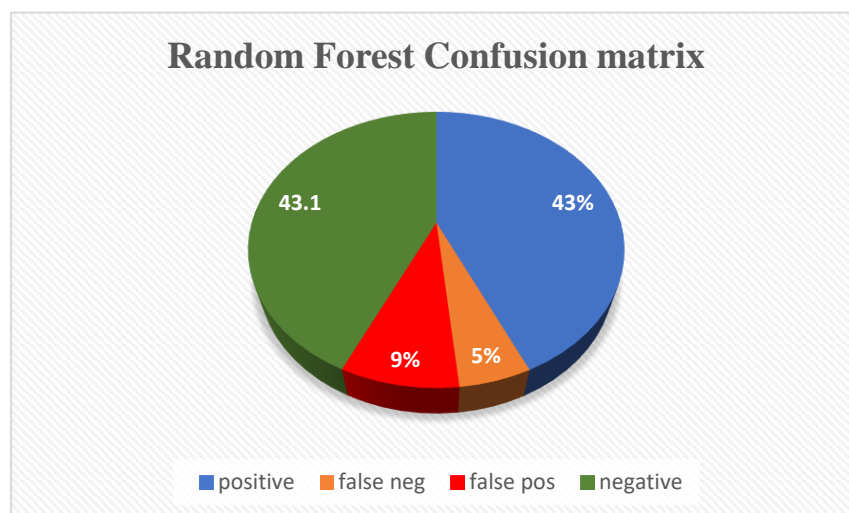


Fig 4.7.1.1: Pie chart representation of Random Forest confusion matrix

- **Precision – 0.85**
- **Recall – 0.89**
- **F1\_score – 0.86**
- **Accuracy score - 0.8626**

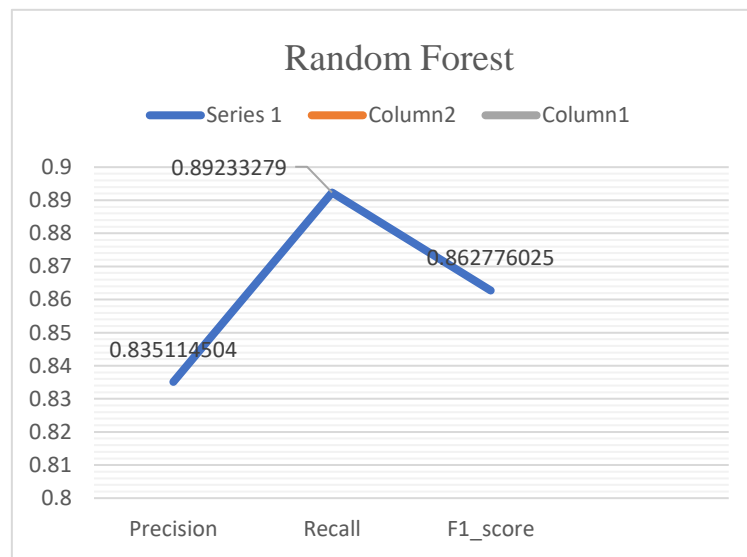


Fig 4.7.1.2: Random Forest *analysis line chart*

#### 4.8 KNN FOR FAKE DATA ANALYSIS:

In data analysis, the k Nearest neighbours(k-NN) algorithm is a non-parametric algorithm which can be used for both regression and classification. In either case, the input consists of the k closest neighbouring training examples in the feature vector. The output will depend on whether k-NN is used for classification or regression task.

KNN is a lazy algorithm. It doesn't mean that KNN is not doing anything. This means that it doesn't use training data to do any kind of generalization. There is no explicit training phase for the KNN algorithm, though it's there its very minimal compared to the other. This also reflects the fact that the training process for the KNN algorithm is very fast comparatively. This also implies the fact that algorithm needs the entire data in order to perform any operations. So, it requires huge storage space for the testing phase.

K Nearest Neighbour's, works on the bag of words features where the data of different articles collected is converted into encoded format by using various vectorization techniques based on the requirement some of them are count

vectorizer (CV), term frequency and inverse document frequency vectorizer (TFIDF)

### Distance Functions

Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Manhattan	$\sum_{i=1}^k  x_i - y_i $
Minkowski	$(\sum_{i=1}^k ( x_i - y_i ^q))^{\frac{1}{q}}$

**Accuracy score achieved by K Nearest Neighbour's algorithm is 0.82**

#### 4.8.1 Confusion Matrix Visualization:

Confusion matrix is a great way of analysing a machine learning model. This has the data of *True positive*, *False positive*, *False negative*, *True Negative*.

The confusion matrix of KNN is as follows **[521, 92, 136, 518]**

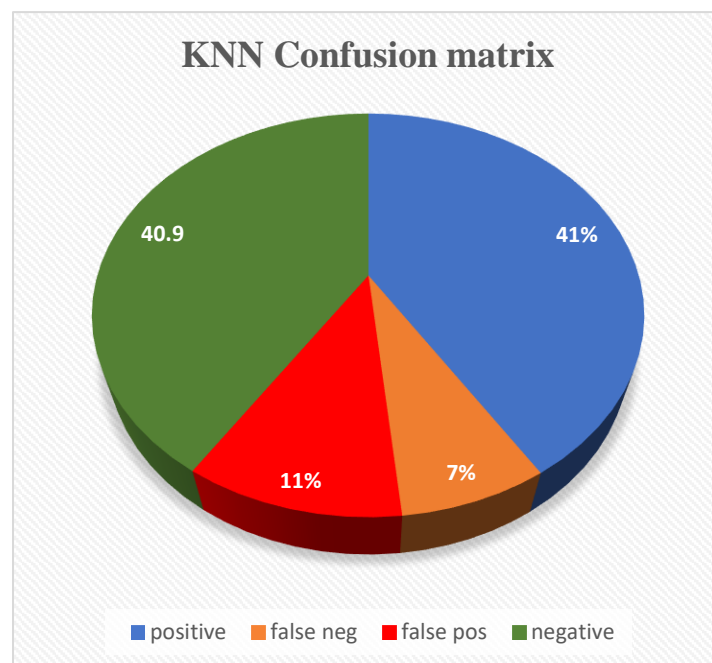


Fig 4.8.1.1: Pie chart representation of KNN confusion matrix



- **Precision – 0.79**
- **Recall – 0.85**
- **F1\_score – 0.82**
- **Accuracy score - 0.82**

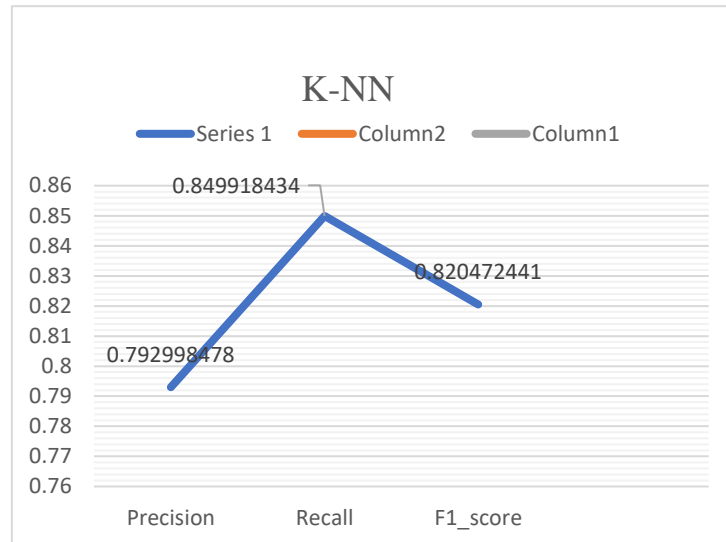


Fig 4.8.1.2: KNN analysis line chart

#### 4.9 DECISION TREE FOR FAKE DATA ANALYSIS:

A decision tree model is a flowchart resembles structure where each node represents a test on an attribute whereas a branch represents the final outcome of the test, and every leaf node represents the label of the class. The paths from root node to leaf node will make the classification rules.

In decision making, a decision tree and a closely related flow diagram is used as the visual and analytical decision support tool, in which the expected values of competing alternatives are calculated by using the flow.

##### ENTROPY:

A decision tree is top-down structure which build from a root node involving partitioning the data into various subsets that contain units with similar values (homogeneous). ID3 algorithm based on entropy to calculate the homogeneity of given sample. If the sample is completely homogeneous

the entropy is zero and if the sample can be equally divided then it results in entropy of one.

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

$$\begin{aligned}
 \text{Entropy (Play Golf)} &= \text{Entropy (5,9)} \\
 &= \text{Entropy (0.36, 0.64)} \\
 &= - (0.36 \log_2 0.36) - (0.64 \log_2 0.64) \\
 &= 0.94
 \end{aligned}$$

### INFORMATION GAIN:

The information gain value is totally based on the reduction in entropy value after a dataset is divided based on an attribute. Forming a decision tree from data is all about detecting the attribute that returns the highest information gain. This means the branch is the most homogeneous branch.

$IG(T, X) = \text{Entropy}(T) - \text{Entropy}(T, X)$  where IG stands for the information gain

Gini index defines as, if two items has been selected from a population in a random way then both of them from the same class and probability for this action is 1 if population is totally pure.

- It works even with categorical variable *Success or Failure*.
- It can perform Binary splits only.
- Gini Index value and the homogeneity of the attribute are directly proportional.

Classification and Regression Tree (CART) uses Gini method for the binary splits.

$$\text{Variance} = \frac{\sum (x - \bar{x})^2}{n}$$

Decision Tree, works on the bag of words features where the data of different articles collected is converted into encoded format by using various vectorization techniques based on the requirement some of them are count vectorizer (CV), term frequency and inverse document frequency vectorizer (TFIDF)

**Accuracy score achieved by Decision Tree algorithm is 0.8279**

#### 4.9.1 Confusion Matrix Visualization:

Confusion matrix is a great way of analysing a machine learning model. This has the data of *True positive*, *False positive*, *False negative*, *True Negative*.

The confusion matrix of Decision Tree is as follows [495, 118, 97, 557]

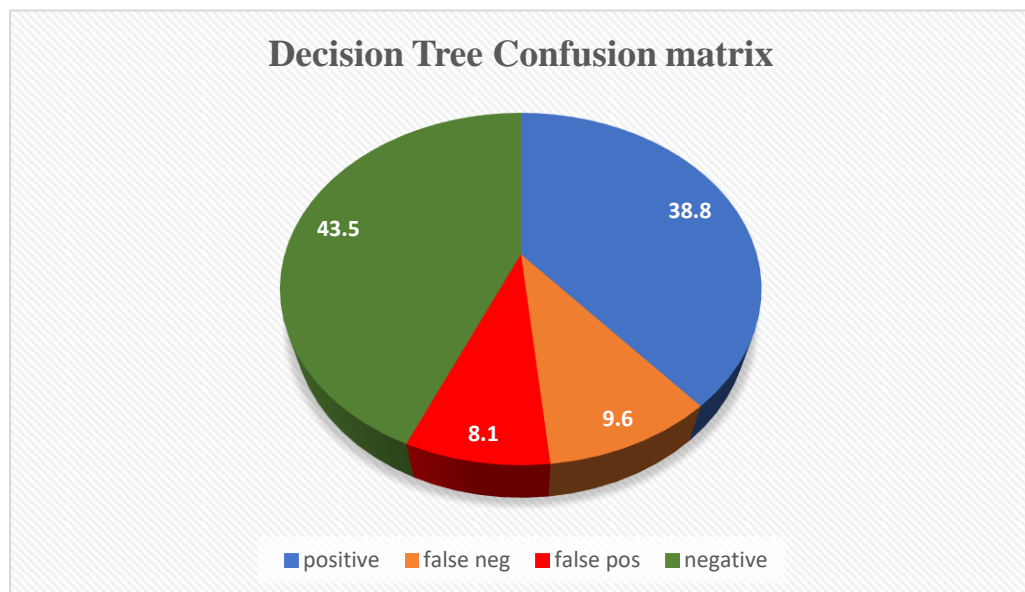


Fig 4.9.1.1: Pie chart representation of Decision Tree confusion matrix

- **Precision – 0.83**
- **Recall – 0.80**
- **F1\_score – 0.82**
- **Accuracy score - 0.8279**

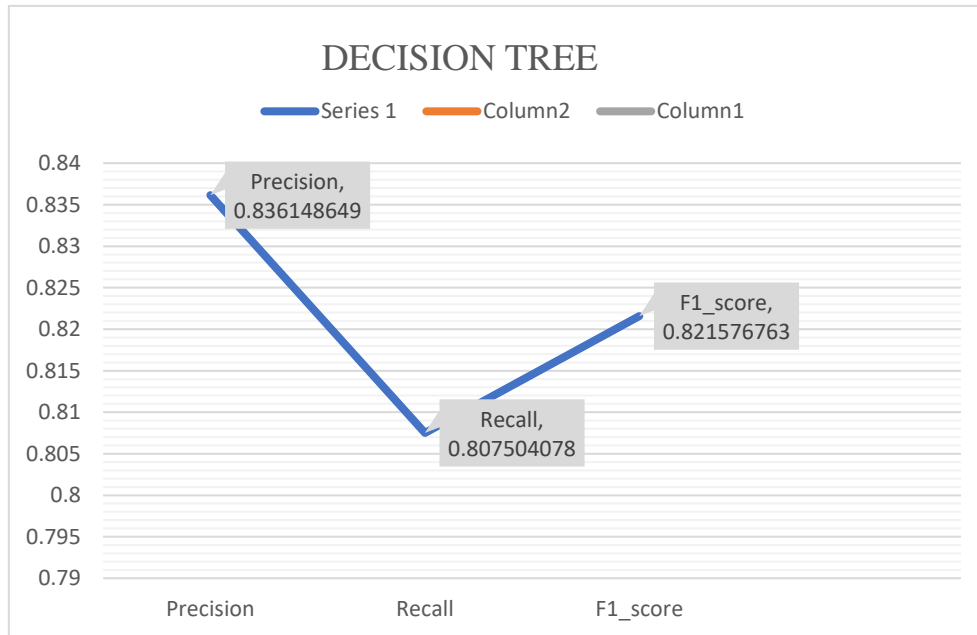


Fig 4.9.1.2: Decision Tree analysis line chart

#### 4.10 IMPLEMENTATION OF THE ENSEMBLED ALGORITHM:

The hybrid algorithm proposed is an ensembled algorithm like the random forest which works on the means of the list of algorithms that have been used here is as follows **Naïve Bayes, Random Forest, Decision Tree, Support Vector Machine, K Nearest Neighbours**. Based on them the combined hybrid algorithm has been created. All these algorithms use training data as the bag of words model which was created using Count Vectorizer.

By combining all the algorithms, a great change was witnessed in the precision, recall, F1 score, and accuracy values of the test data so this hybrid machine learning algorithm can be used in place of neural network by which time and computational power can be saved.

#### ALGORITHM: -

Input = test data article

Procedure:

C.V = Count Vectorizer (Input) a vectorized matrix created from the input test data

S.M = (np.array(C.V)) a sparse matrix is generated from the vectorized data

This sparse matrix will be passed to different trained machine learning algorithms to get the result

Naive () = trained (multinomial naive Bayes algorithm)

NaivePred = Naive (S.M) result from the trained naive Bayes algorithm

SVM () = trained (Support Vector Machine algorithm)

SVMPred = SVM (S.M) result from the trained SVM

KNN = trained (K Nearest Neighbour algorithm)

KNNPred = result from the trained naive Bayes algorithm

Decision = trained (Decision Tree algorithm)

DecisionPred = result from the trained Decision Tree algorithm

Random Forest () = trained (Random Forest algorithm)

RandomForestPred = result from the trained RandomForest algorithm

Predictions from all the above algorithms are either 1 or 0 where 1 denotes the real value and 0 denotes the fake value

Ensembled algorithm will be based on all the above algorithm results

HEVA = Hybrid Ensembled Voting algorithm

HEVA = NaivePred + SVMPred + KNNPred + DecisionPred + RandomForestPred

if HEVA greater than or equal to 3 then:

HEVAPred = 1 (So the predicted result will be real)

else:

HEVAPred = 0 (So the predicted result will be fake)

This is considered as the final output of the given input data.

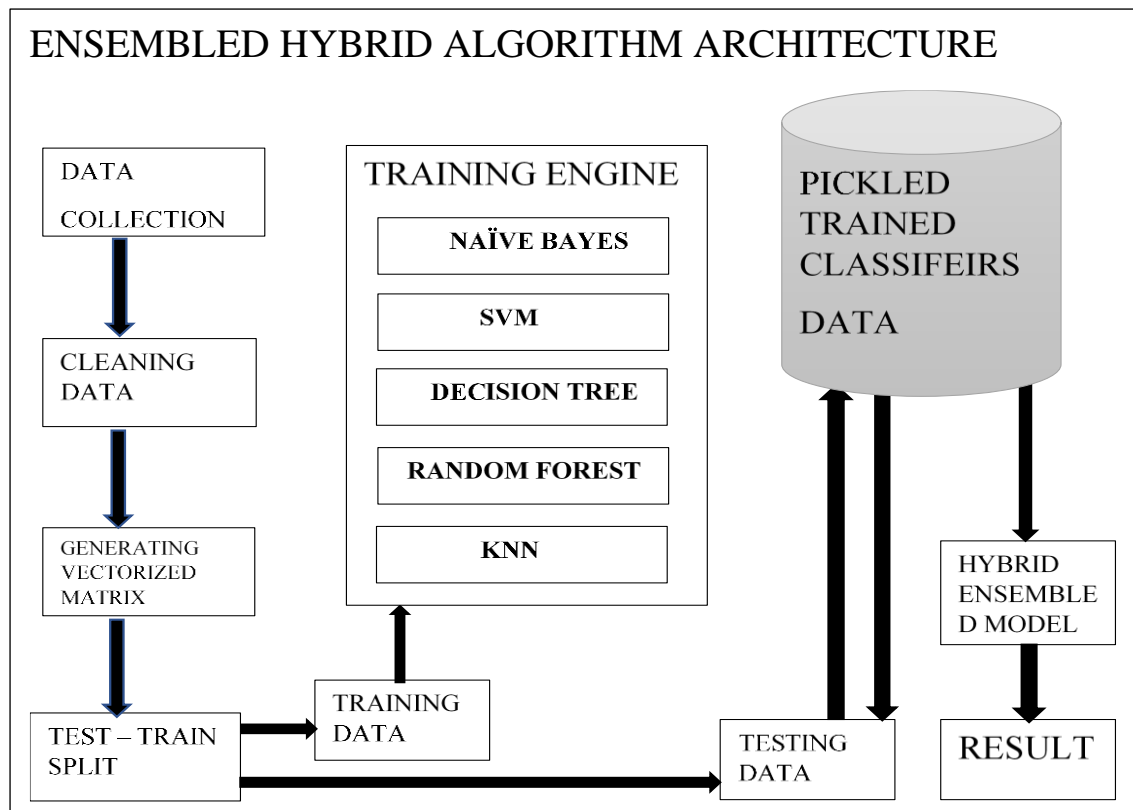


Fig 4.10: Hybrid algorithm architecture diagram

#### 4.10.1 Algorithm Explanation:

1. As a first step data will get collected from various resources examples online data providers like Kaggle, UCI and so on else using web scraping data can be collected or else by approaching people and directly getting data. Anyway, the result of this step will hold the data required for the entire process
2. What every be the origin of the data. The data will not be in the form that can directly go through all the other processes. So there should be a standalone mechanism that can extract the required data from the entire data by eliminating the unnecessary data.

Steps of cleaning as follows:

- i. Remove all the unnecessary spaces and the weird symbols that people use which won't carry any meaning to the text content that was collected.

- ii. Remove all the stop words in the English language as follows a, about, above, after, again, against, all, am, an, and, any, are, as, at, be, because, been, before, being, below, between, both, but, by, could, did, do, does, doing, down, during, each, few, for, from, further, had, has, have, having, he, he'd, he'll, he's, her, here, here's, hers, herself, him, himself, his, how, how's, I, I'd, I'll, I'm, I've, if, in, into, is, it, it's, its, itself, let's, me, more, most, my, myself, nor, of, on, once, only, or, other, ought, our, ours, ourselves, out, over, own, same, she, she'd, she'll, she's, should, so, some, such, than, that, that's, the, their, theirs, them, themselves, then, there, there's, these, they, they'd, they'll, they're, they've, this, those, though, to, too, under, until, up, very, was, we, we'd, we'll, we're, we've, were, what, what's, when, when's, where, where's, which, while, who, who's, whom, why, why's, with, would, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves. Even though these words are less compared to the huge English language these words occupy more than 65% of the sentence. By eliminating these words 65% of useless words can get removed.
- iii. Stemming is yet another important step of the cleaning. In grammar there are different tenses all carrying the same core meaning but they are only changed because of the tense. In classification there is no need of the word tense. So, all these words can get converted into there root or stem form. This step reduces at least 50% of the remaining words.
- iv. Now the data is ready to use for the next step after all those steps the boy's cars are different colours will get converted into the boy car be differ colour. A change can be clearly witnessed here.

3. Though there is a great increase in the computational power and computers can understand and speak the natural language. Everyone should remember the fact that at the end its still a computing device and can only work on the data that's in the form of numbers. Text is like alien language for our computers.

So, the duty is to convert the text data possessing into the numbers. This thing can be achieved by a technique called encoding there are lot of encoding techniques.

Some of them are Term Frequency Inverse Document Frequency (TFIDF), Inverse Document Frequency (IDF).

At the end of this step program is having the machine understandable format of the entire article.

4. The next important step is to train our data. A prediction is only possible if the algorithm gets perfectly trained. So, by transfer the data to all the algorithms selected each of them get trained by the data. Those algorithms are as follows Naïve Bayes, Support Vector Machine, Decision Tree, Random Forest, K Nearest Neighbours, training the data every time the application get started is not a good practice so this should get stored in a place.
5. The place which store the trained data is nothing but a pickle file. Its nothing but a simple data file where objects can be dumped and extracted based on the need. It acts like a trained algorithms data base.
6. When input is received from the website. It will go through all the algorithms, and everyone will predict the result for the particular text input and all the data will get passed to the hybrid model.



7. Hybrid model is like mode operator. It checks all the results and select the one which result is most common among all the classifiers that was already trained. This is the final result of the algorithm

**Accuracy score achieved by hybrid algorithm is 0.9258.**

The overall accuracy is 3 percent more than any other algorithm that has been used for the creation of the multinomial hybrid voting algorithm.

Let's visualize the output of the test dataset we have used to get better insights about the ensembled hybrid algorithm.

#### 4.10.2 Confusion Matrix Visualization:

Confusion matrix is a great way of analysing a machine learning model. This has the data of *True positive*, *False positive*, *False negative*, *True Negative*

The confusion matrix of Ensembled Hybrid Algorithm is as follows  
[570, 43, 57, 597]

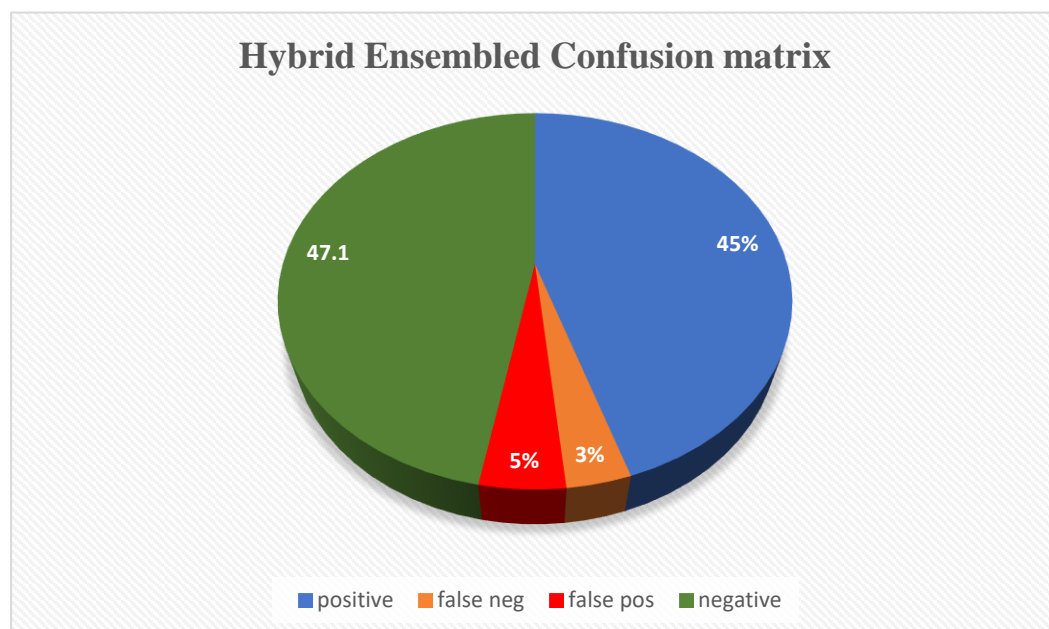


Fig 4.10.1.1: Pie chart representation of Hybrid Ensembled algorithm confusion matrix

- **Precision – 0.91**
- **Recall – 0.93**
- **F1\_score – 0.92**
- **Accuracy score - 0.9258**

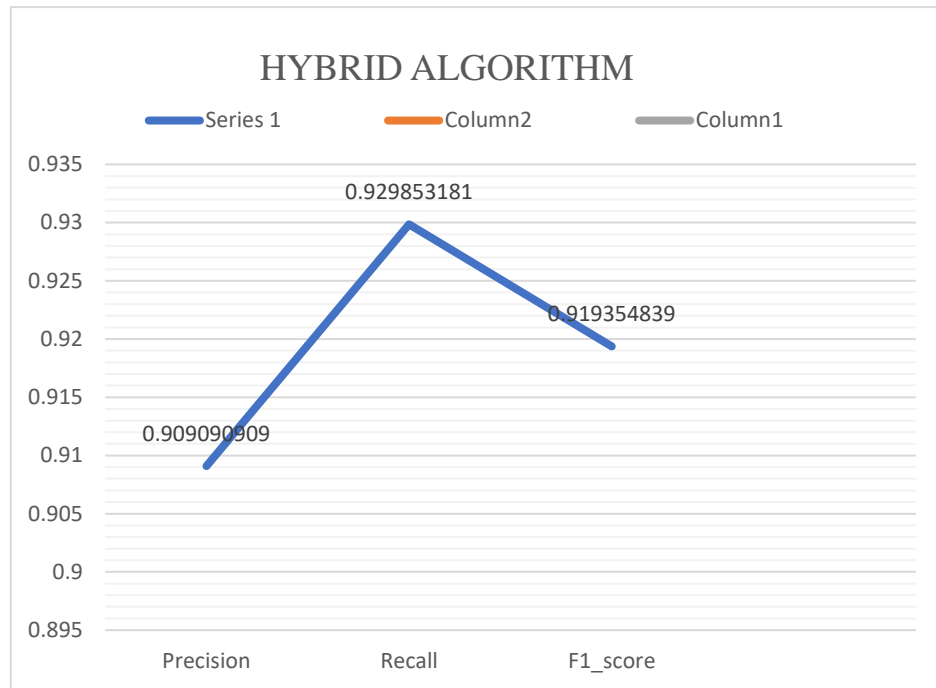


Fig 4.10.1.2: Hybrid Ensembled algorithm analysis line chart

#### 4.10.3 PERFORMANCE ANALYSIS OF PROPOSED ALGORITHM:

This is the combined representation of the precision, recall and F1 scores of all the algorithms in one graph.

The f1-score, precision, and recall are showcased below. The formulas for calculating them are as below

Precision is defined as the fraction of the relevant instances over the retrieved instances as true.

$$\diamond \text{ Precision} = \text{True Positive} / (\text{True positive} + \text{False Positive})$$

Recall is defined as the fraction of the relevant instances over the total relevant instances.

$$\diamond \text{ Recall} = \text{True Positive} / (\text{True positive} + \text{False Negative})$$

F1-score is based on both the precision and the recall values it is defined as the harmonic mean of the precision and recall

$$\diamond \text{ F1-score} = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

Both precision and recall are very important for a model because of which F1-score is used as a standard measure for the model comparison

Based on the below bar graph we can clearly pretend that Hybrid algorithm is performing better among all with 91.93% of F1-score.

Based on the above graphical representation Naïve Bayes is performing best if only precision is considered with 92.2% and then its followed by our Hybrid Algorithm with 91%.

Similarly, SVM is having far better value compared to the hybrid algorithm. Here SVM recall score is 93.8%, whereas Hybrid Algorithm recall score is about 93%.

But in order to declare an algorithm as the best algorithm both the precision and recall scores should be more, but in the above situation if one is having the recall score then precision will be less and vice versa. In order to take correct decisions new score called F1 score is introduced which is a combination of both the scores. This is the final score based on which algorithm performance is calculated.

Based on the graph its clearly visible that our Hybrid algorithm is having the highest F1 score i.e., 92%, whereas the next highest is registered by SVM with 89.3% which is still a great difference.

By this we can conclude that Hybrid algorithm is always a best performer in any case of the dataset provided.

## F1-SCORE, RECALL, PRECISION

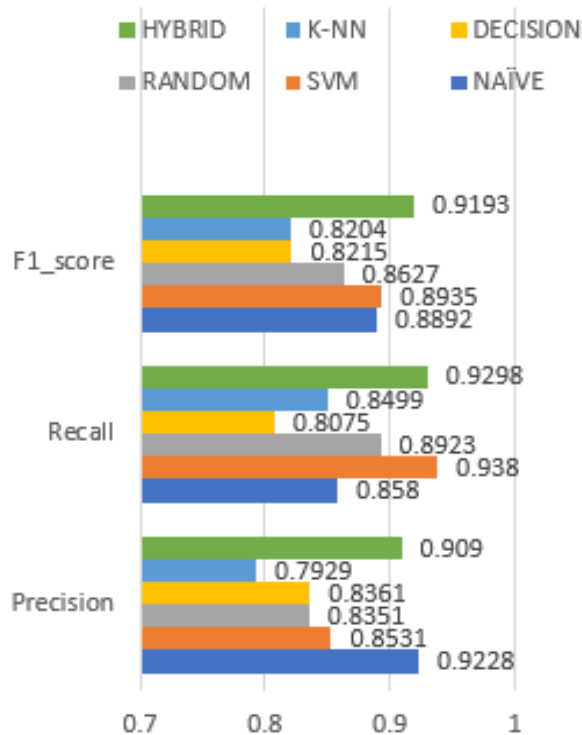


Figure 4.10.3.1: Overall Comparisons

### Accuracy Scores Comparison:

Accuracy scores are yet another important measure for a machine learning model. This score will clearly explain how a model is performing on the test data which is not used for training.

This below drawn graph acts like an evidence. It holds all the algorithms accuracy values, as everyone knows that is the directly, most important and attractive measure of the machine learning algorithm. Because everyone can understand what this score represents.

In that graph Hybrid Algorithm is having the highest score of 92.6% among all the other algorithms.

The next algorithm is Naïve Bayes with an accuracy score of 89.6% which is also a great score.

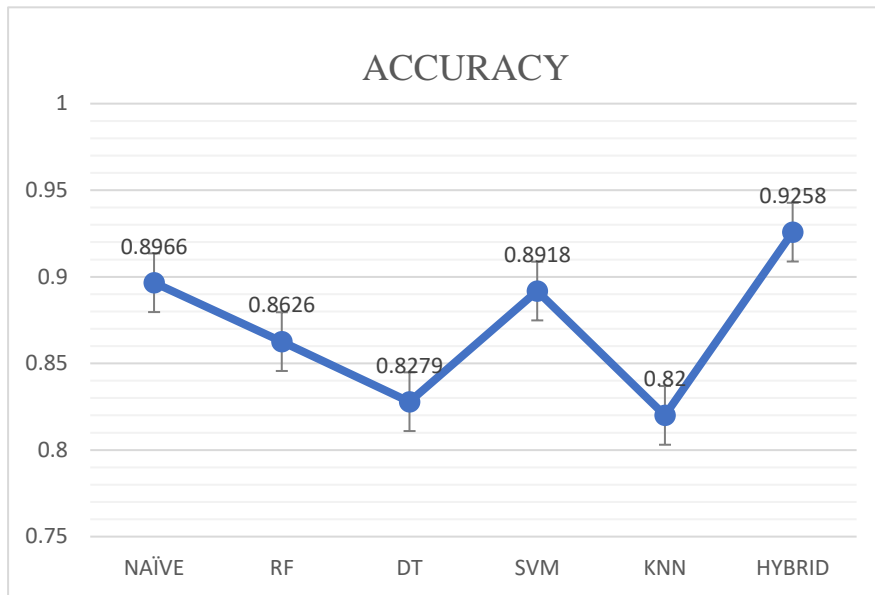


Fig 4.10.3.2: Comparison of Accuracy's

But its clearly visible that the Hybrid Algorithm is over performing all the other algorithms there is an increase of 3% which is a great increase in the field on machine learning. So, this algorithm can be employed if the accuracy or F1 scores are considered in the project.

#### 4.11 SUMMARY:

Three datasets have been analysed using six different algorithms including the hybrid algorithm. A dataset has picked among them and clearly analysed the properties like precision, recall, F1 score and accuracy of all the algorithms. This research concluded that hybrid algorithm is best compared to any other simple machine learning algorithm.

## **CHAPTER V**

### **SYSTEM DESIGN**

## **5.1 INTRODUCTION:**

Fake news detection is the most important problem to be addressed in the recent years, there is lot of research going on in this field. Because of its serious impacts on the readers. researchers, government and private agencies working together to solve the issue. This paper represents a hybrid approach for fake data detection using the multinomial voting algorithm.

This algorithm was tested with multiple fake news dataset which resulted in an accuracy score of 94 percent which is a benchmark in the machine learning field where the other algorithms are at a range of 82 to 88 percent. The list of algorithms that have been used here is as follows Naïve Bayes, Random Forest, Decision Tree, Support Vector Machine, K Nearest Neighbours.

All these algorithms use training data as the bag of words model which was created using Count Vectorizer. Experimental data has collected from the Kaggle data world. Python is used as a language to verify and validate the results. Tableau is used as a visualization tool. Implementation is carried out using default algorithm values.

## **5.2 OBJECTIVES OF THE DESIGN:**

The objective of this application design is to create a new fake news detection website that's powered with the upgraded machine learning algorithm which can surely outperform the other algorithms which are available. This can be equipped with the user access and data utilization techniques which can store and use data for other valuable use cases.

## **5.3 FACTORS CONSIDERED IN THE DESIGN:**

The following factors are taken into the account while designing the software

- A user-friendly system.
- A time effective system.

- A user understandable system.
- Easy to implement.
- Non typical user interface.
- Easy to handle data storage.
- Should run on any system.

#### **5.4 OUTPUT DESIGN:**

The output of the program consists of a usable website where the user can login and test the news articles they are having as real or fake articles. All these searches will get stored and can be used for other purposes. Every user has their own rights on their account, whereas the admin has the overall control in the website.

#### **5.5 INPUT DESIGN:**

Because all the core processing happens in the background. This application is a pure website which is a valuable proof that the hybrid algorithm can better over the other algorithms.

Pages in the site:

1. Home page
2. Register
3. Login
4. About us
5. Search page
6. Result page
7. History
8. Edit profile
9. Change password
10. Admin view
11. Logout



## 5.6 HOME PAGE:

This is the first page that will be witnessed by any person who is going to open the website. This is non login centric which means anyone can open this no user has extra rights.

Here a new user can create an account, an existing user can log into there account, about us page can be reached from here.

Because this is the first page addressed by the users, it has been named as the home page of the entire website.

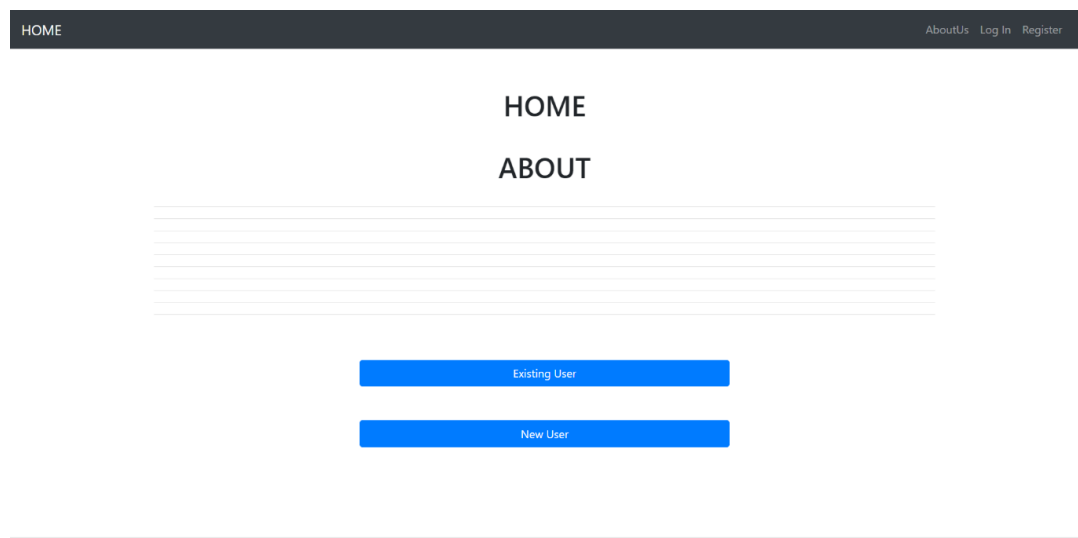


Fig 5.6: Homepage

## 5.7 REGISTER:

This is a very important page for any new user because without having an account with us will keep their actions very limited.

In order to create an account people can navigate to the create new account page and they can address the required details like User name, First name, Last name, Email, and Password to completely the registration process.

The entire process is hashed so no one can see the password the user is providing and lot of password protection rules has been insisted so it's safe and secure.

This is non login centric which means anyone can open this no user has extra rights.

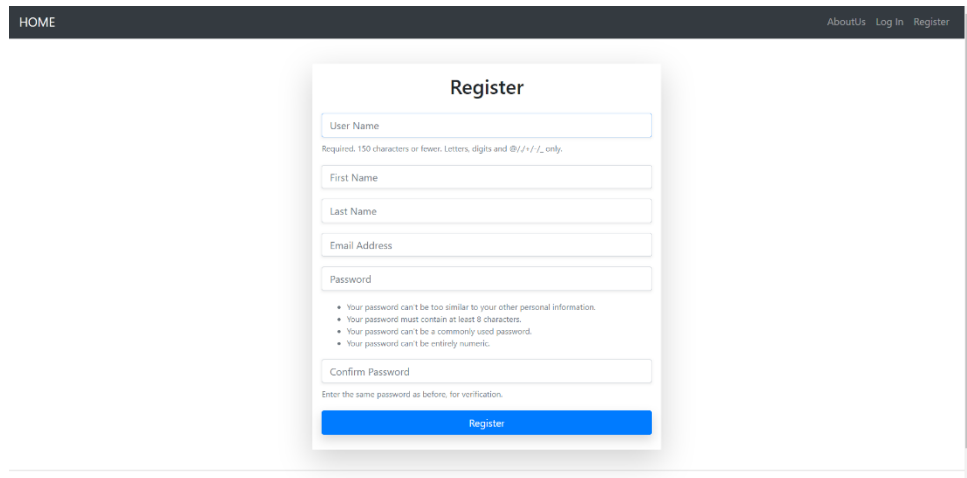
A screenshot of a web application's 'Register' page. The page has a dark grey header with 'HOME' on the left and 'About Us', 'Log In', and 'Register' on the right. The main content area is white and features a central registration form. The form is titled 'Register' and contains several input fields: 'User Name' (with a note: 'Required. 150 characters or fewer. Letters, digits and @/./+/\_ only.'), 'First Name', 'Last Name', 'Email Address', 'Password', and 'Confirm Password'. The 'Password' field has a list of requirements: 'Your password can't be too similar to your other personal information.', 'Your password must contain at least 8 characters.', 'Your password can't be a commonly used password.', and 'Your password can't be entirely numeric.'. A blue 'Register' button is at the bottom of the form.

Fig 5.7: Register

## 5.8 LOGIN PAGE:

Either the user is a normal user or an admin. They can log into there account and access the data using other pages.

Login is a must in the website because if people searches with their complete details it will be very helpful in analysing lot of this like the current main problem, user mental condition, advertisement, save traffic and so on.

This is non login centric which means anyone can open this no user has extra rights.

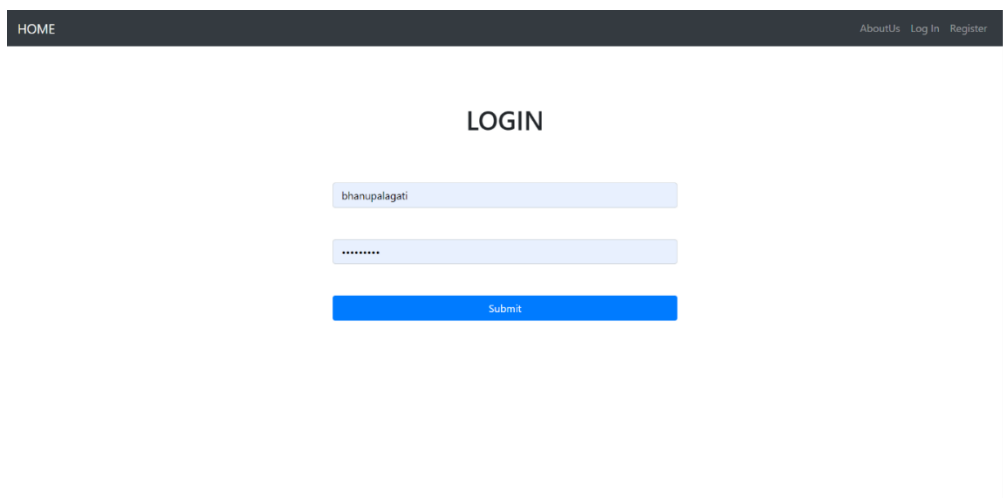
A screenshot of a web application's 'Login' page. The page has a dark grey header with 'HOME' on the left and 'About Us', 'Log In', and 'Register' on the right. The main content area is white and features a central login form. The form is titled 'LOGIN' and contains two input fields: a text field for the username (containing 'bhanupalagati') and a password field (containing '\*\*\*\*\*'). A blue 'Submit' button is at the bottom of the form.

Fig 5.8: Login Page

## 5.9 ABOUT US:

This page holds the entire recipe of how the Ensembled Hybrid Algorithm is created. There are six cards each card telling about an algorithm with the analysed results of the algorithm.

The algorithms are as follows:

- Naïve Bayes classification algorithm with the analysis.
- Support Vector Machine algorithm with the analysis.
- Random Forest algorithm with the analysis.
- Decision Tree algorithm with the analysis.
- K Nearest Neighbours algorithm with the analysis.

This is non login centric which means anyone can open this no user has extra rights.

### Naive Bayes classifier

Naïve Bayes classifier is a simple probabilistic classifier based on the Bayes theorem with great(naive) independence assumption between the data features, where the class labels are drawn from some finite set. It is not a single algorithm to train such classifiers, but a collection of algorithms based on a common principle: every naïve Bayes classifier assumes that the value of a particular feature is independent to the value of any other feature, given the class variable

Naïve Bayes is the most opted statistical technique for the models like email filtering, spam filtering and so on.

Naïve Bayes works on the bag of words features where the data of different articles collected is converted into encoded format by using various vectorization techniques based on the requirement some of them are count vectorizer (CV), term frequency and inverse document frequency vectorizer (TFIDF)

The Bag of words will be passed to the Naïve Bayes model as a training data and based on the data the model will learn

Then when any article is passed to classify vectorizer will create sparse matrix and then model will predict based on the word distribution in the sparse matrix.

Accuracy score achieved by Naïve Bayes algorithm is 0.8966

• The confusion matrix of Naïve Bayes is as follows [526, 87, 44, 610]

### Random Forest confusion matrix

negative

Fig 5.9 Detailed Information

## 5.10 SEARCH PAGE:

This is the heart of the entire website this is the place where the user can place their content from the other sources to test what they have read is fake or real.

Once they had the news placed in the input text box, they can press the predict button to complete the process. This will navigate the user to other page.

This is non login centric which means anyone can open this no user has extra rights.

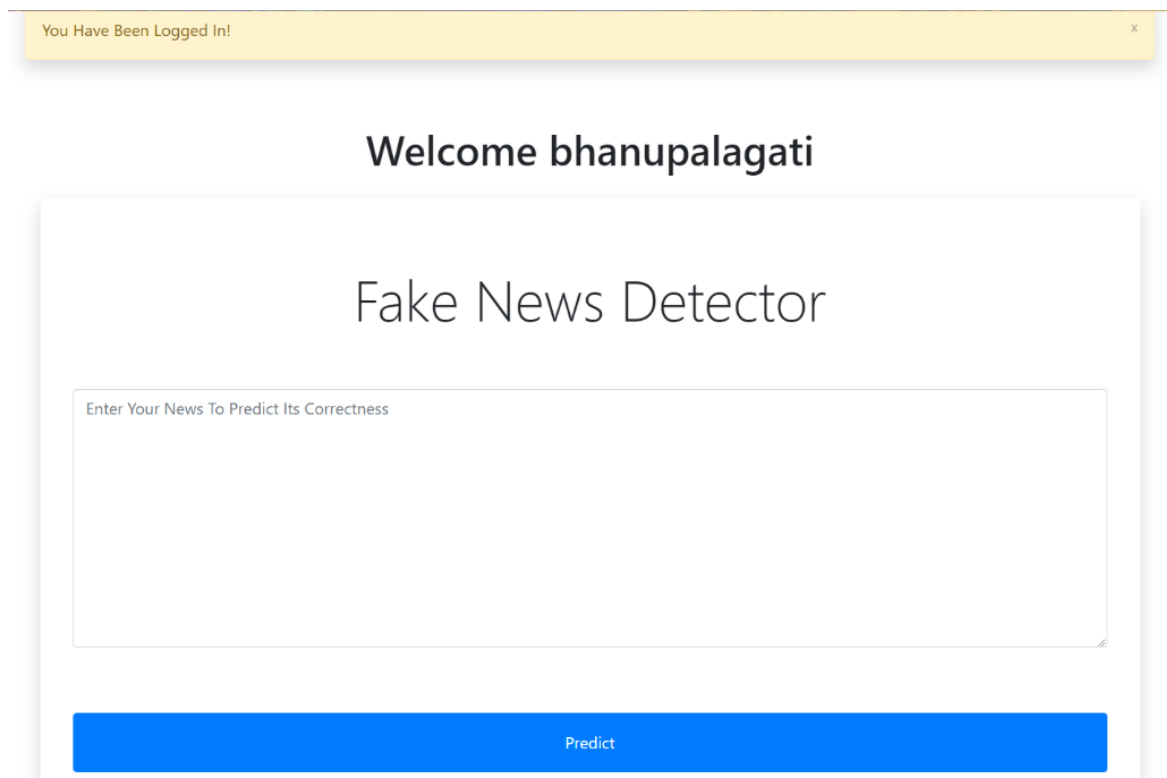


Fig 5.10: Search Page

## 5.11 RESULT PAGE:

When the user clicks on the navigate button the Hybrid algorithm along with all the five individual algorithms will get active. They will use the data provided by the previous stage and process the data and reply with the respective output.

All those algorithms outputs will be stored and used to asses the Hybrid algorithm output.

At the end these results will get navigated to the results page and the page will look like the below.

This is non login centric which means anyone can open this no user has extra rights.

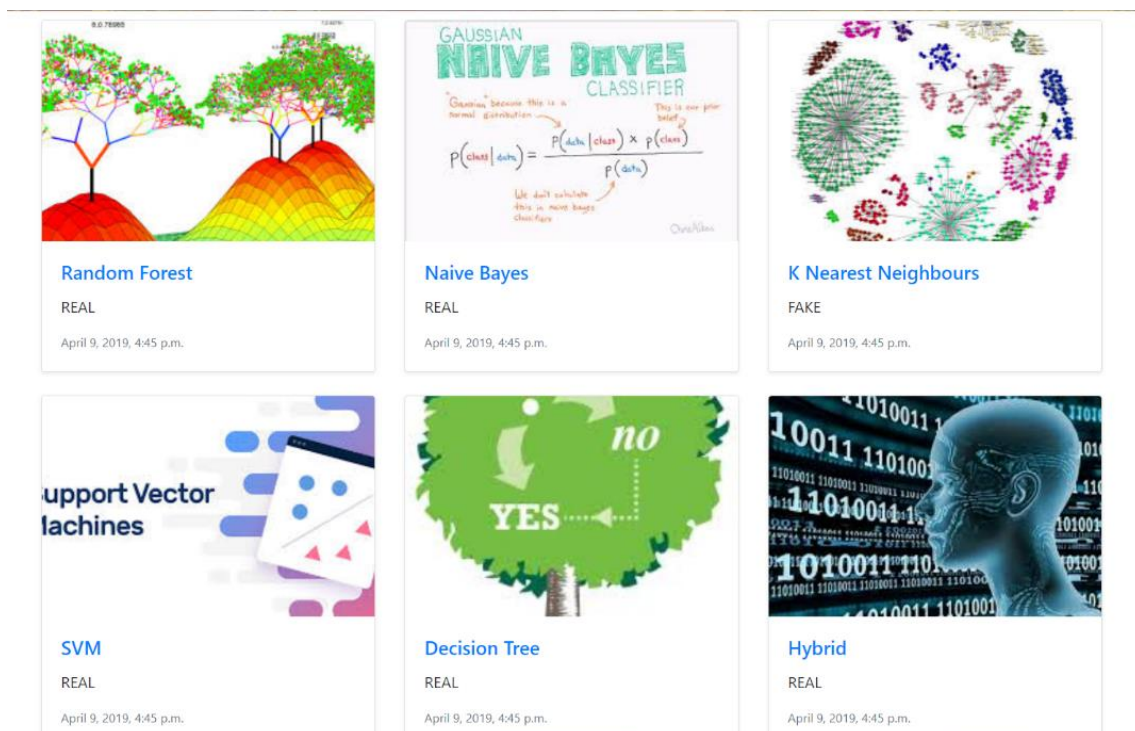


Fig 5.11: Result Page

## 5.12 HISTORY:

The data user searches are the training data for the future improvement. That's why google is providing free google photos so in order to make future improvements this will get stored in database.

Every user can have their own history page, when the user searches it will get stored in the user table. So, they can watch their previous searches with the result they got along with the date and time in this page.

This is login centric page and dynamic based on the current user login.

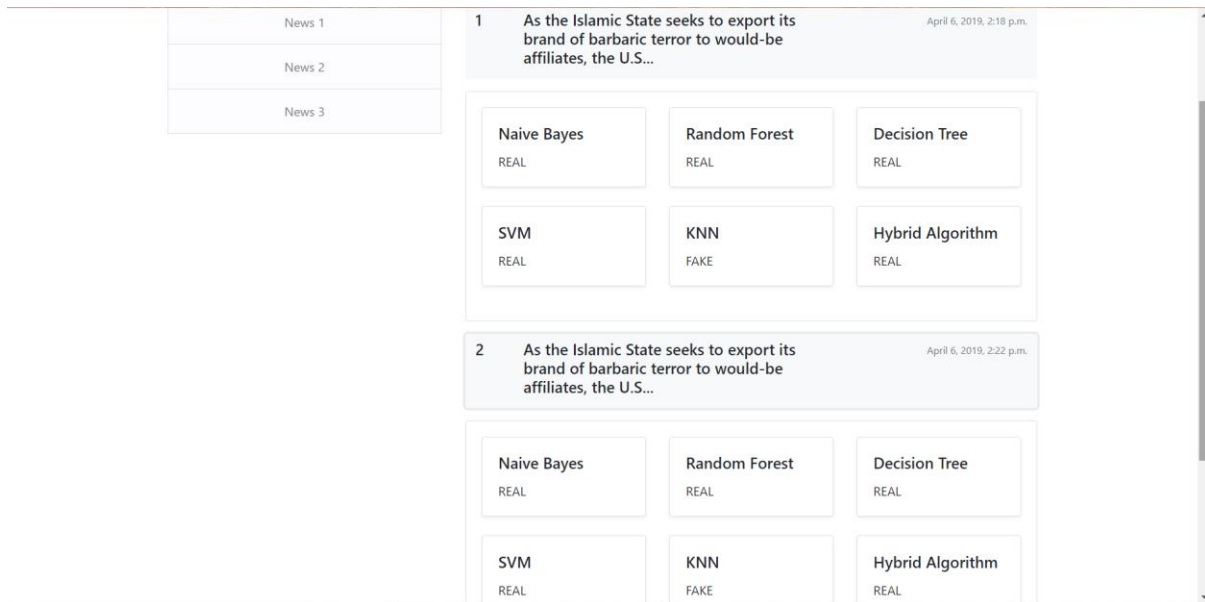


Fig 5.12: History

### 5.13 EDIT PROFILE:

Anyone will make mistakes the designers strongly believe that even they made lot of such mistakes. So, they provided a dedicated edit profile page where the user can correct their information that has been already provided.

The user can change their User name, First name, Last name and email address with the help of this page.

This is login centric page and dynamic based on the current user login.

The screenshot shows a web application with a dark header bar containing 'HOME' on the left and 'bhanupalagati' on the right. In the center, there is a white card titled 'Edit Profile'. The card contains four input fields: 'bhanupalagati' (Username), 'First Name', 'Last Name', and 'bhanupalagati@gmail.com' (Email). Below these fields is a blue 'Save' button. At the bottom of the page, there is a footer with '© Bhanu Palagati 2019' on the left and 'Back to top' on the right.

Fig 5.13: Edit Profile

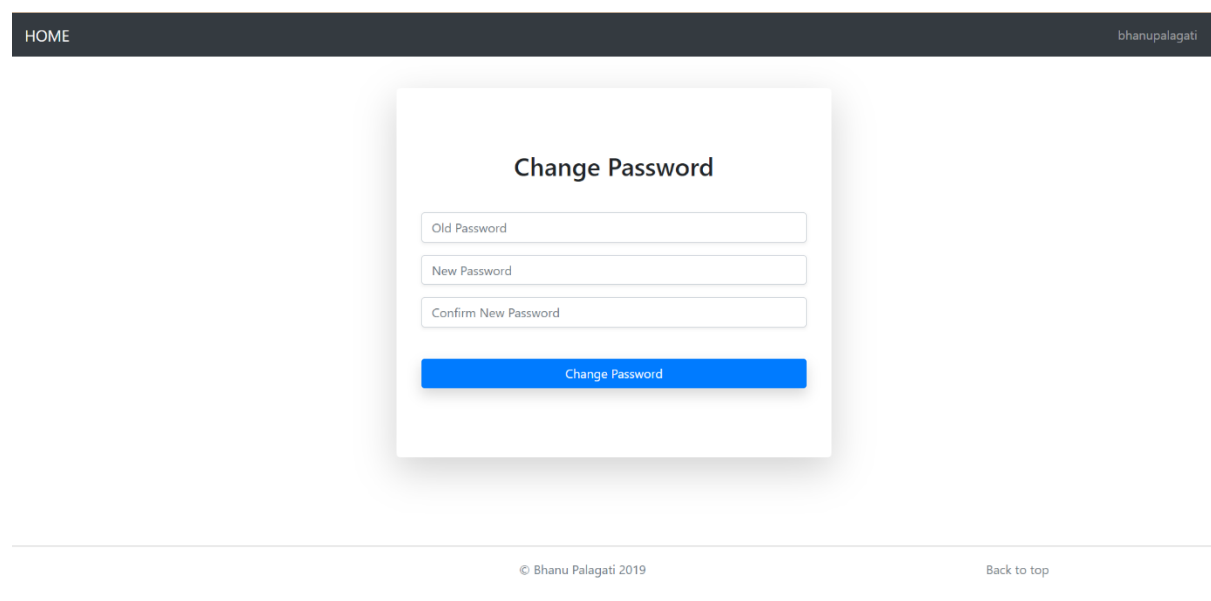
## 5.14 CHANGE PASSWORD:

Everyone want to control their presence in different ways, password changing is one among them. By this page when the user is in their account, they are free to change their password that they have already provided.

This can be done by entering the old password and new password again conforming the new password.

This page can only address the change password, if you forget you can always use the forget password to gain your authority to your account by providing the email address you have used to register with us.

This is login centric page and dynamic based on the current user login.



The screenshot shows a web application interface with a dark header bar containing 'HOME' on the left and 'bhanupalagati' on the right. The main content area features a white card titled 'Change Password'. Inside the card, there are three input fields: 'Old Password', 'New Password', and 'Confirm New Password'. Below these fields is a blue button labeled 'Change Password'. At the bottom of the page, there is a light gray footer bar with '© Bhanu Palagati 2019' on the left and 'Back to top' on the right.

Fig 5.14: Change Password

## 5.15 ADMIN VIEW:

In order to maintain the hierarchy in the website and have control over the other users and to address the queries there should be one admin body.

The users in this admin body are called as the admin users, at the time of creating a website there will be one admin and he/she can promote the other users as admins. They will be having the same rights he is having.

In order to access this page, the person should be an admin user

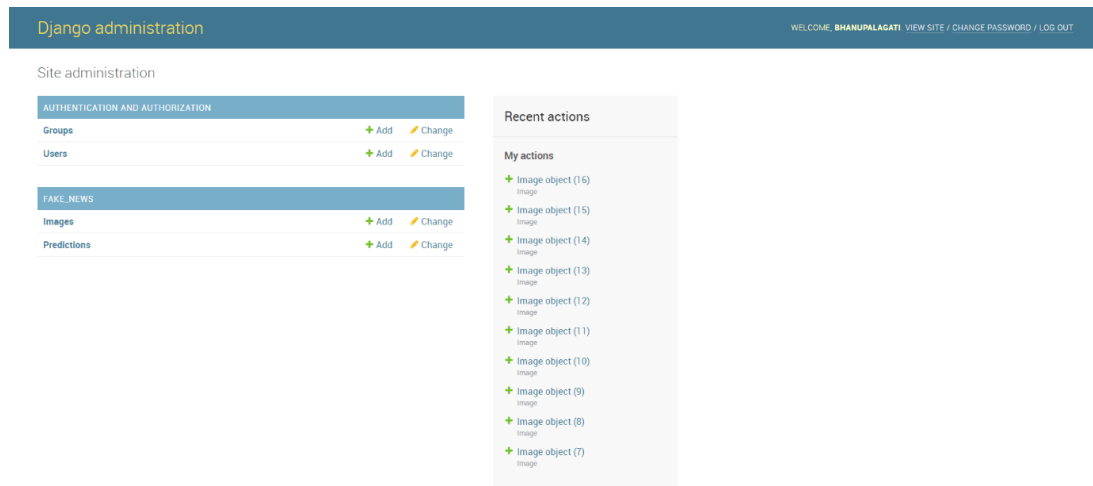


Fig 5.15: Admin View

## 5.16 LOGOUT:

If the user is done with their requests, they can guard their account by logging out of the website.

When the user logs out this webpage will be again navigated to the home page from where the entire process begins again

This is non login centric which means anyone can open this no user has extra rights.

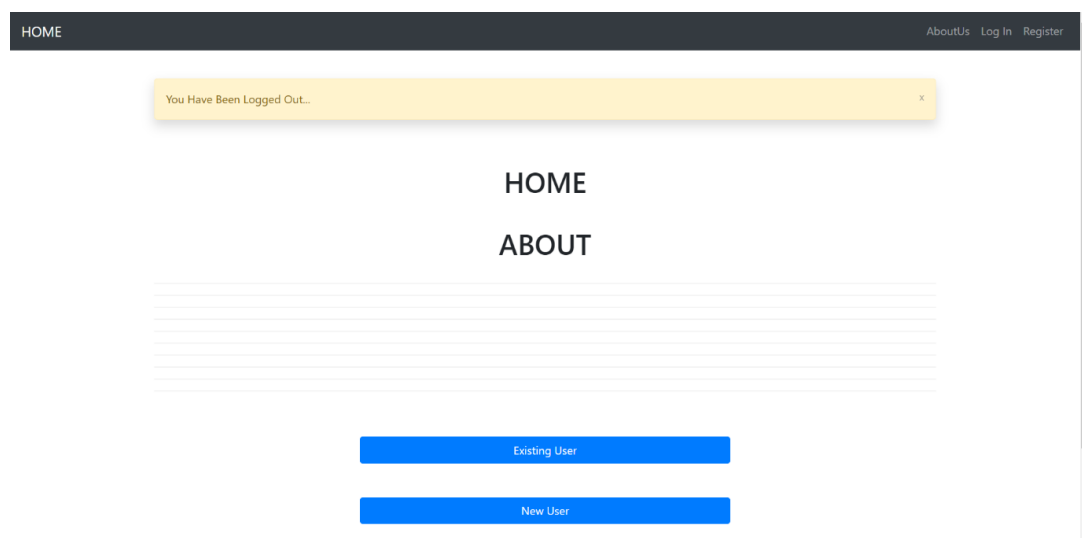


Fig 5.16: Logout



## 5.17 SYSTEM ARCHITECTURE:

The below diagram represents the architecture of the system. The left part of the diagram is solely consigned about the website project which is going to act as the user interface and verification and validation of the Hybrid Algorithm that was constructed.

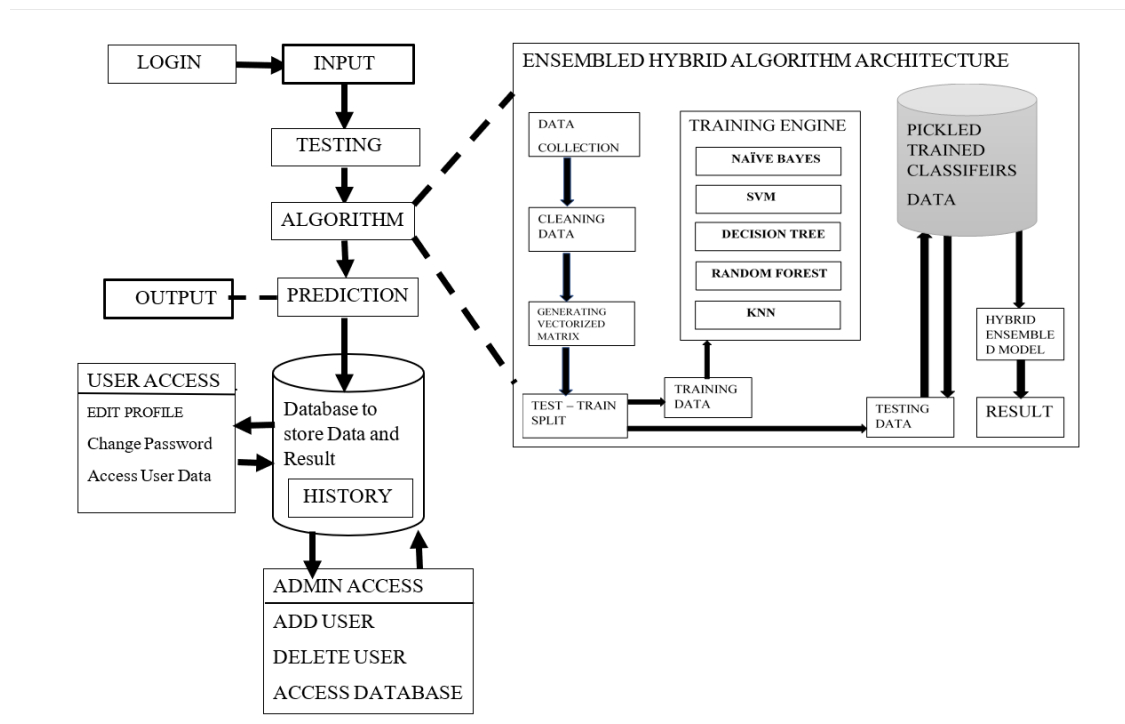


Fig 5.17: System Architecture

Whereas the right side of the diagram represents the algorithm work flow and how the algorithm is constructed and connected with the website that is making use of the website created.

In order to have a clear idea about every module in left side the user can navigate to the project overview section where everything was described in a proper way.

In order to have a clear idea about every module in right side the user can navigate to the Algorithm Analysis section where everything was described in a proper way.

## 5.18 DATABASE CONNECTIVITY:

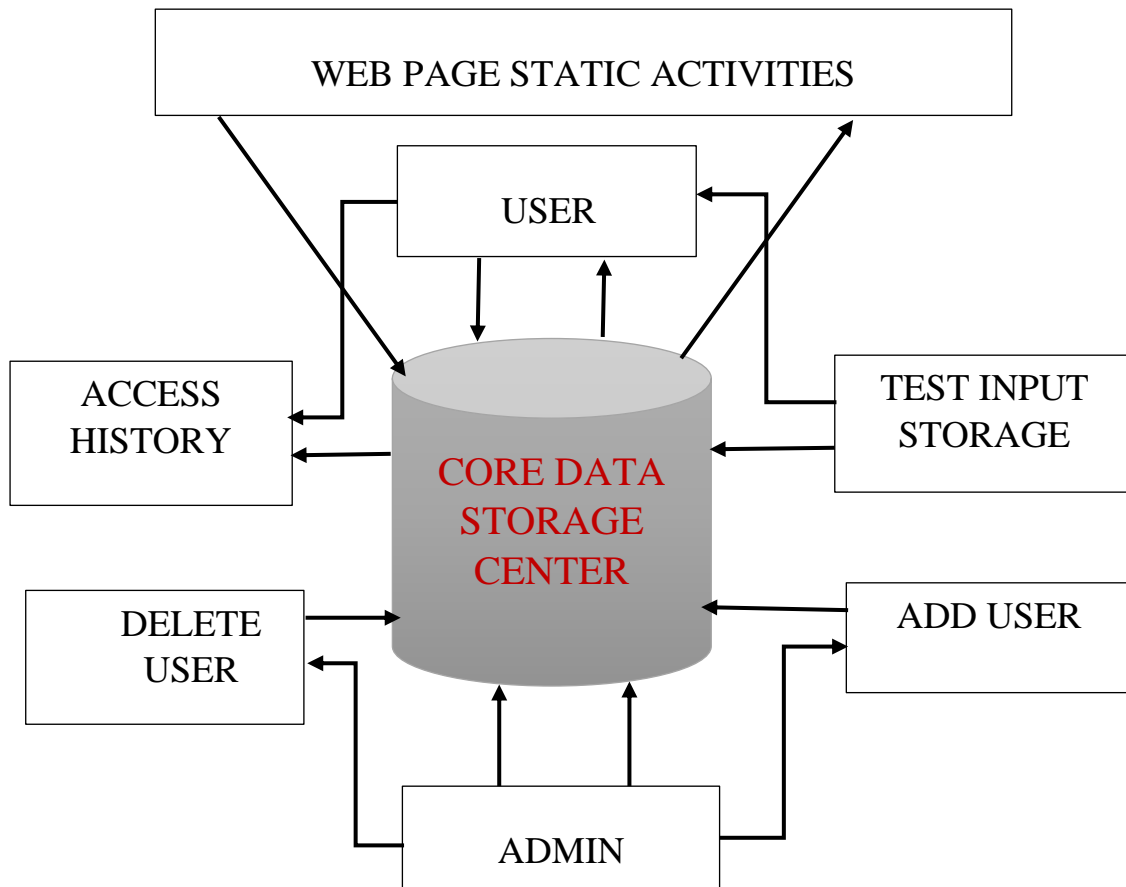


Fig 5.18: Database Connectivity

- A proper website or application can only be maintained with the proper database design.
- Hence database for this project is designed in such a way that everything is mostly independent of the other.
- This database holds the following information
- User data like all the details provided by the user during the registration period and the rights of user like admin or normal user.
- Complete webpage activities like static files, events will get stored here.
- Pickle files for running the algorithms will be hosted in the database.
- The user search history, results, date and time and various other things will get stored in the database.

- The above diagram is the database architecture diagram for the project.

### **5.19 SUMMARY:**

This section is the application design for the algorithm that was constructed to prove that combined algorithms can surely out perform the single algorithms. The application was created using the Django framework, HTML, CSS, and JAVASCRIPT. It explains about each page and the input output data flows with the architecture diagrams.

## **CHAPTER VI**

### **TESTING**

## **6.1 INTRODUCTION:**

First the user will start testing with individual module and will perform unit testing on that. So, user successfully verify and validate all modules by successively integrating each module. More over checking the work done was very important to reduce risk factor.

Checking was being ultimately handled by testing but interim checking was required. So, user planned work done by one member was tested by other for some time and again revolved for other level check. This technique proved to be very much helpful as it came out with innovative ideas to reduce error very low level. The objective of this testing phase is to prove that the developed system satisfies the requirements defined earlier.

Several types of tests will be conducted in this phase. Testing is an important phase of system development because it can ensure the system matches the specifications.

Besides that, testing also ensures that the system functions in the correct and proper manner with the minimum amount of errors. Bottom-up testing strategy is used in this system to avoid unnecessary duplication of effort.

Individual objects will be tested in isolation using unit testing and gradually integrated for the higher-level integration testing and system testing. The components failed will return back to the development phase for rework, and components that are working properly will migrate ahead for implementation.

So, user planned work done by one member was tested by other for some time and again revolved for other level check. This technique proved to be very much helpful as it came out with innovative ideas to reduce error very low level.

The objective of this testing phase is to prove that the developed system satisfies the requirements defined earlier.

## **6.2 TESTING METHODS:**

### **6.2.1 Performance Testing:**

This testing method is used to test the running time of the constructed system. This tests the user performance at the modular level and the system level.

### **6.2.2 Black Box Testing:**

This is an automated testing process, where the program will get passed through a function whose process is not known but the input and output of that black box function is known.

### **6.2.3 Unit Testing:**

Unit testing is used to analyze the semantic and syntactic errors even from the small programming unit. In this system unit testing was used to analyze each site and algorithm.

If there are some errors common in multiple units of the program if these kinds of errors found in one unit can be addressed in the other units as well this reduces the testing time.

Because this testing phase is purely dynamic there is no track of documentation.

### **6.2.4 Selenium Testing:**

Selenium software is an open source application which is used to test the websites in the web browsers. But this can't be used for other software like mobile or pc applications.

This selenium software is used for testing the web app created using the Django framework (python).

Based on the above selenium test results created web app passed all the test cases that was used to test the application.

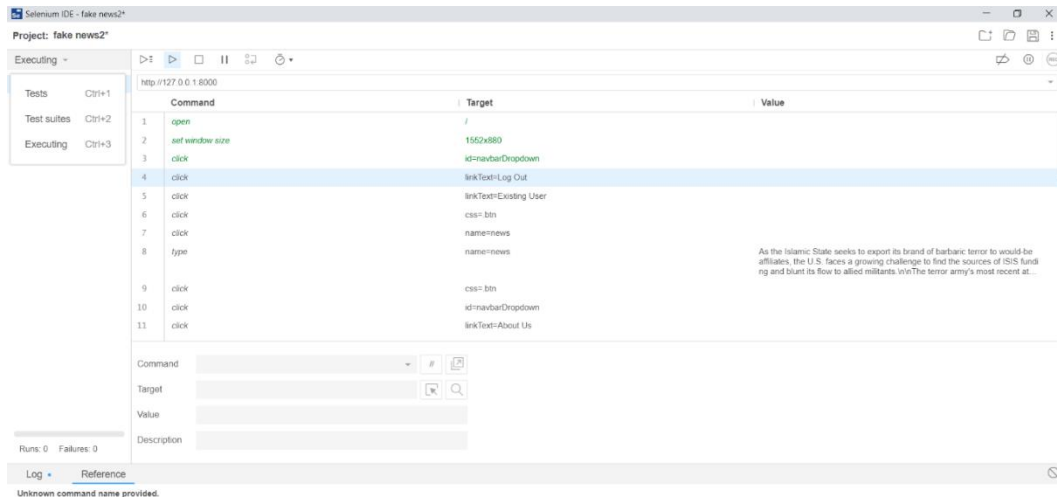


Fig 6.2.4: selenium test result for the web app software.

## 6.2.5 Python Testing:

Python itself is packed with lot of testing classes. These can be used to all the testing operations that has been discussed earlier in this chapter. Unit testing, Integrated Testing, Complexity Testing all these operations can be carried out using this python built in testing solutions.

Some of these packages are as below:

- UNITTEST
- NOSE OR NOSE2
- PYTEST

Any of the above packages can be imported and can be directly used for getting the security and other loops in the project.

## 6.3 SUMMARY:

This testing chapter discusses about all the usable testing methods like Performance Testing, Black Box Testing, Interface Testing, Unit testing on the software, and also the importance of the testing. Mainly these testing methods are used to reduce the risks integrated with the project. Unit testing is used for single module. Interface testing for single interface. Performance testing will analyze about the time and space complexity of the website. Testing is one of the very important elements in the software development because a single loop hole can destroy the entire project and can lead to the down fall of the entire organization.



## **CHAPTER VII**

### **CONCLUSION & FUTURE WORK**

## 7.1 CONCLUSION:

Data or information is the most valuable asset in this data driven world. The most important problem to be solved is to evaluate whether the data is relevant or irrelevant. Fake data has a huge impact on lot of people and organizations that may even lead to the end of the organization or panic the people.

Machine learning researchers believe that this problem can be solved using the machine learning algorithms and there is lot of on-going research in this field which lead to the new branch called Natural Language Processing.

When the classification test is carried out by the humans this led to an accuracy score of 65%. When the same is carried out with a simple machine learning algorithm like Naïve Bayes gave a 75% result. This shows that machine learning algorithm is always better than a human.

This entire research showed that ensembled algorithm can perform better when compared with single algorithm in an aspect. By observing confusion matrix, a conclusion can be drawn that various algorithms are learning in various ways some are making less false negatives and some are making less false positives. If various algorithms are trained and a test data is passed by all the algorithms and mode of their results will be announced as the final result gave huge change in the accuracy scores. There is an increase 2.5 percent at least and 10 percent at most. So, the research concluded that ensembled algorithm is far better than a single algorithm.

The application developed can be used for the fake news detection. There are lot of detection applications like this. But the change is this project is powered

with a combined algorithm which will surely increase the accuracy of the classification and make it perfect.

## **7.2 FUTURE WORK:**

As a future project this ensembled algorithm will get compared with the deep neural networks and test results will be drawn. If this performs better then lot of time can be saved in training the deep neural networks. This can even change usage of machine learning over the datasets.

The data that has been collected from the user searches is stored and get used for increasing the accuracy of the algorithm. As everyone know if the data is higher than the algorithm will get better and better.

Multi core usable programs can be written where multiple algorithms will run simultaneously by which the time required for running these kinds of combined algorithms will run much faster.

Sinking the search data with the real time will identify the most rapidly spreading news so that the admin can identify the news and can act according to the situation. This can resolve human and asset loss.

## **CHAPTER VIII**

### **REFERENCES**

### References:

- [1] Mykhailo Granik, Volodymyr Mesyura, “Fake News Detection Using Naive Bayes Classifier”, IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) (2017).  
<https://ieeexplore.ieee.org/document/8100379>
- [2] Akshay Jain, Amey Kasbe, “Fake News Detection”, IEEE International Students Conference on Electrical, Electronics and Computer Science (SCEECS) (2018).  
<https://ieeexplore.ieee.org/document/8546944>
- [3] Kai Shu, Suhang Wang, Huan Liu, “Understanding User Profiles on Social Media for Fake News Detection”, IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) (2018).  
<https://ieeexplore.ieee.org/document/8397048>
- [4] Hari Krishna M, Rahamathulla K, Ali Akbar, “A Feature Based Approach for Sentiment Analysis using SVM and Coreference Resolution”, International Conference on Inventive Communication and Computational Technologies (ICICCT) (2017).  
<https://ieeexplore.ieee.org/document/7975227>
- [5] Himdweep Walia, Ajay Rana, Vineet Kansal, “A Naïve Bayes Approach for working on Gurmukhi Word Sense Disambiguation”, 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) (2017).  
<https://ieeexplore.ieee.org/document/8342465>
- [6] Yaguang Ji, Songnian Yu, Yafeng Zhang, “A novel Naive Bayes model: Packaged Hidden Naive Bayes”, 6th IEEE Joint International Information Technology and Artificial Intelligence Conference (2011).  
<https://ieeexplore.ieee.org/document/6030379>

- [7] Peng Hong, Lin Chengde, Luo Linkai, Zhou Qifeng, “Accuracy of Classifier Combining Based on Majority Voting”, IEEE International Conference on Control and Automation (2007).  
<https://ieeexplore.ieee.org/document/4376843>
- [8] Andrew Christian Flores, Rogelyn I. Icoy, Christine F Pena, Ken D. Gorro, “An Evaluation of SVM and Naive Bayes with SMOTE on Sentiment Analysis Data Set”, International Conference on Engineering, Applied Sciences, and Technology (ICEAST) (2018).  
<https://ieeexplore.ieee.org/document/8434401>
- [9] Sukirty Jain, Sanyam Shukla, Bhagat singh Raghuwanshi, “Analysis of ordering-based ensemble pruning techniques for Voting based Extreme Learning Machine”, IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (2018).  
<https://ieeexplore.ieee.org/document/8546952>
- [10] Md Rafiqul Islam, Abu Raihan M. Kamal, “Detecting Depression Using K-Nearest Neighbour’s (KNN) Classification Technique”, International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2) (2018).  
<https://ieeexplore.ieee.org/document/8465641>
- [11] Shlok Gilda, “Evaluating Machine Learning Algorithms for Fake News Detection”, IEEE 15th Student Conference on Research and Development (SCOREd) (2017).  
<https://ieeexplore.ieee.org/document/8305411>
- [12] Chandra Mouli Madhav Kotteti, Na Li, Lijun Qian, “Fake News Detection Enhancement with Data Imputation”, IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC3/PiCom/DataCom/CyberSciTech) (2018).  
<https://ieeexplore.ieee.org/document/8511885>

- [13] Saranya Krishnan, Min Chen, “Identifying tweets with Fake News”, IEEE International Conference on Information Reuse and Integration (IRI) (2018).  
<https://ieeexplore.ieee.org/document/8424744>
- [14] Ramadhan WP, Astri Novianty S.T.M.T, Casi Setianingsih S.T.M.T, “Sentiment Analysis Using Multinomial Logistic Regression, International Conference on Control”, Electronics, Renewable Energy and Communications (ICCREC) (2017).  
<https://ieeexplore.ieee.org/document/8226700>
- [15] Yashaswini Hegde, S.K Padma, “Sentiment Analysis using Random Forest Ensemble for Mobile Product Reviews in Kannada”, IEEE 7th International Advance Computing Conference (IACC) (2017).  
<https://ieeexplore.ieee.org/document/7976894>

**CHAPTER IX**  
**SAMPLE CODE**



### Sample code

Since the project is completely based on the software there are nearly 2000 lines of code, which is impossible to accommodate. So please find the DVD providing in order to get the complete code.

Yet another simple way is to find on github <https://github.com/bhanupalagati> page.

This code only provides a minimal information of how the Hybrid algorithm is created.

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Wed Jan 23 19:05:30 2019
```

```
@author: bhanu
```

```
"""
```

```
# importing pandas for dataframe modification
```

```
import pandas as pd
```

```
#importing text extraction vectorizers
```

```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer,  
HashingVectorizer
```

```
#importing the test train splitter for making the test and train split
```

```
from sklearn.cross_validation import train_test_split
```

```
# importing the pickle file
```

```
import pickle
```

```
# import numpy
```

```
import numpy as np
```

```
#Creating a pickle file
```

```
pickle_out = open("trained_classifiers.pickle", "wb")
```

```
# reading a text file from the local host
```

```

#df = pd.read_csv("fake-news/train.csv")
#dftest = pd.read_csv("fake-news/test.csv")
df = pd.read_csv("fake_or_real_news.csv")
# encoding the fake value with 0
df.loc[df["label"]=="FAKE", "label"] = 0
# encoding the real value with 1
df.loc[df["label"]=="REAL", "label"] = 1
# creating new dataframe of dependent variables
y = df.label

X_train, X_test, y_train, y_test = train_test_split(df['text'], y, test_size=0.2,
random_state=53)
pickle.dump(X_train, pickle_out)
# count vectorizer which will convert the words into the mathematical dataset
cv = CountVectorizer(stop_words='english')
x_traincv = cv.fit_transform(X_train)
a = x_traincv.toarray()
#inversed array
cv_inversed = cv.inverse_transform(a[0])
# tfidf vectorizer which will convert the words into the mathematical dataset
td = TfidfVectorizer(stop_words='english', max_df=0.7)
x_traintd = td.fit_transform(X_train)
atd = x_traintd.toarray()
#inversed array
td_inversed = td.inverse_transform(atd[0])
# hv vectorizer which will convert the words into the mathematical dataset
hv = HashingVectorizer(stop_words='english', non_negative=True)
x_trainhv = hv.fit_transform(X_train)
# creating a tfidf transformed x_test

```

```

x_testtd = td.transform(X_test)
# creating a count vectorizer transformed x_test
x_testcv = cv.transform(X_test)
# dumping the x_testcv
pickle.dump(X_test,pickle_out)
pickle.dump(y_test, pickle_out)
# creating a hash vectorizer transformed x_test
x_testhv = hv.transform(X_test)
# converting a string dependent variable to a int
y_train = y_train.astype('int')
from sklearn.naive_bayes import MultinomialNB
# machine learning algorithm for the multinomial Naive Bayes Classifier using
the count vectorizer
mnb2 = MultinomialNB().fit(x_traincv,y_train)
# dumping mnb2
pickle.dump(mnb2, pickle_out)

# machine learning algorithm Random Forest Classifier
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators = 10, random_state =
42).fit(x_traincv,y_train)
# dumping random forest
pickle.dump(rf, pickle_out)

# machine learning algorithm Decision Tree
from sklearn.tree import DecisionTreeClassifier
decision = DecisionTreeClassifier().fit(x_traincv, y_train)
pickle.dump(decision, pickle_out)

```

```

# machine learning algorithm support vector machine
from sklearn import svm
clf = svm.SVC(gamma=0.001).fit(x_traincv, y_train)

# dumping svm
pickle.dump(clf, pickle_out)

# machine learning algorithm kneighbours
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=3).fit(x_traincv, y_train)

# dumping knn
pickle.dump(neigh, pickle_out)

predmnbcv = mnbc2.predict(x_testcv)
predrfcv = rf.predict(x_testcv)
preddecisioncv = decision.predict(x_testcv)
predlgcv = logistic.predict(x_testcv)
predsvmcv = clf.predict(x_testcv)
predneighcv = neigh.predict(x_testcv)

# creating a compound machine learning algorithm storing in the final
final = []
for i in range(1267):
    c = 0
    c=
    predmnbcv[i]+predrfcv[i]+preddecisioncv[i]+predsvmcv[i]+predneighcv[i]
# based on the vote we will opt for the most
    if c>=3:

```

```
    final.append(1)
else:
    final.append(0)
```