**Fake News Detection**

**Abstract:** Most of the smart phone users prefer to read the news via social media over internet. The news websites are publishing the news and provide the source of authentication. The question is how to authenticate the news and articles which are circulated among social media like WhatsApp groups, Facebook Pages,

Twitter and other micro blogs & social networking sites. It is harmful for the society to believe on the rumors and pretend to be a news. The need of an hour is to stop the rumors especially in the developing countries like India, and focus on the correct, authenticated news articles. This paper demonstrates a model

and the methodology for fake news detection. With the help of Machine learning and natural language processing, it is tried to aggregate the news and later determine whether the news is real or fake using Logistic regression. The proposed model is working well and defining correctness of the results 97.21 % of accuracy.

Keywords: Accuracy level, Analysing, Logistic regression

**1.Introduction**

In the world of rapidly increasing technology information sharing has become an easy task. There is no doubt that internet has made our lives easier and access to lots of information. This is an evolution in human history, but at the same time it un-focusses the line between true media and maliciously forged media. Today anyone can publish content – credible or not – that can be consumed by the world wide web. Sadly, fake news accumulates a great deal of attention over the internet, especially on social media. People get deceived and don’t think twice before circulating such mis-informative pieces to the world. This kind of news vanishes but not without doing the harm it intended to cause. The social media sites like Facebook, Twitter, Whatsapp play a major role in supplying these false news. Many scientists believe that counterfeited news issue may be addressed by means of machine learning and artificial intelligence. Fake news detection is made to stop the rumors that are being spread through the various platforms whether it be social media or messaging platforms, this is done to stop spreading fake news which leads to activities like mob lynching, this has been a great reason motivating us to work on this project. We have been continuously seeing various news of mob lynching that leads to the murder of an individual; fake news detection works on the objective of detecting this fake news and stopping activities like this thereby protecting the society from these unwanted acts of violence. The most common algorithms used by fake news detection systems include machine learning algorithms such as Support Vector Machines, Random Forests, Decision trees,

Stochastic Gradient Descent, Logistic Regression and so on. In this project have attempted to implement one out of these algorithms to train and test our results to attain the maximum accuracy using logistic algorithm.

**1.1 Aim and Objective**

**Aim:**

To predict the Fake News using Dataset.

**Objective:**

We are developing a ML project to predict the Fake News.

**1.2Existing System**

Hence it is crucial to determine the integrity of the information available on the internet in order to keep the prevalence of fake news in check.

So,we’ll now try to build a simple Machine Learning Model using Logistic Regression to detect whether a news is fake or not.

**1.3Problem Statement**

Fake News detection.

**1.4Proposed System/Solution**

In the proposed system we are developing a machine learning project. We are detecting the Fake News. Using Logistic Regression we are predicting whether the News is Real or Fake.

**2.Literature survey.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL**  **NO** | **Reference paper** | **Literature review** | **Merits** | **Demerits** |
| 1. | Models for Accurate Detection of Fake News  Author:  Umadevi K S  Karishnu Poddar  Geraldine Bessie Amali D  Year  2012 | Abstract—Fake news consists of news that is not well researched or deliberate steps have been taken to spread misinformation or hoaxes via different forms of news distribution networks. This paper aims to tackle this issue using a computational model of probabilistic and geometric machine learning models. Moreover, the scores of two different vectorizers namely count and Term Frequency Inverse Document Format(TF-IDF) will be compared to find the appropriate vectorizer for fake news detection. English stop words have been used to improve the scores. Various classifiers like Naive Bayes, Support Vector Machine(SVM), Logistic regression and decision tree classifier were used to predict the fake news. Simulation results indicate Support Vector Machine (SVM) with the TF-IDF gave the most accurate prediction. | This models helps to finding a all types of fake news | The decision  on  the **model for accurate detection of fake news** is often  critical, as each **metric**  may  favor a different algorithm. |
| 2. | Scrutinizing of Fake News using Machine Learning Techniques Author :  [S. Gowri](https://ieeexplore.ieee.org/author/37085694734); [J Jenila](https://ieeexplore.ieee.org/author/37088839205); [Bathula Sowmya Reddy](https://ieeexplore.ieee.org/author/37088840519); [M.Antony Sheela](https://ieeexplore.ieee.org/author/37088837852)  Year’  2000 | Fake news usually comprises of false or misleading information that has the aim of damaging the personality of people. Furthermore, Fake news contains the information that has not been properly researched and proved information, wherein it spreads false alarm via different social media applications. In order to overcome this significant bottleneck, a computational model is proposed here with a likelihood machine learning technique that detects the fake news. TF-IDF vectorizer on the proposed dataset uses the classification algorithm like SGD (Stochastic Gradient Descent) Classifier for achieving a better efficiency etc. The TFIDF (Term Frequency Inverse Document Frequency) vectorizer gives the classification accuracy of the proposed model and converts a raw document into a confusion matrix of TF-IDF vectorizer. | Due to the quantity of resampling, we can average the different test metrics obtained, beating the effect that random chance has when we only have one test set. | **Permutation**  tests rely on  **resampling**   the original data assuming the  null hypothesis. Based on the resampled data  it can be concluded how likely the original data is to occur under the null hypothesis. |
| 3. | Automatic Online Fake News Detection Combining Content and Social SignalsAuthor:Marco L. Della Vedova and Eugenio Tacchini Ballarin,Year1997 | The proliferation and rapid diffusion of fake news on the Internet highlight the need of automatic hoax detection systems. In the context of social networks, machine learning (ML) methods can be used for this purpose. Fake news detection strategies are traditionally either based on content analysis (i.e. analyzing the content of the news) or - more recently - on social context models, such as mapping the news' diffusion pattern. In this paper, we first propose a novel ML fake news detection method which, by combining news content and social context features, outperforms existing methods in the literature, increasing their already high accuracy by up to 4.8%. Second, we implement our method within a Facebook Messenger chatbot and validate it with a real-world application, obtaining a fake news detection accuracy of 81.7%. | Automaticomline  Fake newes detectipn can be thought of as a more sophisticated version of oversampling or a specific data augmentation algorithm. | Over  generalizatioprocedure is inherently dangerous  since it blindly generalizes the minority area without regard  to the majority class. |

**Python Machine Learning:**

**Machine Learning (ML)** is an automated learning with little or no human intervention. It involves programming computers so that they learn from the available inputs. The main purpose of machine learning is to explore and construct algorithms that can learn from the previous data and make predictions on new input data.

The **input** to a learning algorithm is training data, representing experience, and the **output** is any expertise, which usually takes the form of another algorithm that can perform a task. The input data to a machine learning system can be numerical, textual, audio, visual, or multimedia. The corresponding output data of the system can be a floating-point number, for instance, the velocity of a rocket, an integer representing a category or a class, for example, a pigeon or a sunflower from image recognition.

In this chapter, we will learn about the training data our programs will access and how learning process is automated and how the success and performance of such machine learning algorithms is evaluated.

**Concepts of Learning**

Learning is the process of converting experience into expertise or knowledge.

Learning can be broadly classified into three categories, as mentioned below, based on the nature of the learning data and interaction between the learner and the environment.

* Supervised Learning.
* Unsupervised Learning.
* Semi-supervised learning.

Similarly, there are four categories of machine learning algorithms as shown below:

* Supervised learning algorithm.
* Unsupervised learning algorithm.
* Semi-supervised learning algorithm.
* Reinforcement learning algorithm.

However, the most commonly used ones are **supervised** and **unsupervised learning**.

**Supervised Learning**

Supervised learning is commonly used in real world applications, such as face and speech recognition, products or movie recommendations, and sales forecasting. Supervised learning can be further classified into two types: **Regression** and **Classification**.

**Regression** trains on and predicts a continuous-valued response, for example predicting real estate prices.

**Python Machine Learning – Types of Learning**

**Classification** attempts to find the appropriate class label, such as analyzing positive/negative sentiment, male and female persons, benign and malignant tumors, secure and unsecure loans etc.

In supervised learning, learning data comes with description, labels, targets or desired outputs and the objective is to find a general rule that maps inputs to outputs. This kind of learning data is called **labeled data**. The learned rule is then used to label new data with unknown outputs.

Supervised learning involves building a machine learning model that is based on **labeled samples**. For example, if we build a system to estimate the price of a plot of land or a house based on various features, such as size, location, and so on, we first need to create a database and label it. We need to teach the algorithm what features correspond to what prices. Based on this data, the algorithm will learn how to calculate the price of real estate using the values of the input features.

Supervised learning deals with learning a function from available training data. Here, a learning algorithm analyses the training data and produces a derived function that can be used for mapping new examples. There are many **supervised learning algorithms** such as Logistic Regression, Neural networks, Support Vector Machines (SVMs), and Naive Bayes classifiers.

Common **examples** of supervised learning include classifying e-mails into spam and not-spam categories, labeling webpages based on their content, and voice recognition.

**Unsupervised Learning**

Unsupervised learning is used to detect anomalies, outliers, such as fraud or defective equipment, or to group customers with similar behaviors for a sales campaign. It is the opposite of supervised learning. There is no labeled data here.

When learning data contains only some indications without any description or labels, it is up to the coder or to the algorithm to find the structure of the underlying data, to discover hidden patterns, or to determine how to describe the data. This kind of learning data is called **unlabeled data**.

Suppose that we have a number of data points, and we want to classify them into several groups. We may not exactly know what the criteria of classification would be. So, an unsupervised learning algorithm tries to classify the given dataset into a certain number of groups in an optimum way. Unsupervised learning algorithms are extremely powerful tools for analyzing data and for identifying patterns and trends. They are most commonly used for clustering similar input into logical groups. Unsupervised learning algorithms include K-means, Random Forests, Hierarchical clustering and so on. If some learning samples are labeled, but some other are not labeled, then it is semi-supervised learning. It makes use of a large amount of **unlabeled data for training** and a small amount of **labeled data for testing**. Semi-supervised learning is applied in cases where it is expensive to acquire a fully labeled dataset while more practical to label a small subset. For example, it often requires skilled experts to label certain remote sensing images, and lots of field experiments to locate oil at a particular location, while acquiring unlabeled data is relatively easy.

**Reinforcement Learning**

Here learning data gives feedback so that the system adjusts to dynamic conditions in order to achieve a certain objective. The system evaluates its performance based on the feedback responses and reacts accordingly. The best known of instances include self-driving cars and chess master algorithm AlphaGo.

**Purpose of Machine Learning**

Machine learning can be seen as a branch of AI or Artificial Intelligence, since, the ability to change experience into expertise or to detect patterns in complex data is a mark of human or animal intelligence.

As a field of science, machine learning shares common concepts with other disciplines such as statistics, information theory, game theory, and optimization.

As a subfield of information technology, its objective is to program machines so that they will learn.

Python Machine Learning – Environment Setup

Similarly, we can download and install necessary libraries like numpy, matplotlib etc. individually using installers like pip. For this purpose, you can use the commands shown here:

pip install regex

pip install numpy

pip install matplotlib

pip install pandas

pip install seaborn

pip install wordcloud

pip install nltk

pip install Tqdm

pip install scikit-learn

**Python libraries**

A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer. As we don’t need to write the same code again and again for different programs. Python libraries play a very vital role in fields of Machine Learning, Data Science, Data Visualization, etc.

**Working of Python Library**

As is stated above, a Python library is simply a collection of codes or modules of codes that we can use in a program for specific operations. We use libraries so that we don’t need to write the code again in our program that is already available. But how it works. Actually, in the MS Windows environment, the library files have a DLL extension (Dynamic Load Libraries). When we link a library with our program and run that program, the linker automatically searches for that library. It extracts the functionalities of that library and interprets the program accordingly. That’s how we use the methods of a library in our program. We will see further, how we bring in the libraries in our Python programs.

**Python standard library**

The Python Standard Library contains the exact syntax, semantics, and tokens of Python. It contains built-in modules that provide access to basic system functionality like I/O and some other core modules. Most of the Python Libraries are written in the C programming language. The Python standard library consists of more than 200 core modules. All this work together to make Python a high-level programming language. Python Standard Library plays a very important role. Without it, the programmers can’t have access to the functionalities of Python. But other than this, there are several other libraries in Python that make a programmer’s life easier. Let’s have a look at some of the commonly used libraries:

1. **Re (regular expression):** Re library in python holds the key to deal with all the problems relating to**textual data analysis**. This library provides a range of methods that can help you build patterns and extract or substitute the desired string.
2. **Matplotlib:**This library is responsible for plotting numerical data. And that’s why it is used in data analysis. It is also an open-source library and plots high-defined figures like pie charts, histograms, scatterplots, graphs, etc.
3. **Pandas:**Pandas are an important library for data scientists. It is an open-source machine learning library that provides flexible high-level data structures and a variety of analysis tools. It eases data analysis, data manipulation, and cleaning of data. Pandas support operations like Sorting, Re-indexing, Iteration, Concatenation, Conversion of data, Visualizations, Aggregations, etc.
4. **Numpy:**The name “Numpy” stands for “Numerical Python”. It is the commonly used library. It is a popular machine learning library that supports large matrices and multi-dimensional data. It consists of in-built mathematical functions for easy computations. Even libraries like TensorFlow use Numpy internally to perform several operations on tensors. Array Interface is one of the key features of this library.
5. **Seaborn:** Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.
6. **Wordcloud:** Wordcloud is a visualization technique to represent the frequency of words in a text where the size of the word represents its frequency.
7. **NLTK:** NLTK stands for Natural Language Toolkit, is a Python package that you can use for NLP. A lot of the data that you could be analysing is unstructured data and contains human-readable text.
8. **Tqdm library:** The tqdm module allows for the generation of progress bars in Python. The name is derived from the Arabic word, “taqaddum,” which translates as “progress.” It is designed to have minimal overhead, using algorithms to predict the remaining time and to skip unnecessary iteration displays.
9. **Scikit-learn:** Scikit-learn is an opensource data analysis library, and the gold standard for Machine Learning (ML) in the Python ecosystem.

**Remove stopwords**

Stopwords are english words that do not add much meaning to sentence. They can be safely removed without sacrificing the meaning of the sentence for instance, the words like the, he, have, etc. If we notice, stopwords are some of the most frequently appearing words in any paragraph and do not contribute much meaning to sentences.

**Tokenization**

Tokenization is breaking down the sentence into words and paragraphs into sentences. These broken pieces are called tokens, which help understand the context and create a vocabulary. It works by separating the words by spaces or punctuations

**Stemming**

Stemming works by slicing the end of the word using a list of common prefixes and suffixes like (-ing, -ed, -es). this slicing can be successful on most occasions, but not always.

**Vectorization**

Vectorization is the process of transforming a scalar operation acting on individual data elements (Single Instruction Single Data—SISD) to an operation where a single instruction operates concurrently on multiple data elements (SIMD).

Vectorization offers potential speedups in codes with significant array-based computations—speedups that amplify the improved performance obtained through higher-level, parallel computations using threads and distributed execution on clusters.

**TF-IDF vectorization**

TF-IDF vectorization involves calculating the TF-IDF score for every word in your corpus relative to that document and then putting that information into a vector (see image below using example documents “A” and “B”).

TF-IDF is one of the most popular text vectorizers, the calculation is very simple and easy to understand. It gives the rare term high weight and gives the common term low weight.

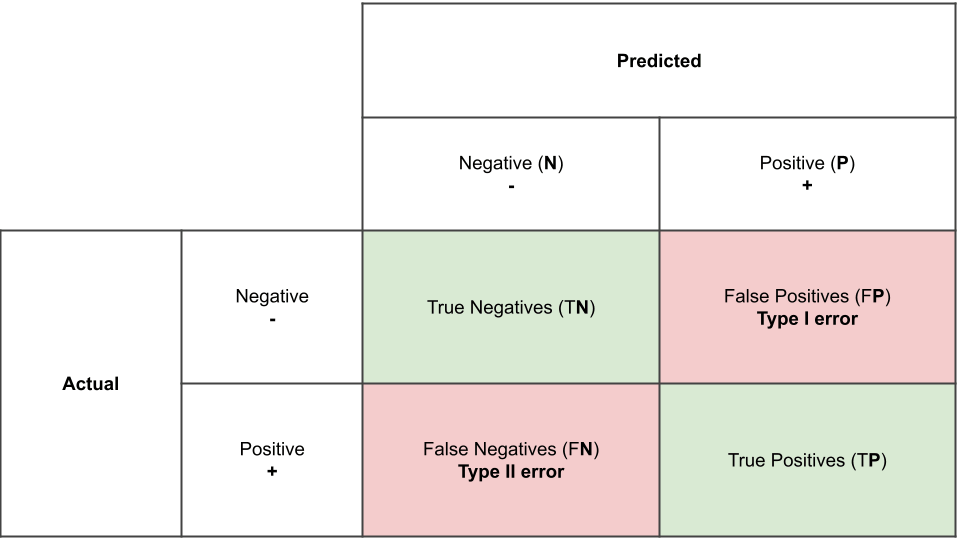
**Data evaluation:**

**Accuracy score**

An Accuracy score (or simply Accuracy) is a Classification measure in Machine Learning that represents a percentage of correct predictions made by a model. Due to its simplicity in calculation and interpretation, the measure has found widespread use. Additionally, the performance of the model is quantified by a single number.

**Confusion matrix**

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model.



**Pickle:**

Pickle in Python is primarily used in serializing and deserializing a Python object structure. In other words, it's the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network.

“Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy.

**3.Training**

Model training is at the heart of the [data science development lifecycle](https://blog.dominodatalab.com/adopting-the-4-step-data-science-lifecycle-for-data-science-projects) where the data science team works to fit the best weights and biases to an algorithm to minimize the loss function over prediction range. Loss functions define how to optimize the ML algorithms. A data science team may use different types of loss functions depending on the project objectives, the type of data used and the type of algorithm.

When a supervised learning technique is used, model training creates a mathematical representation of the relationship between the data features and a target label. In unsupervised learning, it creates a mathematical representation among the data features themselves.

### Importance of Model Training

Model training is the primary step in machine learning, resulting in a working model that can then be validated, tested and deployed. The model’s performance during training will eventually determine how well it will work when it is eventually put into an application for the end-users.

Both the quality of the training data and the choice of the algorithm are central to the model training phase. In most cases, training data is split into two sets for training and then validation and testing.

The selection of the algorithm is primarily determined by the end-use case. However, there are always additional factors that need to be considered, such as algorithm-model complexity, performance, interpretability, computer resource requirements, and speed. Balancing out these various requirements can make selecting algorithms an involved and complicated process.

## **How To Train a Machine Learning Model**

Training a model requires a systematic, repeatable process that maximizes your utilization of your available training data and the time of your data science team. Before you begin the training phase, you need to first determine your problem statement, access your data set and clean the data to be presented to the model.

In addition to this, you need to determine which algorithms you will use and what parameters (hyperparameters) they will run with. With all of this done, you can split your dataset into a training set and a testing set, then prepare your model algorithms for training.

### Split the Dataset

Your initial training data is a limited resource that needs to be allocated carefully. Some of it can be used to train your model, and some of it can be used to test your model – but you can’t use the same data for each step. You can’t properly test a model unless you have given it a new data set that it hasn’t encountered before. Splitting the training data into two or more sets allows you to train and then validate the model using a single source of data. This allows you to see if the model is overfit, meaning that it performs well with the training data but poorly with the test data.

A common way of splitting the training data is to use cross-validation. In [10-fold cross-validation](https://www.dominodatalab.com/blog/guide-to-building-models-with-cross-validation), for example, the data is split into ten sets, allowing you to train and test the data ten times. To do this:

1. Split the data into ten equal parts or folds.
2. Designate onefold as the hold-out fold.
3. Train the model on the other nine folds.
4. Test the model on the hold-out fold.

Repeat this process ten times, each time selecting a different fold to be the hold-out fold. The average performance across the ten hold-out folds is your performance **estimate, called the cross-validated score.**

### Select Algorithms to Test

In machine learning, there are thousands of algorithms to choose from, and there is no sure way to determine which will be the best for any specific model. In most cases, you will likely try dozens, if not hundreds, of algorithms in order to find the one that results in an accurate working model. Selecting candidate algorithms will often depend on:

* Size of the training data.
* Accuracy and interpretability of the required output.
* Speed of training time required, which is inversely proportional to accuracy.
* Linearity of the training data.
* Number of features in the data set.

### Tune the Hyperparameters

Hyperparameters are the high-level attributes set by the data science team before the model is assembled and trained. While many attributes can be learned from the training data, they cannot learn their own hyperparameters.

As an example, if you are using a [regression algorithm](http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/153-penalized-regression-essentials-ridge-lasso-elastic-net/), the model can determine the regression coefficients itself by analysing the data. However, it cannot dictate the strength of the penalty it should use to regularize an overabundance of variables. As another example, a model using the random forest technique can determine where decision trees will be split, but the number of trees to be used needs to be tuned beforehand.

### Fit and Tune Models

Now that the data is prepared and the model’s hyperparameters have been determined, it’s time to start training the models. The process is essentially to loop through the different algorithms using each set of hyperparameter values you’ve decided to explore. To do this:

1. Split the data.
2. Select an algorithm.
3. Tune the hyperparameter values.
4. Train the model.
5. Select another algorithm and repeat steps 3 and 4.

Next, select another set of hyperparameter values you want to try for the same algorithm, cross-validate it again and calculate the new score. Once you have tried each hyperparameter value, you can repeat these same steps for additional algorithms.

Think of these trials as track and field heats. Each algorithm has demonstrated what it can do with the different hyperparameter values. Now you can select the best version from each algorithm and send them on to the final competition.

### Choose the Best Model

Now it’s time to test the best versions of each algorithm to determine which gives you the best model overall.

1. Make predictions on your test data.
2. Determine the ground truth for your target variable during the training of that model.
3. Determine the performance metrics from your predictions and the ground truth target variable.
4. Run each finalist model with the test data.

Once the testing is done, you can compare their performance to determine which are the better models. The overall winner should have performed well (if not the best) in training as well as in testing. It should also perform well on your other performance metrics (like speed and [empirical loss](https://developers.google.com/machine-learning/crash-course/descending-into-ml/training-and-loss)), and – ultimately – it should adequately solve or answer the question posed in your problem statement.

## **Systematic Approach to Model Training**

Using a systematic and repeatable model training process is of paramount importance for any organization planning to build successful [machine learning model](https://www.dominodatalab.com/blog/a-guide-to-machine-learning-models) at scale. Central to this is having all of your resources, [tools](https://www.dominodatalab.com/blog/data-science-tools), libraries and documentation in a single enterprise platform that will foster collaboration instead of hindering it.

**4.Algorithm**

**Regression**

Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables. More specifically, Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. It predicts continuous/real values such as **temperature, age, salary, price,** etc.

**Logistic Regression**

* Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
* Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
* Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
* In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
* The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
* Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
* Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.



**Advantage**

* Logistic regression is easier to implement, interpret, and very efficient to train.
* It makes no assumptions about distributions of classes in feature space.
* It can easily extend to multiple classes (multinomial regression) and a natural probabilistic view of class predictions.
* It is very fast at classifying unknown records.
* Good accuracy for many simple data sets and it performs well when the dataset is linearly separable
* It not only provides a measure of how appropriate a predictor (coefficient size) is, but also its direction of association (positive or negative)

**Disadvantages**

* If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting.
* It constructs linear boundaries.
* The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables.
* It can only be used to predict discrete functions. Hence, the dependent variable of Logistic Regression is bound to the discrete number set.
* Non-linear problems can’t be solved with logistic regression because it has a linear decision surface. Linearly separable data is rarely found in real-world scenarios.
* Logistic Regression requires average or no multicollinearity between independent variables.
* It is tough to obtain complex relationships using logistic regression. More powerful and compact algorithms such as Neural Networks can easily outperform this algorithm.

## **Type of Logistic Regression:**

On the basis of the categories, Logistic Regression can be classified into three types:

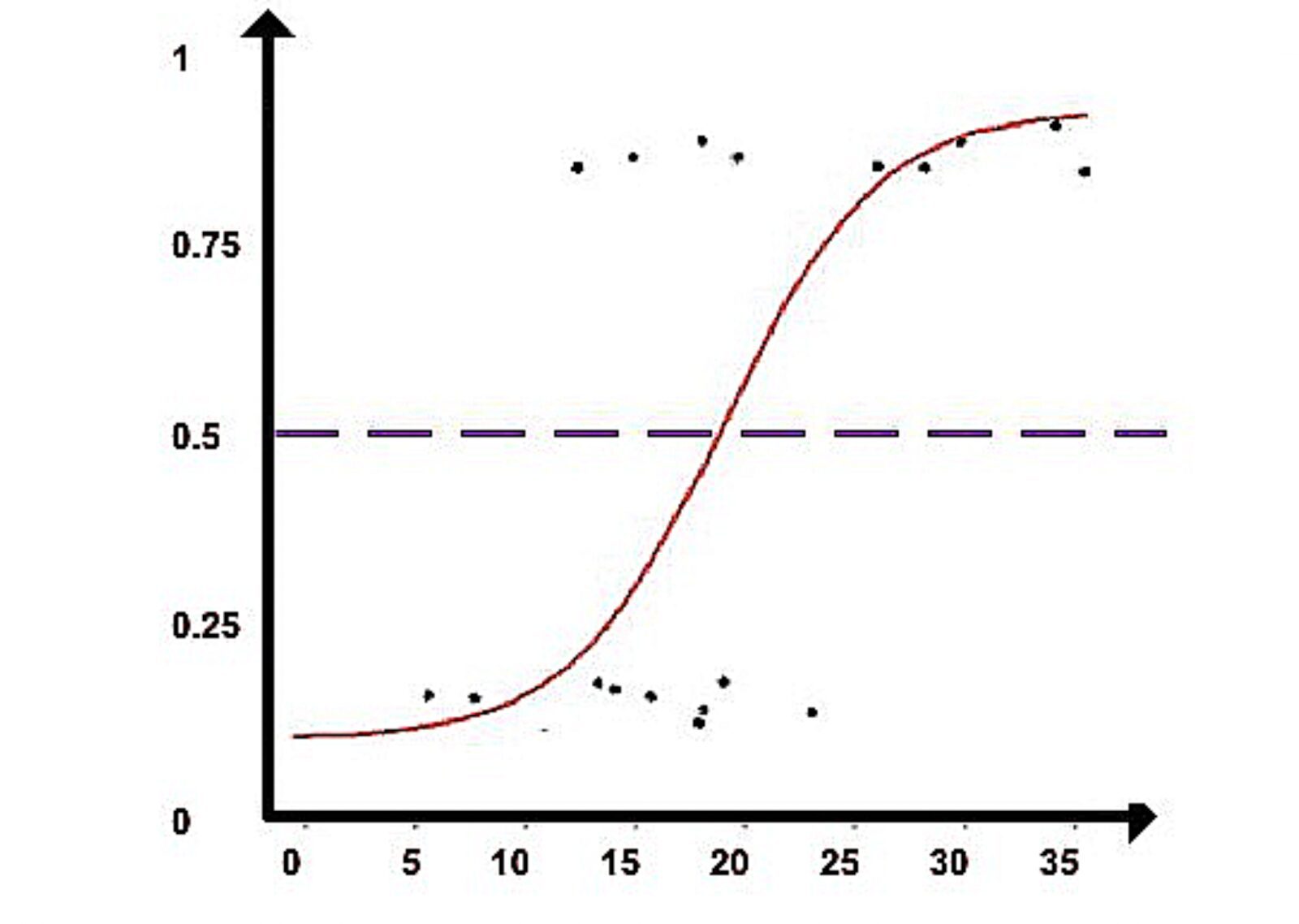
1. **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
2. **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
3. **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

**How Does Logistic Regression Work?**

Machine learning generally involves predicting a quantitative outcome or a qualitative class. The former is commonly referred to as a regression problem. In the scenario of linear regression, the input is a continuous variable, and the prediction is a numerical value. When predicting a qualitative outcome (class), the task is considered a classification problem. Examples of classification problems include predicting what products a user will buy or if a target user will click on an online advertisement.

Not all algorithms fit cleanly into this simple dichotomy, though, and logistic regression is a notable example. Logistic regression is part of the regression family as it involves predicting outcomes based on quantitative relationships between variables. However, unlike linear regression, it accepts both continuous and discrete variables as input and its output is qualitative. In addition, it predicts a discrete class such as “Yes/No” or “Customer/Non-customer”.

In practice, the [logistic regression algorithm](https://www.interviewbit.com/data-science-interview-questions/) analyzes relationships between variables. It assigns probabilities to discrete outcomes using the Sigmoid function, which converts numerical results into an expression of probability between 0 and 1.0. Probability is either 0 or 1, depending on whether the event happens or not. For binary predictions, you can divide the population into two groups with a cut-off of 0.5. Everything above 0.5 is considered to belong to group A, and everything below is considered to belong to group B.



**5. Software Requirements Specifications**

# **5.1 H/W System Configuration:**

|  |  |
| --- | --- |
| Processor | **Dual Core.** |
| Speed | **1.1 G Hz.** |
| RAM | **8 GB (min).** |
| Hard Disk | **B.** |

# **5.2 S/W System Configuration:**

|  |  |
| --- | --- |
| Operating System | **Windows 11.** |
| Technology | **Machine Learning.** |
| IDLE | **Python 3.7 or higher.** |

**Methodology:**

1. We are using dataset.

2. We are Training the dataset.

3. Next we are applying the Logistic Regression algorithm.

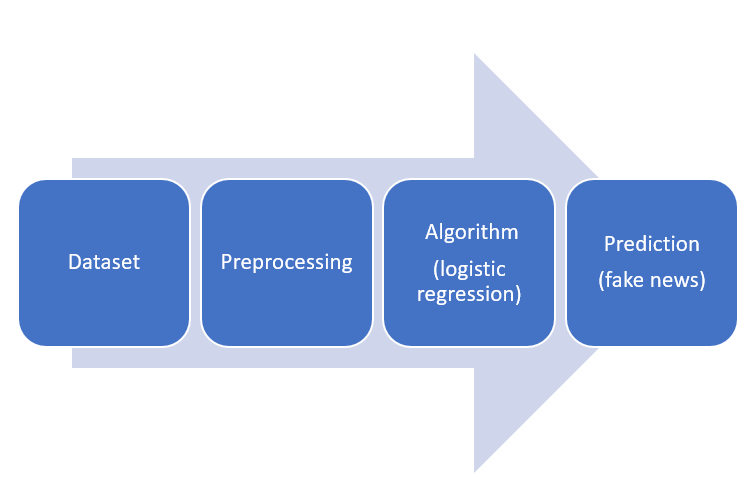
4. We are Testing the dataset.

5.Finally detect whether the News is Real or Fake and prediction is done.

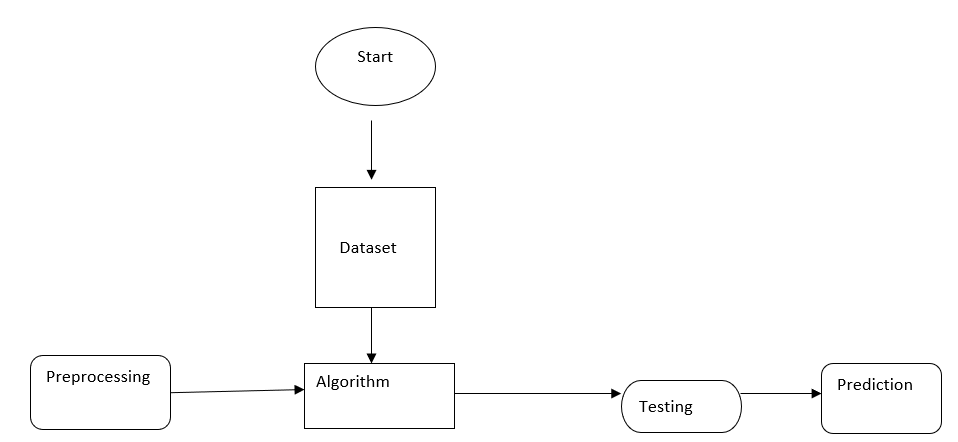
**6. System Design and Architecture**

**6.1 Design**

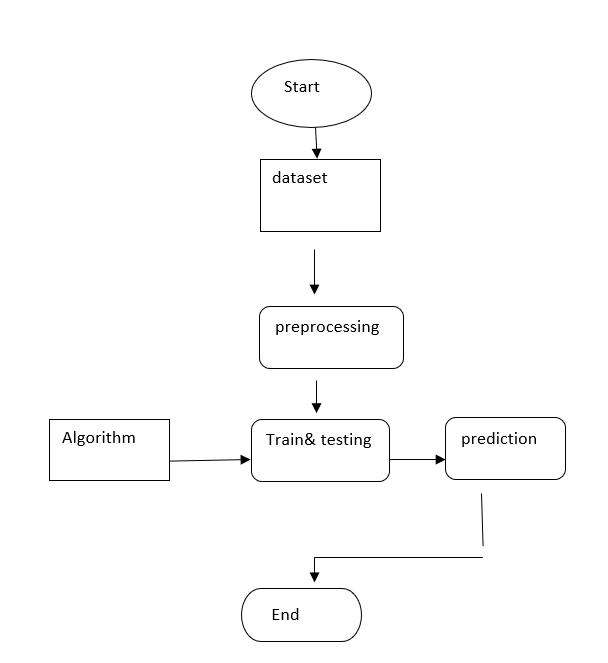
**6.1.1 Architecture Design/ System Architecture**



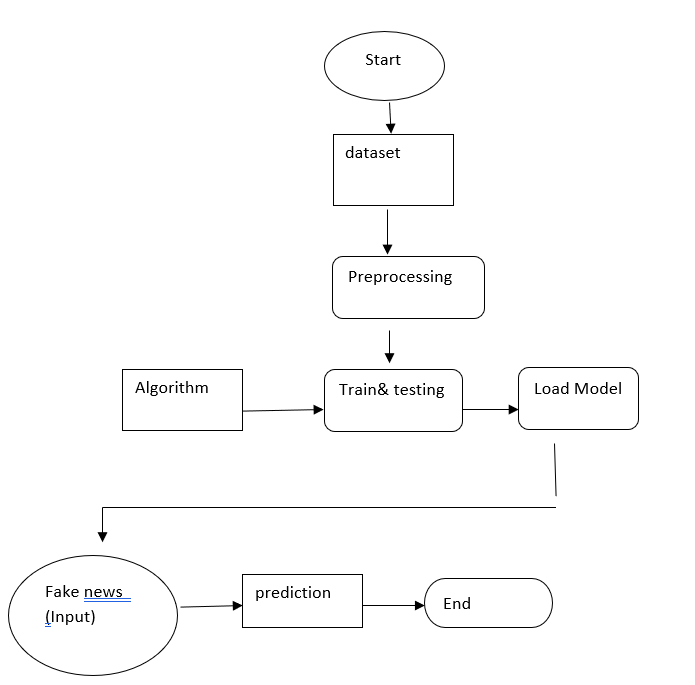
Algorithm



**Fig1:** Architecture of the proposed model



**Fig2.1:** Flow chart for training.



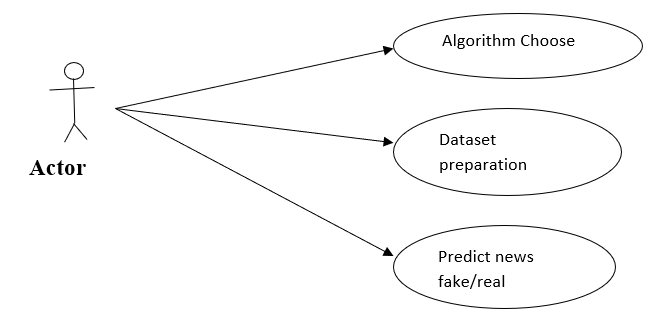
**Fig2.2:** Flow chart for testing.

**UML Diagrams**

Unified Modelling Language (UML) is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created, by the Object Management Group.

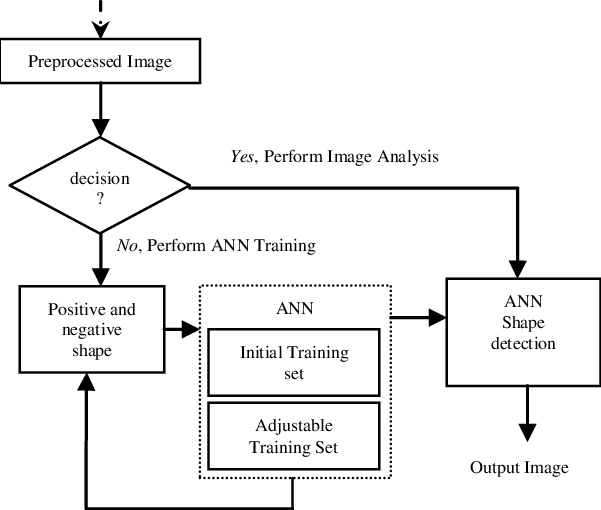
**Use Case Diagrams**

A use case diagram at its simplest is a graphical representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.

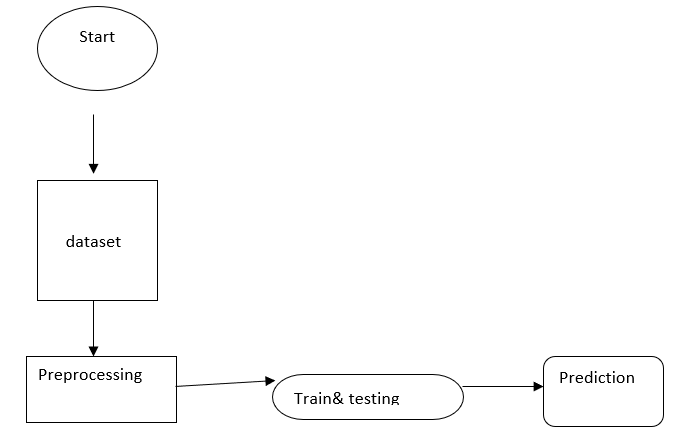


**Sequence Diagram:**

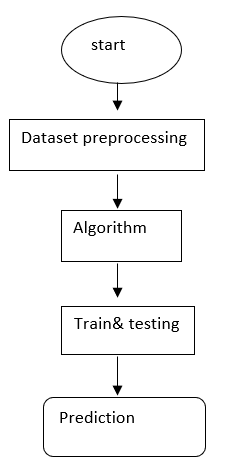
A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.



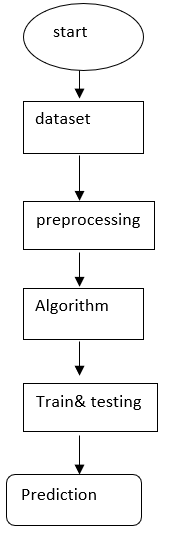
**Activity diagram:**



# **Object diagram:**



**Components diagram:**



**Output Image:**

**Add some output images….**

**7.TESTING**

Testing is a critical element which assures quality and effectiveness of the proposed system in (satisfying) meeting its objectives. Testing is done at various stages in the System designing and implementation process with an objective of developing an transparent, flexible and secured system. Testing is an integral part of software development. Testing process, in a way certifies, whether the product, that is developed, complies with the standards, that it was designed to. Testing process involves building of test cases, against which, the product has to be tested.

**Test objectives**

* Testing is a process of executing a program with the intent of finding an error.
* A good case is one that has a high probability of finding an undiscovered error.
* A successful test is one that uncovers a yet undiscovered error. If testing is conducted successfully (according to the objectives) it will uncover errors in the software. Testing can't show the absences of defects are present. It can only show that software defects are present.

**Testing principles**

Before applying methods to design effective test cases, a software engineer must understand the basic principle that guides software testing. All the tests should be traceable to customer requirements.

**Testing design**

Any engineering product can be tested in one of two ways:

**White box Testing**

This testing is also called as glass box testing. Inthis testing, by knowing the specified function that a product has been designed to perform test can be conducted that demonstrates each function is fully operation at the same time searching for errors in each function. it is a test case design method that uses the control structure of the procedural design to derive test cases.

**Black box Testing**

Inthis testing by knowing the internal operation of a product, tests can be conducted to ensure that "all gears mesh", that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

* Graph based testing methods
* Equivalence partitioning
* Boundary value analysis
* Comparison testing

**Testing strategies**

A software testing strategy provides a road map for the software developer. Testing is a set of activities that can be planned in advanced and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be defined for software engineering process.

**Any software testing strategy should have the following characteristics:**

* 1. Testing begins at the module level and works outward toward the integration of the entire computer-based system.
  2. Different testing techniques are appropriate at different points in time.
  3. The developer of the software and an independent test group conducts testing.
  4. Testing and debugging are different activities but debugging must be accommodated in any testing strategy.

**Levels of Testing**

Testing can be done in different levels of SDLC. They are:

**Unit Testing**

The first level of testing is called unit testing. Unit testing verifies on the smallest unit of software designs-the module. The unit test is always white box oriented. In this, different modules are tested against the specifications produced during design for the modules. Unit testing is essentially for verification of the code produced during the coding phase, and hence the goal is to test the internal logic of the modules. It is typically done by the programmer of the module. Due to its close association with coding, the coding phase is frequently called “coding and unit testing.” The unit test can be conducted in parallel for multiple modules.

**Integration Testing**

The second level of testing is called integration testing. Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. In this, many tested modules are combined into subsystems, which are then tested. The goal here is to see if all the modules can be integrated properly.

There are three types of integration testing:

* + - *Top-Down Integration*: Top-down integration is an incremental approach to construction of program structures. Modules are integrated by moving downwards throw the control hierarchy beginning with the main control module.
    - *Bottom-Up Integration*: Bottom-up integration as its name implies, begins Construction and testing with automatic modules.
    - *Regression Testing*: In this contest of an integration test strategy, regression testing is the re execution of some subset of test that have already been conducted to ensure that changes have not propagated unintended side effects.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Table X: Functional Testing items

|  |  |
| --- | --- |
| Input Images | Input must be accepted. |
| Train | Model will analyse the fake news and train model then  Save model |
| Testing | Yes, pass. |
| Output | Yes, the model predicted, and model analyses whether the news is real or fake. |

**Systems/Procedures:** Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**Validation testing**

At the culmination of integration testing, software is completely assembled as a package; interfacing errors have been covered and corrected, and final series of software tests-validating testing may begin. Validation can be defined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by customers. Reasonable expectation is defined in the software requirement specification- a document that describes all user visible attributes of the software. The specification contains a section title “validation criteria”. Information contained in that section forms the basis for validation testing approach.

**Alpha testing**

It is virtually impossible for a software developer to for see how the customer will really use a program. Instructions for use may be misinterpreted; strange combination of data may be regularly used and output that seemed clear to the tester may be unintelligible to a user in field.

When custom software is built for one customer, a series of acceptance tests are conducted to enable the customer to validate all requirements by the end user rather than system developer and acceptable test can range from an informal “test drive” to a planned and systematically executed series of tests. In fact, acceptance testing can be conducted over a period of weeks or months, thereby uncovering cumulative errors that might degrade the system over time. If software is developed as a product to be used by many customers, it is impractical to perform formal acceptance test with each one. Most software product builders use a process called alpha and beta testing to uncover errors that only the end user seems able to find.

A customer conducts the alpha test at the developer’s site. The software is used in a natural setting with the developer “Looking over the shoulder” of the user and recording errors and usage problems. Alpha tests are conducted in controlled environment.

**Beta testing**

The beta test is conducted at one or more customer sites by the end user of the software. Unlike alpha testing, the developer is generally not present. Therefore, the beta test is a “live” application of the software in an environment that cannot be controlled by the developer. The customer records all problems that are encountered during beta testing and reports these to the developer at regular intervals. As a result of problems reported during beta test, the software developer makes modification and then prepares for release of the software product to the entire customer base.

**System Testing and Acceptance Testing**

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Include recovery testing during crashes, security testing for unauthorized user, etc.

**Test case**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Description** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| It will predict if the news is real when you enter | Pope Francis used his annual Christmas Day mes.. | Fake | Real | Fail |
| It will predict if the news is fake when you enter | Donald Trump just couldn’t wish all Americans ... | Fake | Fake | Pass |

**Conclusion:**

Vast spreading of fake news through the net will deliver bad impacts to the society. Fake news will mislead readers and deceive them to the ultimate confusion in believing something that is not true to be true. This is the danger of fake news as people nowadays still are unable to differentiate between fake news and real news in their daily life with their naked eyes. However, this problem can be certainly solved by harnessing the power of machine learning to predict news to be fake or not. Within this capability. Here we predicted the accuracy: 97.21 for the real and fake news using Logistic Regression.

**References:**

1. <https://www.jetir.org/papers/JETIRFN06014.pdf>
2. <https://www.javatpoint.com/logistic-regression-in-machine-learning>
3. <https://www.kdnuggets.com/2022/07/logistic-regression-work.html>
4. <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/>
5. <https://www.pantechsolutions.net/fake-news-detection-using-machine-learning>
6. <https://in.images.search.yahoo.com/search/images;_ylt=AwrPqBp.Z1NkGwQHc9m9HAx.;_ylu=c2VjA3NlYXJjaARzbGsDYnV0dG9u;_ylc=X1MDMjExNDcyMzAwNQRfcgMyBGZyA21jYWZlZQRmcjIDcDpzLHY6aSxtOnNiLXRvcARncHJpZAMybVhzSUVsWVRZT0U0QjFOX2xuZnVBBG5fcnNsdAMwBG5fc3VnZwMwBG9yaWdpbgNpbi5pbWFnZXMuc2VhcmNoLnlhaG9vLmNvbQRwb3MDMARwcXN0cgMEcHFzdHJsAzAEcXN0cmwDMzAEcXVlcnkDc2VxdWVuY2UlMjBkaWFncmFtJTIwZm9yJTIwZGV0ZWN0aW9uBHRfc3RtcAMxNjgzMTg3NzQw?p=sequence+diagram+for+detection&fr=mcafee&fr2=p%3As%2Cv%3Ai%2Cm%3Asb-top&ei=UTF-8&x=wrt#id=61&iurl=https%3A%2F%2Fwww.researchgate.net%2Fprofile%2FNavneet_Singh13%2Fpublication%2F237100263%2Ffigure%2Ffig3%2FAS%3A667121986052104%401536065684364%2FFlow-chart-of-buried-object-detection-by-neural-network.png&action=click>