

A Machine Learning Approach For Fake News Detection

**Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Computer Science and Engineering**

**by
Names of all the students with
register Number
T.RAJESH(38110595),
T.SRIVAKSHA(38110597)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI -
600 119**

MARCH - 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of, T.RAJESH(38110595), T.SRIVAKSHA(38110597) who carried out the project entitled "A MACHINE LEARNING APPROACH FOR FAKE NEWS DETECTION" under my supervision .

Dr. S. Prince Mary M.E., Ph.D.,
Internal Guide

DR. L. LAKSHMANAN M.E., Ph.D.

DR. S. VIGNESHWARI M.E., Ph.D.

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

Examiner

External

DECLARATION

I **T.RAJESH(38110595)** hereby declare that the Project Report entitled “**A machine learning approach for fake news detection**” done by me under the guidance of **Dr. S. Prince Mary M.E.,Ph.D.,**(Internal) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in computer science and engineering

DATE:09-04-2022

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. S. Prince Mary M.E.,Ph.D.,for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

Abstract

Digital media has become a significant factor in many person's day to day routine. Fake news is a story that has been created in an intention to distract or misguide the readers. Due to increase in the online social network development in the past few years due to different purposes fake news appear in large numbers and in the online world has a widespread. By these online fake news online social networks users can get effected easily Fake news have become a society problem, in some occasion spreading more and faster than the true information. A human being is unable to detect all these fake news. So there is a need for machine learning model that can detect these fake news automatically. Machine learning models are made to build using the algorithms so that it can classify whether a news is fake or not.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	I
	TABLES AND CONTENTS	II
1	INTRODUCTION	1
2	LITERATURE SURVAY	5
2.1	PROPOSED SYSTEM	9
2.2	EXISTING SYSTEM	11
3	SCOPE OF THE PROJECT	12
3.1	OBJECTIVES	12
3.2	PROJECT GOALS	12
3.3	FEASIBILITY STUDY	13
4	METHODOLOGY	14
4.1	LIST OF MODULES	14
4.2	ALGORITHM EXPLANATION	14
4.3	SYSTEM ARCHITECTURE	20
4.3.1	WORKFLOW DIAGRAM	21

5	RESULT	22
5.1	DATA VALIDATION	22
5.2	ACCURACY	29
6	CONCLUSION	32
6.1	FUTURE WORK	32
6.2	REFERENCES	33
6.3	APENDICES	33
6.3.1	CODING	33
6.3.2	SCREEN SHOTS	45
6.3.3	PLAGARISM REPORT	46

LIST OF FIGURES

FIG NO.	TITTLE	PAGE NO.
2.1	ARCHITECTURE OF PROPOSED SYSTEM	10
4.3	SYSTEM ARCHITECTURE	20
4.3.1	WORKFLOW DIAGRAM	31
5.1	DATA VALIDATION	26
5.2	PREPROCEING DATA	27

CHAPTER-1

INTRODUCTION

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains.

The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science. In 1996, the International Federation of Classification Societies became the first conference to specifically feature data science as a topic. However, the definition was still in flux.

The term "data science" was first coined in 2008 by D.J. Patil, and Jeff Hammerbacher, the pioneer leads of data and analytics efforts at LinkedIn and Facebook. In less than a decade, it has become one of the hottest and most trending professions in the market.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data.

Data science can be defined as a blend of mathematics, business acumen, tools, algorithms and machine learning techniques, all of which help us in finding out the hidden insights or patterns from raw data which can be of major use in the formation of big business decision.

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by humans or animals. Leading AI textbooks

define the field as the study of "intelligent agents" any system that perceives its environment and takes actions that maximize its chance of achieving its goals. Some popular accounts use the term "artificial intelligence" to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving", however this definition is rejected by major AI researchers.

Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.

Artificial intelligence was founded as an academic discipline in 1956, and in the years since has experienced several waves of optimism, followed by disappointment and the loss of funding (known as an "AI winter"), followed by new approaches, success and renewed funding. AI research has tried and discarded many different approaches during its lifetime, including simulating the brain, modeling human problem solving, formal logic, large databases of knowledge and imitating animal behavior. In the first decades of the 21st century, highly mathematical statistical machine learning has dominated the field, and this technique has proved highly successful, helping to solve many challenging problems throughout industry and academia.

Natural language processing (NLP) allows machines to read and understand human language. A sufficiently powerful natural language processing system would enable natural-language user interfaces and the acquisition of knowledge directly from human-written sources, such as newswire texts. Some straightforward applications of natural language processing include information retrieval, text mining, question answering and machine translation. Many current approaches use word co-occurrence frequencies to construct syntactic representations of text. "Keyword spotting" strategies for search are popular and scalable but dumb; a search query for "dog" might only match documents with the literal word "dog" and miss a document with the word "poodle". "Lexical affinity" strategies use the occurrence of words such as "accident" to assess the sentiment of a document. Modern statistical

NLP approaches can combine all these strategies as well as others, and often achieve acceptable accuracy at the page or paragraph level. Beyond semantic NLP, the ultimate goal of "narrative" NLP is to embody a full understanding of commonsense reasoning. By 2019, transformer-based deep learning architectures could generate coherent text.

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

Data scientists use many different kinds of machine learning algorithms to discover patterns in python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they “learn” about data to make predictions: supervised and unsupervised learning. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function from input variables(X) to discrete output variables(y). In machine learning and statistics, classification is a supervised learning

approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.



Process Of Machine Learning

Supervised Machine Learning is the majority of practical machine learning uses supervised learning. Supervised learning is where have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output is $y = f(X)$. The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include logistic regression, multi-class classification, Decision Trees and support vector machines etc. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. Supervised learning problems can be further grouped into Classification problems. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for categorical for classification. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. A

classification problem is when the output variable is a category, such as “red” or “blue”.

CHAPTER-2

LITERATURE SURVEY

General

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them

Review of Literature Survey

Title : Automatic Deception Detection: Methods for Finding Fake News

Author: Niall J. Conroy, Victoria L. Rubin

Year : 2015

This research surveys the current state-of-the-art technologies that are instrumental in the adoption and development of fake news detection. “Fake news detection” is defined as the task of categorizing news along a continuum of veracity, with an associated measure of certainty. Veracity is compromised by the occurrence of intentional deceptions. The nature of online news publication has changed, such that traditional fact checking and vetting from potential deception is impossible against the flood arising from content generators, as well as various formats and genres. The paper provides a typology of several varieties of veracity assessment methods emerging from two major categories – linguistic cue approaches (with machine learning), and network analysis approaches. We see promise in an innovative hybrid approach that combines linguistic cue and machine learning, with network-based behavioral data. Although designing a fake news detector is not a straightforward problem, we propose operational guidelines for a feasible fake news detecting system.

Title : Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques

Author: Hadeer Ahmed

Year : 2017

In the recent years, online content has been playing a significant role in swaying users decisions and opinions. Opinions such as online reviews are the main source of information for e-commerce customers to help with gaining insight into the products they are planning to buy. Recently it has become apparent that opinion spam does not only exist in product reviews and customers’ feedback. In fact, fake news and misleading articles is another form of opinion spam, which has gained traction. Some of the biggest sources of spreading fake news or rumors are social

media websites such as Google Plus, Facebook, Twitters, and other social media outlet [1]. Even though the problem of fake news is not a new issue, detecting fake news is believed to be a complex task given that humans tend to believe misleading information and the lack of control of the spread of fake content [2]. Fake news has been getting more attention in the last couple of years, especially since the US election in 2016. It is tough for humans to detect fake news. It can be argued that the only way for a person to manually identify fake news is to have a vast knowledge of the covered topic. Even with the knowledge, it is considerably hard to successfully identify if the information in the article is real or fake.

Title : Development of Fake News Model Using Machine Learning through Natural Language Processing

Author: Sajjad Ahmed, Knut Hinkelmann

Year : 2020

Fake news detection research is still in the early stage as this is a relatively new phenomenon in the interest raised by society. Machine learning helps to solve complex problems and to build AI systems nowadays and especially in those cases where we have tacit knowledge or the knowledge that is not known. We used machine learning algorithms and for identification of fake news; we applied three classifiers; Passive Aggressive, Naïve Bayes, and Support Vector Machine. Simple classification is not completely correct in fake news detection because classification methods are not specialized for fake news. With the integration of machine learning and text-based processing, we can detect fake news and build classifiers that can classify the news data. Text classification mainly focuses on extracting various features of text and after that incorporating those features into classification. The big challenge in this area is the lack of an efficient way to differentiate between fake and non-fake due to the unavailability of corpora. We applied three different machine

learning classifiers on two publicly available datasets. Experimental analysis based on the existing dataset indicates a very encouraging and improved performance.

Title : Fake News Detection on Social Media: A Data Mining Perspective

Author: Kai Shu , Amy Sliva, Suhang Wang

Year : 2019

Social media for news consumption is a double-edged sword. On the one hand, its low cost, easy access, and rapid dissemination of information lead people to seek out and consume news from social media. On the other hand, it enables the wide spread of “fake news”, i.e., low quality news with intentionally false information. The extensive spread of fake news has the potential for extremely negative impacts on individuals and society. Therefore, fake news detection on social media has recently become an emerging research that is attracting tremendous attention. Fake news detection on social media presents unique characteristics and challenges that make existing detection algorithms from traditional news media ineffective or not applicable. First, fake news is intentionally written to mislead readers to believe false information, which makes it difficult and nontrivial to detect based on news content; therefore, we need to include auxiliary information, such as user social engagements on social media, to help make a determination. Second, exploiting this auxiliary information is challenging in and of itself as users’ social engagements with fake news produce data that is big, incomplete, unstructured, and noisy. Because the issue of fake news detection on social media is both challenging and relevant, we conducted this survey to further facilitate research on the problem. In this survey, we present a comprehensive review of detecting fake news on social media, including fake news characterizations on psychology and social theories, existing algorithms from a data mining perspective, evaluation metrics and representative datasets. We also discuss related research areas, open problems, and future research directions for fake news detection on social media.

Title : Fake Job Recruitment Detection Using Machine Learning Approach

Author: Shawni Dutta#1 and Prof.Samir Kumar Bandyopadhyay

Year : 2020

To avoid fraudulent post for job in the internet, an automated tool using machine learning based classification techniques is proposed in the paper. Different classifiers are used for checking fraudulent post in the web and the results of those classifiers are compared for identifying the best employment scam detection model. It helps in detecting fake job posts from an enormous number of posts. Two major types of classifiers, such as single classifier and ensemble classifiers are considered for fraudulent job posts detection. However, experimental results indicate that ensemble classifiers are the best classification to detect scams over the single classifiers. Employment scam detection will guide job-seekers to get only legitimate offers from companies. For tackling employment scam detection, several machine learning algorithms are proposed as countermeasures in this paper. Supervised mechanism is used to exemplify the use of several classifiers for employment scam detection. Experimental results indicate that Random Forest classifier outperforms over its peer classification tool. The proposed approach achieved accuracy 98.27% which is much higher than the existing methods.

2.1 PROPOSED SYSTEM

The proposed model is to build a machine learning model that is capable of classifying whether the news is fake or not. The fake news are considered to be widespread and controlling them is very difficult as the world is developing toward digital everyone now has access to internet and they can post whatever they want. So there is a greater chance for the people to get misguided. The machine learning is generally build to tackle these type of complicated task like it takes more amount of time to analyse these type of data manually. The machine learning can be used to classify the news is fake or not by using the previous data and make them to understand the pattern and improve the accuracy of the model by adjusting parameters and use that model as the classification model. Different algorithms can be compared and the best model can be used for classification purpose.

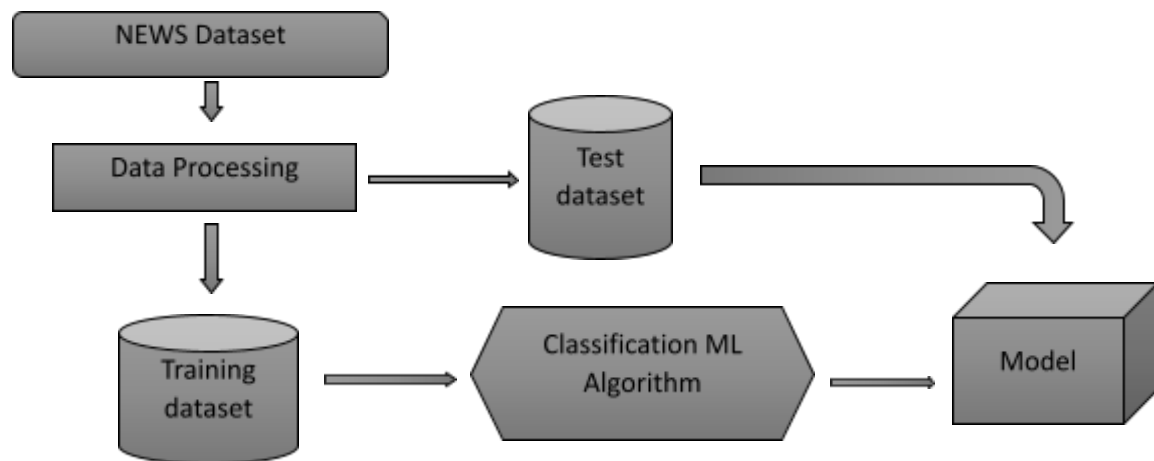


Fig.2.1 Architecture of Proposed model

Advantages:

1. The classification model build had the ability to classify the new whether it is real or not

2.By classifying the new by using this process the chance of people getting misguided can be reduced.

2.2 EXISTING SYSTEM

They propose a fake news detection multitask learning (FDML) model to improve performances of short fake news detection and topic classification, simultaneously. The FDML model was such that news with certain topics have high probabilities to be classified as fake news and some authors have high probability to publish fake news. FDML introduce a task gate to selective integrate representations based on different tasks and design a dynamic weight strategy to balance the importance between two tasks. The imbalance learning problem to propose a simple dynamic weighting strategy where the weight of each task is dynamically adjusted in each iteration. FDML model consists of representation learning and multi-task learning parts to train the fake news detection task and the news topic classification.

Disadvantages:

1. The proposed model had problem with classification as it is not that much accurate.

To a certain extent only it will predict correctly

CHAPTER-3

SCOPE OF THE PROJECT

The main Scope is to detect the fake news, which is a classic text classification problem with a help of NLP and machine learning algorithm. It is needed to build a model that can differentiate between “Real” news and “Fake” news

3.1 OBJECTIVES

The goal is to develop a machine learning model for real and fake news Prediction, to potentially replace the updatable supervised machine learning classification models by predicting results in the form of best accuracy by comparing supervised algorithm.

3.2 PROJECT GOALS

Exploration data analysis of variable identification

- Loading the given dataset
- Import required libraries packages
- Analyze the general properties
- Find duplicate and missing values
- Checking unique and count values

Uni-variate data analysis

- Rename, add data and drop the data
- To specify data type

Exploration data analysis of bi-variate and multi-variate

- Plot diagram of pairplot, heatmap, bar chart and Histogram

Method of Outlier detection with feature engineering

- Pre-processing the given dataset
- Splitting the test and training dataset
- Comparing the Decision tree and Logistic regression model and random forest etc.

Comparing algorithm to predict the result

- Based on the best accuracy

3.3 Feasibility study

Data Wrangling

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

Data collection

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, logistic, Decision tree algorithms and Support vector classifier (SVC) are applied on the Training set and based on the test result accuracy, Test set prediction is done.

Preprocessing

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done.

Building the classification model

The prediction of real and fake news, A high accuracy prediction model is effective because of the following reasons: It provides better results in classification problem.

It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.

It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

CHAPTER-4

METHODOLOGY

4.1 LIST OF MODULES

Data Pre-processing

Data Analysis of Visualization

Comparing Algorithm with prediction in the form of best accuracy result

4.2 ALGORITHM EXPLANATION

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

Used Python Packages:

sklearn:

- In python, sklearn is a machine learning package which include a lot of ML algorithms.

- Here, we are using some of its modules like `train_test_split`, `DecisionTreeClassifier` or `Logistic Regression` and `accuracy_score`.

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas:

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

Remove stopwords: There are a few words which are very commonly used when humans interact, but these words don't make any sense or add any extra value. Additionally, there might be few words which are not required for the business case given in hand. So, these words need to be deleted from the data.

The NLTK package has a defined set of stopwords for different languages like English. Here, we will focus on 'english' stopwords. One can also consider additional stopwords if required.

Tokenization : This is one of the common practices while working on text data. This helps to split a phrase, sentence, or paragraph into small units like words or terms. Each unit is called a token. There are different types of tokenization. We have already used this in above examples for stemming, POS tagging, and NER. Below are different ways to tokenize the text.

Vectorization/Word Embedding

Once cleaning and tokenization is done, extracting features from the clean data is very important as the machine doesn't understand the words but numbers. Vectorization helps to map the words to a vector of real numbers, which further helps into predictions. This helps to extract the important features.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository.

Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Perez.

Notebook documents are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc...). Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc.) as well as executable documents which can be run to perform data analysis.

Launching Jupyter Notebook App: The Jupyter Notebook App can be launched by clicking on the Jupyter Notebook icon installed by Anaconda in the start menu (Windows) or by typing in a terminal (cmd on Windows): "jupyter notebook"

This will launch a new browser window (or a new tab) showing the Notebook Dashboard, a sort of control panel that allows (among other things) to select which notebook to open.

When started, the Jupyter Notebook App can access only files within its start-up folder (including any sub-folder). No configuration is necessary if you place your notebooks in your home folder or subfolders. Otherwise, you need to choose a Jupyter Notebook App start-up folder which will contain all the notebooks.

Save notebooks: Modifications to the notebooks are automatically saved every few minutes. To avoid modifying the original notebook, make a copy of the notebook document (menu file -> make a copy...) and save the modifications on the copy.

Executing a notebook: Download the notebook you want to execute and put it in your notebook folder (or a sub-folder of it).

- ❖ Launch the jupyter notebook app
- ❖ In the Notebook Dashboard navigate to find the notebook: clicking on its name will open it in a new browser tab.
- ❖ Click on the menu Help -> User Interface Tour for an overview of the Jupyter Notebook App user interface.
- ❖ You can run the notebook document step-by-step (one cell a time) by pressing shift + enter.
- ❖ You can run the whole notebook in a single step by clicking on the menu Cell -> Run All.
- ❖ To restart the kernel (i.e. the computational engine), click on the menu Kernel -> Restart. This can be useful to start over a computation from scratch (e.g. variables are deleted, open files are closed, etc...).

Purpose: To support interactive data science and scientific computing across all programming languages.

File Extension: An IPYNB file is a notebook document created by Jupyter Notebook, an interactive computational environment that helps scientists manipulate and analyze data using Python.

Logistic Regression

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts $P(Y=1)$ as a function of X . Logistic regression Assumptions:

Binary logistic regression requires the dependent variable to be binary.

For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.

Only the meaningful variables should be included.

The independent variables should be independent of each other. That is, the model should have little.

The independent variables are linearly related to the log odds.

Logistic regression requires quite large sample sizes.

Random Forest Classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

- Pick N random records from the dataset.

- Build a decision tree based on these N records.

- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

4.3 SYSTEM ARCHITECTURE

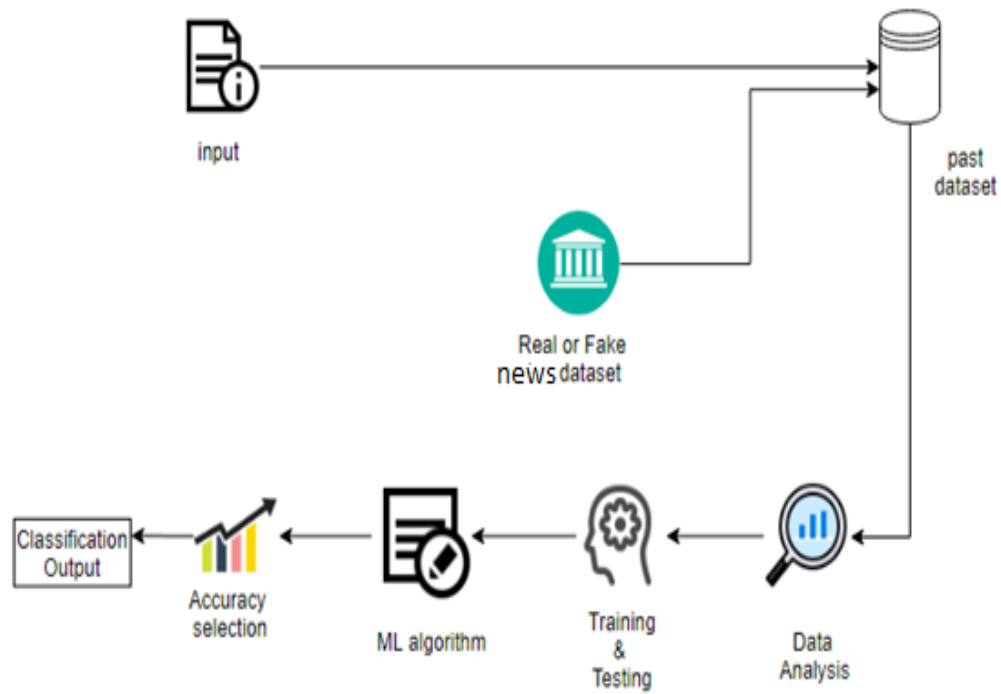


Fig.4.3 system architecture

4.3.1. WORK FLOW DIAGRAM

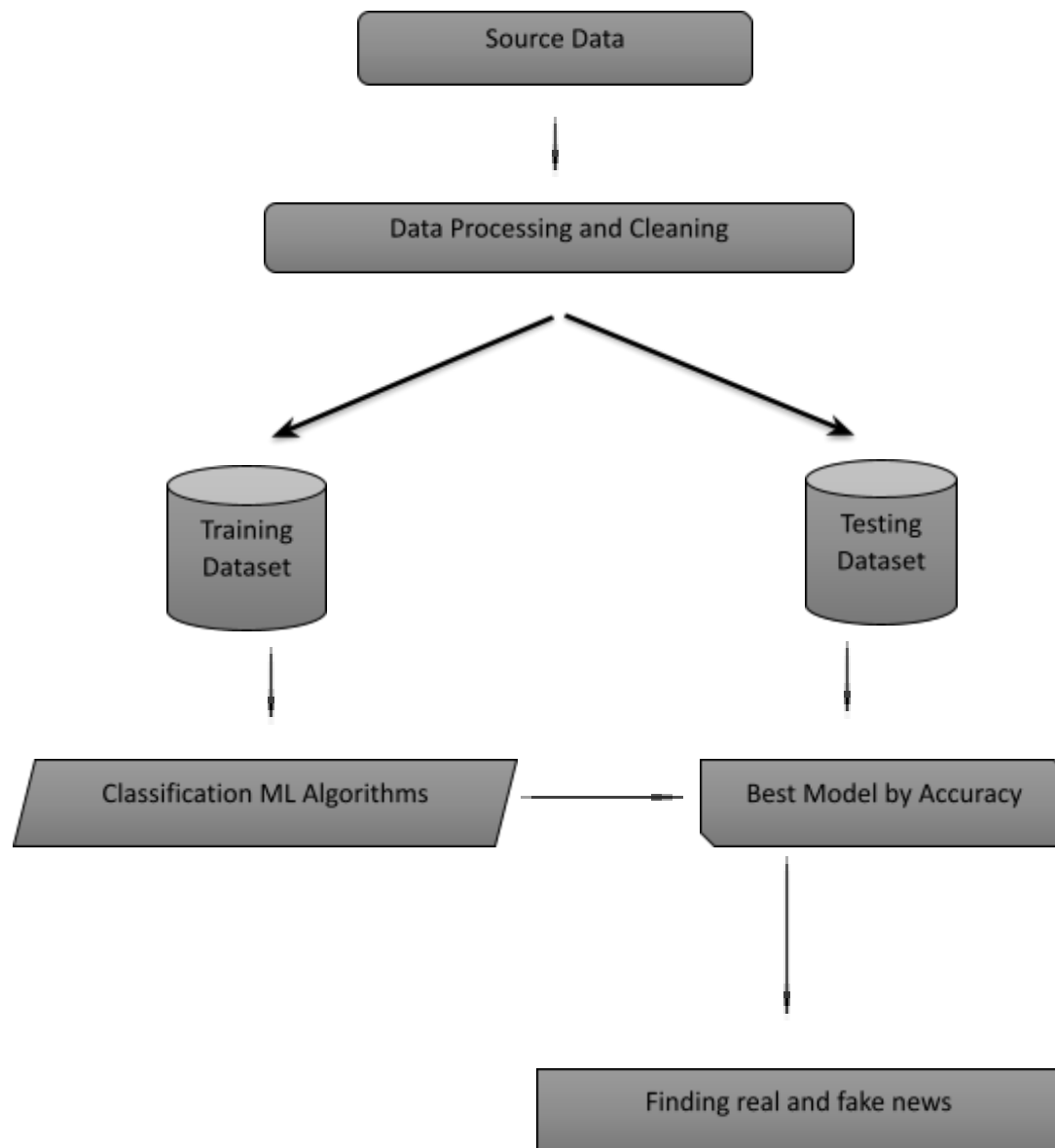


Fig 4.3.1 Workflow Diagram

CHAPTER-5

RESULT

5.1 DATA VALIDATION

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

Module 1: Data validation and pre-processing technique

```
In [1]: #import library packages
import pandas as pd
import numpy as np
```

```
In [2]: import warnings
warnings.filterwarnings("ignore")
```

```
In [3]: #Load given dataset
data = pd.read_csv("news.csv")
```

Before drop the given dataset:

```
In [4]: data.head()
```

```
Out[4]:
```

	Unnamed: 0		title		text	label
0	8476		You Can Smell Hillary's Fear		Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294		Watch The Exact Moment Paul Ryan Committed Pol...		Google Pinterest Digg LinkedIn Reddit Stumbleu...	FAKE
2	3608		Kerry to go to Paris in gesture of sympathy		U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142		Ramla supporters on Twitter rant in anger an		— Kavadee Kinn (@KavadeeKinn) November 9, 2016 T	FAKE

```
In [10]: #To describe the dataframe
df.describe()
```

Out[10]:

	Unnamed: 0
count	6335.000000
mean	5280.415627
std	3038.503953
min	2.000000
25%	2674.500000
50%	5271.000000
75%	7901.000000
max	10557.000000

```
In [11]: #Checking datatype and information about dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6335 entries, 0 to 6334
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -

```

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

import libraries for access and functional purpose and read the given dataset

General Properties of Analyzing the given dataset

Display the given dataset in the form of data frame

show columns

shape of the data frame

To describe the data frame

Checking data type and information about dataset

Checking for duplicate data

- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

Exploration data analysis of visualization

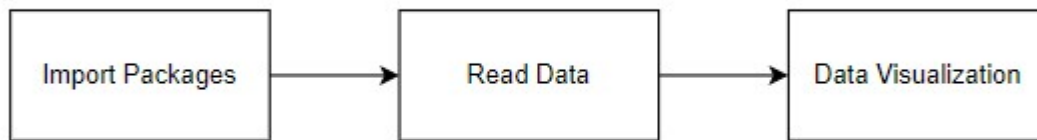
Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- How to chart time series data with line plots and categorical quantities with bar charts.

- How to summarize data distributions with histograms and box plots.

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : visualized data

```
In [9]: #plotting graph for distribution
import matplotlib.pyplot as plt
import seaborn as sns
sns.countplot(x = "label", data = df)
df.loc[:, 'label'].value_counts()
plt.title('Distribution of real and fake ')
```

```
Out[9]: Text(0.5, 1.0, 'Distribution of real and fake ')
```

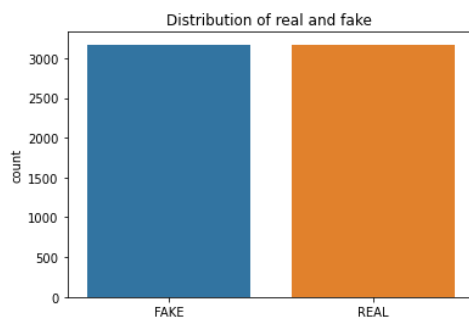


Fig.5.1

Out[17]: Text(0.5, 1.0, '')



sktpok/real and fake_news/CODE/M2.ipynb#

FIG.5.2

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class is no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

Comparing Algorithm with prediction in the form of best accuracy result

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 4 different algorithms are compared:

Logistic Regression

Random Forest

The K-fold cross validation procedure is used to evaluate each algorithm, importantly configured with the same random seed to ensure that the same splits to the training data are performed and that each algorithm is evaluated in precisely the same way. Before that comparing algorithm, Building a Machine Learning Model using install Scikit-Learn libraries. In this library package have to done preprocessing, linear model with logistic regression method, cross validating by KFold method, ensemble with random forest method and tree with decision tree classifier. Additionally, splitting the train set and test set. To predicting the result by comparing accuracy.

5.2 ACCURACY

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. It need the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

True Positive Rate(TPR) = $TP / (TP + FN)$

False Positive rate(FPR) = $FP / (FP + TN)$

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

Accuracy = $(TP + TN) / (TP + TN + FP + FN)$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct.

Precision = $TP / (TP + FP)$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = $TP / (TP + FN)$

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

General Formula:

$$F\text{- Measure} = 2TP / (2TP + FP + FN)$$

F1-Score Formula:

$$F1 \text{ Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

CHAPTER-6

CONCLUSION

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score will be find out. This application can help to find the Prediction of Real and fake news

6.1. FUTURE WORK

Real and fake news prediction to connect with cloud.

To optimize the work to implement in Artificial Intelligence environment.

6.2 REFERENCES

1. E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret, and L. de Al_x0002_faro, "Some like it hoax: Automated fake news detection in social networks," arXiv preprint arXiv:1704.07506, 2017.
2. K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, "defend: Explainable fake news detection," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, 2019, pp. 395–405.
3. N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," in Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community. American Society for Information Science, 2015, p. 82.
4. Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao, "EANN: event adversarial neural networks for multi-modal fake news detection," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 849–857.

6.3. APPENDICES

6.3.1. CODING

Module – 1

Pre-Processing

#import library packages

import pandas **as** pd

import numpy **as** np

In []:

import warnings

warnings.filterwarnings("ignore")

In []:

#Load given dataset

data = pd.read_csv("news.csv")

Before drop the given dataset:

In []:

data.head()

In []:

#shape

data.shape

After drop the given dataset:

In []:

df = data.dropna()

In []:

df.head()

In []:

#shape

df.shape

In []:

#columns

df.columns

In []:

#To describe the dataframe

df.describe()

In []:

#Checking datatype and information about dataset

df.info()

Checking duplicate values of dataframe

In []:

```
#Checking for duplicate data  
df.duplicated()
```

In []:

```
sum(df.duplicated())
```

In []:

```
#Checking sum of missing values  
df.isnull().sum()
```

In []:

```
df.label.unique()
```

In []:

```
df.title.unique()
```

In []:

```
from sklearn.preprocessing import LabelEncoder  
var=['Unnamed: 0', 'title', 'text', 'label']  
le=LabelEncoder()  
for i in var:  
    df[i]=le.fit_transform(df[i]).astype(int)  
  
df.head()
```

In []:

Module – 2

Visualization

```
#import library packages  
import pandas as p  
import matplotlib.pyplot as plt  
import seaborn as s  
import numpy as n
```

In []:

```
import warnings  
warnings.filterwarnings("ignore")
```

In []:

```
#Load given dataset  
data = p.read_csv('news.csv')  
df=data.dropna()
```

```
df
```

In []:

```
df.columns
```

In []:

Splitting Train/Test:

```
#preprocessing, split test and dataset, split response variable
```

```
X = df.drop(labels='label', axis=1)
```

```
#Response variable
```

```
y = df.loc[:, 'label']
```

In []:

```
#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)
```

```
print("Number of training dataset: ", len(X_train))
```

```
print("Number of test dataset: ", len(X_test))
```

```
print("Total number of dataset: ", len(X_train)+len(X_test))
```

In []:

```
df.groupby('label').describe()
```

In []:

```
#plotting graph for distribution
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
sns.countplot(x = "label", data = df)
```

```
df.loc[:, 'label'].value_counts()
```

```
plt.title('Distribution of real and fake ')
```

In []:

```
df['label'].unique()
```

In []:

Training model:

```
#!/pip install nltk
```

In []:

```
import nltk
```

```
nltk.download('stopwords')
```

In []:

```
import nltk
```

```
from nltk.corpus import stopwords
```

In []:

```

from nltk.stem.porter import PorterStemmer
import re
import string
# remove whitespaces
df['title']=df['title'].str.strip()
# lowercase the text
df['title'] = df['title'].str.lower()
#remove punctuation
punc = string.punctuation
table = str.maketrans("",punc)
df['title']=df['title'].apply(lambda x: x.translate(table))
# tokenizing each message
df['word_tokens']=df.apply(lambda x: x['title'].split(' '),axis=1)
# removing stopwords
df['cleaned_text'] = df.apply(lambda x: [word for word in x['word_tokens'] if word not
in stopwords.words('english')],axis=1)
# stemming
ps = PorterStemmer()
df['stemmed']= df.apply(lambda x: [ps.stem(word) for word in
x['cleaned_text']],axis=1)
# remove single letter words
df['final_text'] = df.apply(lambda x: ' '.join([word for word in x['stemmed'] if
len(word)>1]),axis=1)

```

In []:

```

# divide the set in training and test
from sklearn.model_selection import train_test_split
X,X_test,y,y_test = train_test_split(df.loc[:, 'title'],df['label'],test_size=0.2)

```

In []:

```

# Now we'll create a vocabulary for the training set with word count
from collections import defaultdict
vocab=defaultdict(int)
for text in X['final_text'].values:
    for elem in text.split(' '):
        vocab[elem]+=1

```

In []:

```

#!/pip install wordcloud

```

In []:

```

from wordcloud import WordCloud

real= ' '.join(X.loc[y=='REAL','final_text'].values)

```

```
real_text = WordCloud(background_color='white',max_words=2000,width = 800,  
height = 800).generate(real)
```

```
fake=' '.join(X.loc[y=='FAKE','final_text'].values)  
fake_text = WordCloud(background_color='black',max_words=2000,width = 800,  
height = 800).generate(fake)
```

```
plt.figure(figsize=[30,50])
```

```
plt.subplot(1,3,1)  
plt.imshow(real_text,interpolation='bilinear')  
plt.title("")  
plt.axis('off')
```

```
plt.subplot(1,3,2)  
plt.imshow(fake_text, interpolation='bilinear')  
plt.axis('off')  
plt.title("")
```

Module – 3

```
#import library packages
```

```
import pandas as p
```

```
import numpy as n
```

In []:

```
import warnings
```

```
warnings.filterwarnings("ignore")
```

In []:

```
#Load given dataset
```

```
data = p.read_csv('news.csv')
```

```
df=data.dropna()
```

In []:

```
df.columns
```

In []:

```
from sklearn.preprocessing import LabelEncoder
var_mod = ['Unnamed: 0', 'title', 'text', 'label']
le = LabelEncoder()
for i in var_mod:
    df[i] = le.fit_transform(df[i]).astype(int)
```

In []:

```
#According to the cross-validated MCC scores, the random forest is the
best-performing model, so now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report,
matthews_corrcoef, cohen_kappa_score, accuracy_score, average_precision_score,
roc_auc_score
```

In []:

```
X = df.drop(labels='label', axis=1)
#Response variable
y = df.loc[:, 'label']
```

In []:

```
#We'll use a test size of 30%. We also stratify the split on the response variable,
which is very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1,
stratify=y)
```

Decision Tree Classifier :

In []:

```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()

dtree.fit(X_train,y_train)
predictR = dtree.predict(X_test)
print("")
x = (accuracy_score(y_test,predictR)*100)
print('Accuracy result ofDecision Tree Classifier is:', x)
print("")

print("")
print('Classification report of Decision Tree Classifier : Results:')
print("")

print(classification_report(y_test,predictR))
xd = (accuracy_score(y_test,predictR)*100)
```

```

cm2=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Decision Tree Classifier : is:\n',
confusion_matrix(y_test,predictR))
print("")

```

```

sensitivity1 = cm2[0,0]/(cm2[0,0]+cm2[0,1])
print('Sensitivity : ', sensitivity1 )
print("")
specificity1 = cm2[1,1]/(cm2[1,0]+cm2[1,1])
print('Specificity : ', specificity1)

```

Logistic Regression:

In []:

```

from sklearn.linear_model import LogisticRegression
logR= LogisticRegression()
logR.fit(X_train,y_train)
predictR = logR.predict(X_test)
print("")
x = (accuracy_score(y_test,predictR)*100)
print('Accuracy result of Logistic Regression is:', x)
print("")

print("")
print('Classification report of Logistic Regression : Results:')
print("")

print(classification_report(y_test,predictR))
xl = (accuracy_score(y_test,predictR)*100)

cm2=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Logistic Regression : is:\n',
confusion_matrix(y_test,predictR))
print("")

sensitivity1 = cm2[0,0]/(cm2[0,0]+cm2[0,1])
print('Sensitivity : ', sensitivity1 )
print("")
specificity1 = cm2[1,1]/(cm2[1,0]+cm2[1,1])
print('Specificity : ', specificity1)

```

In []:

```
def graph():
    import matplotlib.pyplot as plt
    data=[xd,xl]
    alg="DT","LR"
    plt.figure(figsize=(10,5))
    b=plt.bar(alg,data,color=("green","yellow"))
    plt.title("Prediction of Opinion real or fake using ML",fontsize=15)
    plt.legend(b,data,fontsize=12)
    plt.savefig('comp.png')
```

In []:

```
graph()
```

In []:

```
import tkinter
from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg,
NavigationToolbar2Tk)
from matplotlib.backend_bases import key_press_handler
from matplotlib.figure import Figure
import numpy as np
root = tkinter.Tk()
root.wm_title("Prediction of real or fake news using ML")
fig = Figure(figsize=(10,10),dpi=1)
canvas = FigureCanvasTkAgg(fig, master=root)
canvas.draw()
canvas.get_tk_widget().pack(side=tkinter.TOP, fill=tkinter.BOTH, expand=1)
icon=tkinter.PhotoImage(file='comp.png')
label=tkinter.Label(root,image=icon)
label.pack()
root.mainloop()
```

Module – 4

```
import numpy as np
import pandas as pd
```

In []:

```
data=pd.read_csv("news.csv")
```

```

data.head()
data.shape
data.columns
data1=data.dropna()
data1.shape
data1.label.unique()

import matplotlib.pyplot as plt
import seaborn as sns

sns.countplot(x = "label", data=data1)
data1.loc[:, 'label'].value_counts()
plt.title('Distribution of fraudulent')

data1.label.value_counts()

import re
import string
import nltk
from nltk.corpus import stopwords
stopword=set(stopwords.words('english'))
stemmer = nltk.SnowballStemmer("english")

def clean(text):
    text = str(text).lower()
    text = re.sub('[\.\*?\]', "", text)
    text = re.sub('https?://\S+|www\.\S+', "", text)
    text = re.sub('<.*?>+', "", text)
    text = re.sub('[%s]' % re.escape(string.punctuation), "", text)
    text = re.sub('\n', "", text)
    text = re.sub('\w*\d\w*', "", text)
    text = [word for word in text.split(' ') if word not in stopwords]
    text="" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text="" ".join(text)

```

```
return text
data["title"] = data["title"].apply(clean)
```

In []:

```
x = np.array(data1["title"])
y = np.array(data1["label"])
```

In []:

```
x
```

In []:

```
y
```

In []:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
```

In []:

```
cv = CountVectorizer()
X = cv.fit_transform(x) # Fit the Data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=42)
```

In []:

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
clf.fit(X_train,y_train)
```

In []:

```
pre=clf.predict(X_test)
```

In []:

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,pre)*100)
```

In []:

```
input_word=input("ENTER THE SENTENCE:")
```

In []:

```
data = cv.transform([input_word]).toarray()
print(clf.predict(data))
```

Module – 5

```
import numpy as np
import pandas as pd
```

In []:

```
data=pd.read_csv("news.csv")
```

In []:

```
data.head()
```

In []:

```
data.shape
```

In []:

```
data.columns
```

In []:

```
data1=data.dropna()
```

```
data1.shape
```

In []:

```
print(data1.head())
```

In []:

```
data1.label.unique()
```

In []:

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

In []:

```
sns.countplot(x = "label", data=data1)
```

```
data1.loc[:, 'label'].value_counts()
```

```
plt.title('Distribution of REAL AND FAKE')
```

In []:

```
data1.label.value_counts()
```

In []:

```
import re
```

```
import string
```

```
import nltk
```

```
from nltk.corpus import stopwords
```

```
stopword=set(stopwords.words('english'))
```

```
stemmer = nltk.SnowballStemmer("english")
```

In []:

```
def clean(text):
```

```
    text = str(text).lower()
```

```
    text = re.sub('[\.*?\\]', "", text)
```

```
    text = re.sub('https?://\S+|www\.\S+', "", text)
```

```
    text = re.sub('<.*?>+', "", text)
```

```
    text = re.sub('[%s]' % re.escape(string.punctuation), "", text)
```

```
    text = re.sub('\n', "", text)
```

```
    text = re.sub('\w*\d\w*', "", text)
```

```

text = [word for word in text.split(' ') if word not in stopword]
text=" ".join(text)
text = [stemmer.stem(word) for word in text.split(' ')]
text=" ".join(text)
return text
data["title"] = data["title"].apply(clean)

x = np.array(data1["title"])
y = np.array(data1["label"])

x

y

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split

cv = CountVectorizer()
X = cv.fit_transform(x) # Fit the Data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=42)

from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier()
clf.fit(X_train,y_train)

pre=clf.predict(X_test)

from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,pre)*100)

input_word=input("ENTER THE SENTENCE:")

data = cv.transform([input_word]).toarray()
print(clf.predict(data))

```

In []:

In []:

In []:

In []:

In []:

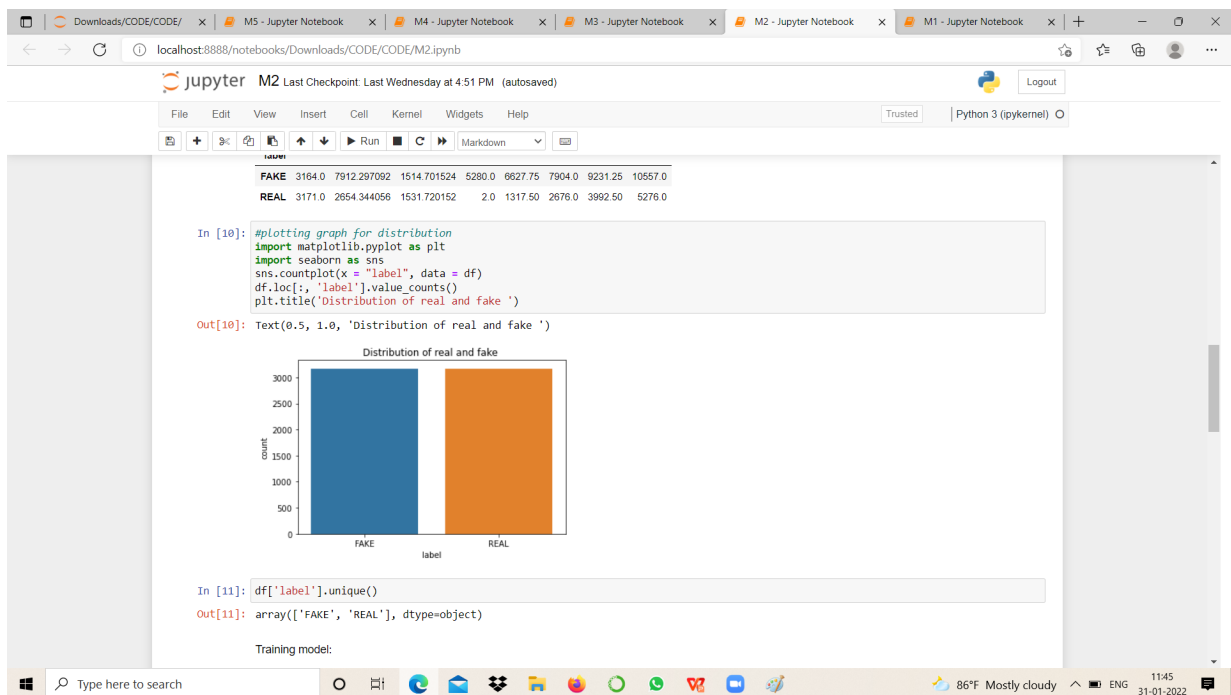
In []:

In []:

In []:

In []:

6.3.2. SCREENSHOTS



Downloads/CODE/CODE/ x M5 - Jupyter Notebook x M4 - Jupyter Notebook x M3 - Jupyter Notebook x M2 - Jupyter Notebook x M1 - Jupyter Notebook x + - □ ×

localhost:8888/notebooks/Downloads/CODE/CODE/M4.ipynb

jupyter M4 Last Checkpoint: Last Wednesday at 4:51 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

In [18]: from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
clf.fit(X_train, y_train)

Out[18]: LogisticRegression()

In [19]: pre=clf.predict(X_test)

In [20]: from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, pre)*100)

81.30081300813008

In [24]: input_word=input("ENTER THE SENTENCE:")
ENTER THE SENTENCE:Kaydee King (@KaydeeKing) November 9, 2016 T.

In [25]: data = cv.transform([input_word]).toarray()
print(clf.predict(data))

['FAKE']

In []:

Type here to search

79°F Mostly sunny 08:09 31-01-2022

6.3.3. PLAGARISM REPORT

ORIGINALITY REPORT

4

%

SIMILARITY INDEX 1%

STUDENT PAPERS

PRIMARY SOURCES

1

d.researchbib.com

Internet Source

2

%

1%

2

"Churn Prediction using Machine Learning-An Analytical CRM Application", International Journal of Innovative Technology and Exploring Engineering, 2020

Publication

3

Vasu Agarwal, H. Parveen Sultana, Srijan Malhotra, Amitrajit Sarkar. "Analysis of

Classifiers for Fake News Detection", Procedia Computer Science, 2019

Publication

1%

4

[Submitted to University of East London](#)

Student Paper

1

