Application of Machine Learning and Neural Networks to Fake News Detection

Summer Internship Project

Report 2

Sreyashi Bandyopadhyay

<sreyashiban14@gmail.com>

Supervised By: Soumotanu Mazumdar

Date: 8th May 2021

औद्योगिक प्रशिक्षण के लिए राष्ट्रीय संस्थान

National Institute for Industrial Training

One Premier Organization with Non Profit Status | Registered Under Govt. of WB

Empanelled Under Planning Commission Govt. of India

Inspired By: National Task Force on IT & SD Government of India

National Institute for Industrial Training- One Premier Organization with Non Profit Status Registered Under Govt.of West Bengal,Empanelled Under Planning Commission Govt.of India , Empanelled Under Central Social Welfare Board Govt. of India , Registered with National Career Services , Registered with National Employment Services.

STUDENT PROFILE

Name: Sreyashi Bandyopadhyay

College: Heritage Institute of Technology

Department –Applied Electronics and Instrumentation

Year of Study- 2nd year, 4th semester

Year of Passing- 2023

Internship Application Id- WB/HIT/965

Abstract

Fake news is false or misleading information presented as news. It often has the aim of damaging the reputation of a person or entity, or making money through advertising revenue. Sometimes these stories may be propaganda that is intentionally designed to mislead the reader, or may be designed as “clickbait” written for economic incentives (the writer profits on the number of people who click on the story). In recent years, fake news stories have proliferated via social media, in part because they are so easily and quickly shared online.

Some stories may include basic verifiable facts, but are written using language that is deliberately inflammatory, leaves out pertinent details or only presents one viewpoint. "Fake news" exists within a larger ecosystem of mis- and disinformation**.**

Misinformation is false or inaccurate information that is mistakenly or inadvertently created or spread; the intent is not to deceive. Disinformation is false information that is deliberately created and spread "in order to influence public opinion or obscure the truth

The technological ease of copying, pasting, clicking and sharing content online has helped these types of articles to proliferate. In some cases, the articles are designed to provoke an emotional response and placed on certain sites ("seeded") in order to entice readers into sharing them widely. In other cases, "fake news" articles may be generated and disseminated by "bots" - computer algorithms that are designed to act like people sharing information, but can do so quickly and automatically.

This project aims to rate reviews using traditional machine learning classifiers (Naïve Bayes) along with neural networks based on Deep Learning and compare which gives better and more accurate results.

Classification is a data mining methodology that assigns classes to a collection of data in order to help in more accurate predictions and analysis. The classification would be binary , that is the news would be classified as either real or fake.

The fake news dataset from Kaggle has been used for this task and can be found attached.

Acknowledgements

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I would like to express my heartfelt thanks and gratitude to my supervisor Mr. Soumotanu Mazumdar of National Institute for Industrial Training who gave me the golden opportunity to do this project and guided me in an exemplary manner. It helped me in doing a lot of research and I came to know about a lot of things related to this topic.

Last but not the least I thank my friends and my family for their wholehearted support.

1. Introduction

1.1 Motivation

False information is news, stories or hoaxes created to deliberately misinform or deceive readers. Usually, these stories are created to either influence people's views, push a political agenda or cause confusion and can often be a profitable business for online publishers. False information can deceive people by looking like trusted websites or using similar names and web addresses to reputable news organizations.

Fake news has become one of the biggest problems of our age. It has serious impact on our online as well as offline discourse. One can even go as far as saying that, to date, fake news poses a clear and present danger to western democracy and stability of the society.

In this project we aim to use classify news given by internet users and try to understand if the news presented is real or fake.

1.2 Dataset

The Kaggle fake news dataset has been used for this purpose of classification of fake news.

The dataset contains 6335 training examples collected from IMDB where each news is labeled as either fake or real. We categorized these ratings as either 1 (real) or 0 (fake) .

. Classical Machine Learning Models like Naïve Bayes and neural networks, namely LSTM and RNN were applied to check the model performance and create a comparison between them .

Attached below is an illustration of how the news snippets look in the dataset.

	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	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL
...	...	...	...	...
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
6331	8062	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE
6333	4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

6335 rows × 4 columns

1.3 Problems with the data

The main problem encountered while dealing with user movie reviews were the fact that it was not processed properly to be able to be fed into ML based algorithms. Feeding them directly to the algorithm would be detrimental to model performance.

Hence the preprocessing step was necessary to get the data ready to work with.

1.4 Data Preprocessing

The data preprocessing for this project can be divided into the following subparts.

1. Removing HTML tags.
2. Removing special characters not of importance to sentiment analysis ,e.g. (-*%/?!) etc
3. Removing punctuation marks. (. ,)
4. Converting every character to lowercase for ease of evaluation
5. Removing words of less importance that do not contribute to a sentiment for e.g.- I , me ,my , the ,is was etc.

6. Stemming - Stemming is a technique used to extract the base form of the words by removing affixes from them.

7. Tokenization of text - We use the method **word_tokenize**() to split a sentence into **words**. The output of **word tokenizer** in **NLTK can** be converted to Data Frame for better text understanding in machine learning applications.

2. About the Project

This project has been performed to automate the task of classifying a news article as real or fake so that a user can decide whether or not to believe it.

Firstly , the data has been thoroughly pre-processed as described in section 1.4 . After that a new file has been made with the formatted data. Next that data has been fed into machine learning algorithms and neural networks to compare and contrast the model performance .

The algorithms used are as follows -

1. Gaussian Naive Bayes
2. Decision Tree
3. RNN

2.1 Objectives

The sole objective of the project is to predict, using machine learning, the truth value of a news article. Different techniques have been applied for this purpose and a comparative study has been made between the different accuracies shown by the various models.

The steps involved are -

1. DATA MINING- Data mining is a process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics and database systems Data mining is an interdisciplinary subfield of computer science and statistics with an overall goal to extract information (with intelligent methods) from a data set and transform the information into a comprehensible structure for further use.

2. DATA CLEANING-Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted.

3. DATA PREPROCESSING- Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues.

4. EXPLORATORY DATA ANALYSIS - In statistics, exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods.

5. DIVIDING INTO TRAINING AND TESTING SET- Data splitting is the act of partitioning available data into two portions; usually for cross - validatory purposes. One portion of the data is used to develop a predictive model and the other to evaluate the model's performance.

6. APPLYING VARIOUS CLASSIFICATION MODELS- Various classification models were used to predict the probability of stroke in patients , after taking into consideration various other factors.

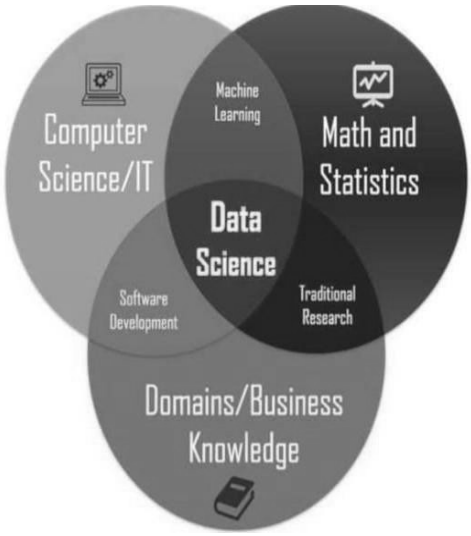
3. General Introduction to Relevant Topics

- **Data Science** - Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions. Data science uses complex machine learning algorithms to build predictive models.

The data used for analysis can be from multiple sources and present in various formats. Data science or data-driven science enables better decision making, predictive analysis, and pattern discovery.

Data science can add value to any business who can use their data well. From statistics and insights across workflows and hiring new candidates, to helping senior staff make better-informed decisions, data science is valuable to any company in any industry.

By extrapolating and sharing these insights, data scientists help organizations to solve vexing problems. Combining computer science, modeling, statistics, analytics, and math skills— along with sound business sense— data scientists uncover the answers to major questions that help organizations make objective decisions.



- Machine Learning - Machine learning is a type of technology that aims to learn from experience. For example, as a human, you can learn how to play chess simply by observing other people playing chess. In the same way, computers are programmed by providing them with data from which they learn and are then able to predict future elements or conditions.

There are various steps involved in machine learning:

1. collection of data
2. filtering of data
3. analysis of data
4. algorithm training
5. testing of the algorithm
6. using the algorithm for future predictions

Machine learning uses different kinds of algorithms to find patterns, and these algorithms are classified into two groups:

- supervised learning
- unsupervised learning

Supervised Learning

Supervised learning is the science of training a computer to recognize elements by giving it sample data. The computer then learns from it and is able to predict future datasets based on the learned data.

For example, you can train a computer to filter out spam messages based on past information.

Supervised learning has been used in many applications, e.g. Facebook , to search images based on a certain description. You can now search images on Facebook with words that describe the contents of the photo. Since the social networking site already has a database of captioned images, it is able to search and match the description to features from photos with some degree of accuracy.

There are only two steps involved in supervised learning:

- training

- testing

Some of the supervised learning algorithms include:

- decision trees
- support vector machines
- naive Bayes
- k-nearest neighbor
- linear regression

Python –

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991.

Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object- oriented , imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Two major versions of Python are currently in active use:

Python 3.x is the current version and is under active development. Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

3.1 Imports

The libraries that have been imported for this project are stated as follows.

1. Numpy- NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

2. Seaborn - Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

3. Pandas- In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

.

4. TensorFlow -TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow is a symbolic math library based on dataflow and differentiable programming.

5. Keras- Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.

6. Scikit learn- The Sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering .

7. NLTK - NLTK (Natural Language Toolkit) is a suite that contains libraries and programs for statistical language processing. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language and reply to it with an appropriate response.

8. Gensim is an open-source library for unsupervised topic modeling and natural language processing, using modern statistical machine learning.

3.2 Programming Language

Advantages of Python

1. Easy Syntax
2. Readability
3. High-Level Language
4. Object-oriented programming
5. It's Open source and Free
6. Cross-platform
7. Widely Supported
8. It's Safe
9. Extensible

Easy Syntax of Python

Python's syntax is easy to learn, so both non- programmers and programmers can start programming right away**.**

Very Clear Readability of Python

Python's syntax is very clear, so it is easy to understand program code. (Python is often referred to as "executable pseudo-code" because its syntax mostly follows the conventions used by programmers to outline their ideas without the formal verbosity of code in most programming languages.

Python High-Level Language

Python looks more like a readable, human language than like a low-level language. This gives you the ability to program at a faster rate than a low-level language will allow you.

Python Is Open-Source and Free

Python is both free and open-source. The Python Software Foundation distributes pre -made binaries that are freely available for use on all major operating systems called CPython. You can get CPython's source-code, too. Plus, we can modify the source code and distribute as allowed by CPython's license.

Python is a Cross-platform

Python runs on all major operating systems like Microsoft Windows, Linux, and Mac OS X.

Python Object-oriented programming

Object-oriented programming allows you to create data structures that can be reused, which reduces the amount of repetitive work that you'll need to do.

Programming languages usually define objects with namespaces, like class or def, and objects can edit themselves by using keyword, like this or self.

Most modern programming languages are object-oriented (such as Java, C++, and C#) or have support for OOP features (such as Perl version 5 and later). Additionally, object-oriented techniques can be used in the design of almost any non-trivial software and implemented in almost any programming or scripting language.

Python Widely Supported Programming Language

Python has an active support community with many websites, mailing lists, and USENET "net news" groups that attract a large number of knowledgeable and helpful contributors.

Python is a Safe

Python doesn't have pointers like other C-based languages, making it much more reliable. Along with that, errors never pass silently unless they're explicitly silenced. This allows you to see and read why the program crashed and where to correct your error**.**

Python Batteries Included Language

Python is famous for being the "batteries are included" language. There are over 300 standard library modules which contain modules and classes for a wide variety of programming tasks.

For example, the standard library contains modules for safely creating temporary files (named or anonymous), mapping files into memory (including use of shared and anonymous memory mappings), spawning and controlling sub-processes, compressing and decompressing files (compatible with gzip or PK-zip) and archives files (such as Unix/Linux "tar").

Accessing indexed "DBM" (database) files, interfacing to various graphical user interfaces (such as the TK toolkit and the popular WxWindows multi-platform windowing system), parsing and maintaining CSV (comma- separated values) and ".cfg" or ".ini" configuration files (similar in syntax to the venerable WIN.INI files from MS-DOS and MS-Windows), for sending e-mail, fetching and parsing web pages, etc. It's possible, for example, to create a custom web server in Python using less than a dozen lines of code, and one of the standard libraries, of course.

Python is Extensible

In addition to the standard libraries there are extensive collections of freely available add-on modules, libraries, frameworks, and tool-kits. These generally conform to similar standards and conventions. For example, almost all of the database adapters (to talk to almost any client-server RDBMS engine such as MySQL, Postgres, Oracle, etc) conform to the Python DBAPI and thus can mostly be accessed using the same code. So it's usually easy to modify a Python program to support any database engine.

.

3.2 Future Scopes of Python

Python is one of the fastest growing languages and has undergone a successful span of more than 25 years as far as its adoption is concerned. This success also reveals a promising future scope of python programming language.
In fact, it has been continuously serving as the best programming language for application development, web development, game development, system administration, scientific and numeric computing, GIS and Mapping etc.

Popularity of python

The reason behind the immense popularity of python programming language across the globe is the features it provides. Have a look at the features of python language.

(1) Python Supports Multiple Programming Paradigms Python is a multi-paradigm programming language including features such as object-oriented, imperative, procedural, functional, reflective etc.

(2) Python Has Large Set Of Library and Tools

Python has very extensive standard libraries and tools that enhance the overall functionality of python language and also helps python programmers to easily write codes. Some of the important python libraries and tools are listed below.

- Built-in functions, constants, types, and exceptions.
- File formats, file and directory access, multimedia services.
- GUI development tools such as Tkinter
- Custom Python Interpreters, Internet protocols and support, data compression and archiving, modules etc.
- Scrappy, wxPython, SciPy, matplotlib, Pygame, PyQT, PyGTK etc.

(3) Python Has a Vast Community Support

This is what makes python a favorable choice for development purposes. If you are having problems writing python a program, you can post directly to python community and will get the response with the solution of your problem. You will also find many new ideas regarding python technology and change in the versions.

(4) Python is Designed For Better Code Readability

Python provides a much better code readability as compared to another programming language. For example, it uses whitespace indentation in place of curly brackets for delimiting the block of codes. Isn't it awesome?

(5)Python Has a Vast Community Support

This is what makes python a favorable choice for development purposes. If you are having problems writing python a program, you can post directly to python community and will get the response with the solution of your problem. You will also find many new ideas regarding python technology and change in the versions.

(6)Python is Designed For Better Code Readability

Python provides a much better code readability as compared to another programming language. For example, it uses whitespace indentation in place of curly brackets for delimiting the block of codes.

(7)Python Contains Fewer Lines Of Codes

Codes are written in python programming language complete in fewer lines thus reducing the efforts of programmers. Let's have a look on the following "Hello World" program written in C, C++, Java, and Python.

While, C, C++, and Java take six, seven and five lines respectively for a simple "Hello World" program. Python takes only a single line which means, less coding effort and time is required for writing the same program.

Future Technologies Counting On Python

Generally, we have seen that python programming language is extensively used for web development, application development, system administration, developing games etc.

But do you know there are some future technologies that are relying on python? As a matter of fact, Python has become the core language as far as the success of these technologies is concerned. Let's dive into the technologies which use python as a core element for research, production and further developments.

(1) Artificial Intelligence (AI)

Python programming language is undoubtedly dominating the other languages when future technologies like Artificial Intelligence (AI) come into the play.

There are plenty of python frameworks, libraries, and tools that are specifically developed to direct Artificial Intelligence to reduce human efforts with increased accuracy and efficiency for various development purposes.

It is only the Artificial Intelligence that has made it possible to develop speech recognition system, autonomous cars, interpreting data like images, videos etc.

We have shown below some of the python libraries and tools used in various Artificial Intelligence branches.

- Machine Learning- PyML, PyBrain, scikit-learn, MDP Toolkit, GraphLab Create, MIPy etc.
- General AI- pyDatalog, AIMA, EasyAI, SimpleAI etc.
- Neural Networks- PyAnn, pyrenn, ffnet, neurolab etc.
- Natural Language & Text Processing- Quepy, NLTK, gensim

(2) Big Data

The future scope of python programming language can also be predicted by the way it has helped big data technology to grow. Python has been successfully contributing in analyzing a large number of data sets across computer clusters through its high- performance toolkits and libraries.

Let's have a look at the python libraries and toolkits used for Data analysis and handling other big data issues.

- Pandas
- Scikit-Learn
- NumPy
- SciPy
- GraphLab Create
- IPython
- Bokeh
- Agate
- PySpark
- Dask

(3)Networking

Networking is another field in which python has a brighter scope in the future. Python programming language is used to read, write and configure routers and switches and perform other networking automation tasks in a cost-effective and secure manner.

For these purposes, there are many libraries and tools that are built on the top of the python language. Here we have listed some of these python libraries and tools especially used by network engineers for network automation.

- Ansible
- Netmiko
- NAPALM(Network Automation and Programmability Abstraction Layer with Multivendor Support)
- Pyeapi
- Junos PyEZ
- PySNMP
- Paramiko SSH

Real-Life Python Success Stories

Python has seemingly contributed as a core language for increasing productivity regarding various development purposes at many of the IT organizations. We have shown below some of the real-life python success stories.

- Australia's RMA Department D-Link has successfully implemented python for creating DSL Firmware Recovery System.
- Python has helped Gusto.com, an online travel site, in reducing development costs and time.
- ForecastWatch.com also uses python in rating the accuracy of weather forecast reports provided by companies such as Accuweather, MyForecast.com and The Weather Channel.
- Python has also benefited many product development companies such as Acqutek, AstraZeneca, GravityZoo, Carmanah Technologies Inc. etc in creating autonomous devices and software.
- Test&Go uses python scripts for Data Validation.
- Industrial Light & Magic(ILM) also uses python for batch processing that includes modeling, rendering and compositing thousands of picture frames per day.

There is a huge list of success stories of many organizations across the globe which are using python for various purposes such as software development, data mining, unit testing, product development, web development, data validation, data visualization etc.

These success stories directly point towards a promising future scope of python programming language.

3.3Top Competitors Of Python

The future scope of python programming language also depends on its competitors in the IT market. But, due to the fact that it has become a core language for future technologies such as artificial intelligence, big data, etc., it will surely rise further and will be able to beat its competitors.

Competitors and Alternatives to Python Programming Language

- Microsoft.
- Oracle.
- IBM.
- Tableau.
- SAP.
- Alteryx.
- Blue Yonder.
- Gurobi.

4. Hardware and Software Requirements

Hardware Requirements

- Speed: 233MHz and above

- Hard Disk 10 GB
- RAM: 256 MB

Software Requirements

- Operating System: Windows/Linux Front End: Python 3.7
- Platform: Anaconda
- RAM: 256 MB

5. Formal description of the Training Models used

This section aims to formally define the various training models used in this study.

5.1 Naïve Bayes model - Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Now, before moving to the formula for Naive Bayes, it is important to know about Bayes' theorem.

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

- Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.
- P(A) is the priori of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
- P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

What are the Pros and Cons of Naive Bayes?

Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.
- Another limitation of Naïve Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Applications of Naïve Bayes :

- Real time Prediction: Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- Multi class Prediction: This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- Text classification/ Spam Filtering/ Sentiment Analysis: Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- Recommendation System: Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

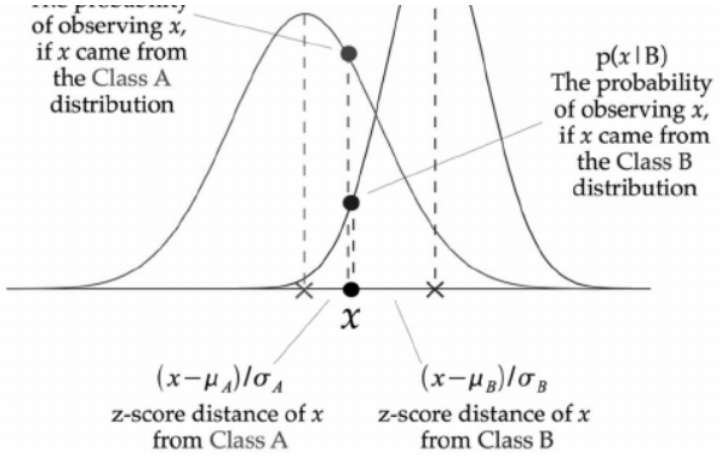
Different Models of Naïve Bayes used –

- Gaussian Naïve Bayes - is a variant of Naive Bayes that follows Gaussian normal distribution and supports continuous data. The likelihood of the features is assumed to be-

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution.

An approach to create a simple model is to assume that the data is described by a Gaussian distribution with no co-variance (independent dimensions) between dimensions. This model can be fit by simply finding the mean and standard deviation of the points within each label, which is all what is needed to define such a distribution.



The above illustration indicates how a Gaussian Naive Bayes (GNB) classifier works. At every data point, the z-score distance between that point and each class-mean is calculated, namely the distance from the class mean divided by the standard deviation of that class.

Thus, we see that the Gaussian Naive Bayes has a slightly different approach and can be used efficiently.

- Multinomial Naïve Bayes - Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output.

- Bernoulli Naïve Bayes - This is used for discrete data and it works on Bernoulli distribution. The main feature of Bernoulli Naive Bayes is that it accepts features only as binary values like true or false, yes or no, success or failure, 0 or 1 and so on. So when the feature values are binary we know that we have to use Bernoulli Naive Bayes classifier.

The Bernoulli distribution

$$p(x) = P[X = x] = \begin{cases} q = 1 - p & x = 0 \\ p & x = 1 \end{cases}$$

Bernoulli Naive Bayes Classifier is based on the following rule:

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

5.2 Recurrent Neural Network

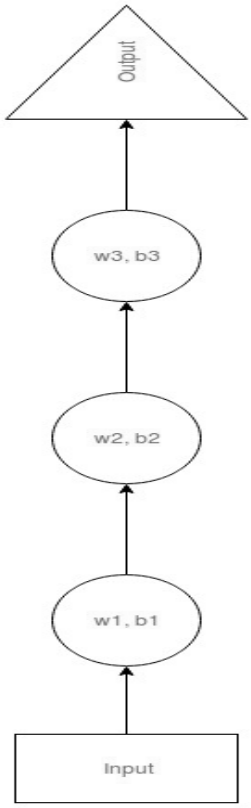
Recurrent Neural Network(RNN) are a type of Neural Network where the output from previous step are fed as input to the current step**.** In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

RNN have a **"memory"** which remembers all information about what has been calculated. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.

- RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous outputs by giving each output as input to the next hidden layer.

- Hence these three layers can be joined together such that the weights and bias of all the hidden layers is the same, into a single recurrent layer.

Suppose there is a deeper network with one input layer, three hidden layers and one output layer. Then like other neural networks, each hidden layer will have its own set of weights and biases, let's say, for hidden layer 1 the weights and biases are (w1, b1), (w2, b2) for second hidden layer and (w3, b3) for third hidden layer. This means that each of these layers are independent of each other, i.e. they do not memorize the previous outputs.



Now the RNN will do the following:

- RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous outputs by giving each output as input to the next hidden layer.
- Hence these three layers can be joined together such that the weights and bias of all the hidden layers is the same, into a single recurrent layer.

Training through RNN

1. A single time step of the input is provided to the network.
2. Then calculate its current state using set of current input and the previous state.
3. The current ht becomes ht-1 for the next time step.
4. One can go as many time steps according to the problem and join the information from all the previous states.
5. Once all the time steps are completed the final current state is used to calculate the output.
6. The output is then compared to the actual output i.e the target output and the error is generated.
7. The error is then back-propagated to the network to update the weights and hence the network (RNN) is trained.

Advantages of Recurrent Neural Network

1. An RNN remembers each and every information through time. It is useful in time series prediction only because of the feature to remember previous inputs as well. This is called Long Short Term Memory.
2. Recurrent neural network are even used with convolutional layers to extend the effective pixel neighborhood.

Disadvantages of Recurrent Neural Network

1. Gradient vanishing and exploding problems.
2. Training an RNN is a very difficult task.
3. It cannot process very long sequences if using tanh or relu as an activation function.

5.4.1 LSTMs

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work.[1] They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.

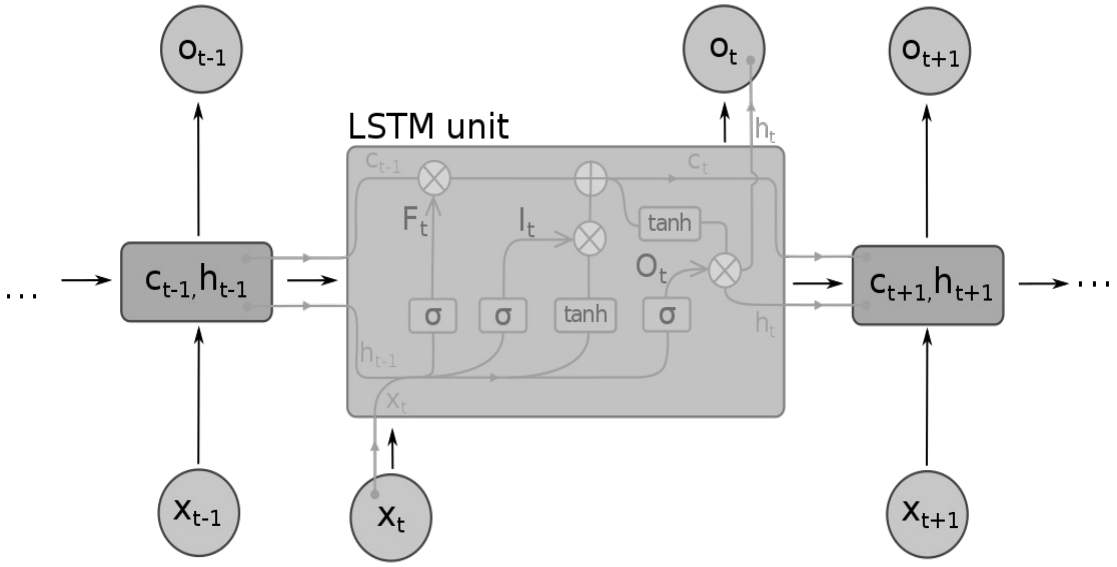
The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.

The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a point wise multiplication operation.

The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!”

An LSTM has three of these gates, to protect and control the cell state.



5.3 Decision Trees

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal, its also widely used in machine learning

6. The TF-IDF Vectorizer

TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.

It has many uses, most importantly in automated text analysis, and is very useful for scoring words in machine learning algorithms for Natural Language Processing (NLP).
TF-IDF (term frequency-inverse document frequency) was invented for document search and information retrieval. It works by increasing proportionally to the number of times a word appears in a document, but is offset by the number of documents that contain the word. So, words that are common in every document, such as this, what, and if, rank low even though they may appear many times, since they don't mean much to that document in particular.

TF-IDF for a word in a document is calculated by multiplying two different metrics:

- The **term frequency** of a word in a document. There are several ways of calculating this frequency, with the simplest being a raw count of instances a word appears in a document. Then, there are ways to adjust the frequency, by length of a document, or by the raw frequency of the most frequent word in a document.
- The **inverse document frequency** of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.
- So, if the word is very common and appears in many documents, this number will approach 0. Otherwise, it will approach 1.

Multiplying these two numbers results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that particular document.

To put it in more formal mathematical terms, the TF-IDF score for the word t in the document d from the document set D is calculated as follows:

$$tf\,idf\,(t, d, D) = tf\,(t, d)\,.\,idf\,(t, D)$$

Where:

$$tf\,(t, d) = log\,(1 + freq\,(t, d))$$

$$idf(t, D) = \log\left(\frac{N}{count(d \in D : t \in d)}\right)$$

7.Optimizers

Optimizers are algorithms or methods used to change the attributes of the **neural network** such as weights and learning rate to reduce the losses. **Optimizers** are used to solve optimization problems by minimizing the function.

Gradient Descent

Gradient Descent is the most basic but most used optimization algorithm. It's used heavily in linear regression and classification algorithms. Backpropagation in neural networks also uses a gradient descent algorithm.

Gradient descent is a first-order optimization algorithm which is dependent on the first order derivative of a loss function. It calculates that which way the weights should be altered so that the function can reach a minima. Through backpropagation, the loss is transferred from one layer to another and the model's parameters also known as weights are modified depending on the losses so that the loss can be minimized.

Advantages:

1. Easy computation.
2. Easy to implement.
3. Easy to understand.

Disadvantages:

1. May trap at local minima.
2. Weights are changed after calculating gradient on the whole dataset. So, if the dataset is too large than this may take years to converge to the minima.
3. Requires large memory to calculate gradient on the whole dataset.

Stochastic Gradient Descent

It's a variant of Gradient Descent. It tries to update the model's parameters more frequently. In this, the model parameters are altered after computation of loss on each training example. So, if the dataset contains 1000 rows SGD will update the model parameters 1000 times in one cycle of dataset instead of one time as in Gradient Descent.

As the model parameters are frequently updated parameters have high variance and fluctuations in loss functions at different intensities.

Advantages:

1. Frequent updates of model parameters hence, converges in less time.
2. Requires less memory as no need to store values of loss functions.
3. May get new minima's.

Disadvantages:

1. High variance in model parameters.
2. May shoot even after achieving global minima.
3. To get the same convergence as gradient descent needs to slowly reduce the value of learning rate.

Mini-Batch Gradient Descent

It's best among all the variations of gradient descent algorithms. It is an improvement on both SGD and standard gradient descent. It updates the model parameters after every batch. So, the dataset is divided into various batches and after every batch, the parameters are updated.

Advantages:

1. Frequently updates the model parameters and also has less variance.
2. Requires medium amount of memory.

All types of Gradient Descent have some challenges:

1. Choosing an optimum value of the learning rate. If the learning rate is too small than gradient descent may take ages to converge.
2. Have a constant learning rate for all the parameters. There may be some parameters which we may not want to change at the same rate.
3. May get trapped at local minima.

Momentum

Momentum was invented for reducing high variance in SGD and softens the convergence. It accelerates the convergence towards the relevant direction and reduces the fluctuation to the irrelevant direction. One more hyperparameter is used in this method known as momentum symbolized by '**γ**'.

$V(t)=\gamma V(t-1)+\alpha.\nabla J(\theta)$

Now, the weights are updated by **$\theta=\theta-V(t)$.**

The momentum term **γ** is usually set to 0.9 or a similar value.

Advantages:

1. Reduces the oscillations and high variance of the parameters.
2. Converges faster than gradient descent.

Disadvantages:

1. One more hyper-parameter is added which needs to be selected manually and accurately.

Nesterov Accelerated Gradient

Momentum may be a good method but if the momentum is too high the algorithm may miss the local minima and may continue to rise up. So, to resolve this issue the NAG algorithm was developed. It is a look ahead method. We know we'll be using **$\gamma V(t-1)$** for modifying the weights so, **$\theta-\gamma V(t-1)$** approximately tells us the future location. Now, we'll calculate the cost based on this future parameter rather than the current one.

$V(t)=\gamma V(t-1)+\alpha. \nabla J(\theta-\gamma V(t-1))$ and then update the parameters using **$\theta=\theta-V(t)$.**

Adagrad

One of the disadvantages of all the optimizers explained is that the learning rate is constant for all parameters and for each cycle. This optimizer changes the learning rate. It changes the learning rate **'η'** for each parameter and at every time step **'t'.** It's a type second order optimization algorithm. It works on the derivative of an error function.

Advantages:

1. Learning rate changes for each training parameter.
2. Don't need to manually tune the learning rate.
3. Able to train on sparse data.

Disadvantages:

1. Computationally expensive as a need to calculate the second order derivative.

2. The learning rate is always decreasing results in slow training.

AdaDelta

It is an extension of **AdaGrad** which tends to remove the *decaying learning Rate* problem of it. Instead of accumulating all previously squared gradients, **Adadelta** limits the window of accumulated past gradients to some fixed size **w**. In this exponentially moving average is used rather than the sum of all the gradients.

Advantages:

1. Now the learning rate does not decay and the training does not stop.

Disadvantages:

1. Computationally expensive.

Adam

Adam (Adaptive Moment Estimation) works with momentums of first and second order. The intuition behind the Adam is that we don't want to roll so fast just because we can jump over the minimum, we want to decrease the velocity a little bit for a careful search.

Advantages:

1. The method is too fast and converges rapidly.
2. Rectifies vanishing learning rate, high variance.

Disadvantages:

1. Computationally costly.

8. Loss Functions

Loss functions play an important role in any statistical model - they define an objective which the performance of the model is evaluated against and the parameters learned by the model are determined by minimizing a chosen loss function.

Loss functions define what a good prediction is and isn't. In short, choosing the right loss function dictates how well your estimator will be. This article will probe into loss functions, the role they play in validating predictions, and the various loss functions used.

Loss functions for classification

Classification problems involve predicting a discrete class output. It involves dividing the dataset into different and unique classes based on different parameters so that a new and unseen record can be put into one of the classes.

A mail can be classified as a spam or not a spam and a person's dietary preferences can be put in one of three categories - vegetarian, non-vegetarian and vegan. Let's take a look at loss functions that can be used for classification problems.

Binary Cross Entropy Loss

This is the most common loss function used for classification problems that have two classes. The word "entropy", seemingly out-of-place, has a statistical interpretation.

Entropy is the measure of randomness in the information being processed, and cross entropy is a measure of the difference of the randomness between two random variables.

If the divergence of the predicted probability from the actual label increases, the cross-entropy loss increases. Going by this, predicting a probability of .011 when the actual observation label is 1 would result in a high loss value. In an ideal situation, a "perfect" model would have a log loss of 0.

Categorical Cross Entropy Loss

Categorical Cross Entropy loss is essentially Binary Cross Entropy Loss expanded to multiple classes. One requirement when categorical cross entropy loss function is used is that the labels should be one-hot encoded.

This way, only one element will be non-zero as other elements in the vector would be multiplied by zero.

Hinge Loss

Another commonly used loss function for classification is the hinge loss. Hinge loss is primarily developed for support vector machines for calculating the maximum margin from the hyperplane to the classes.

Loss functions penalize wrong predictions and does not do so for the right predictions. So, the score of the target label should be greater than the sum of all the incorrect labels by a margin of (at the least) one.

This margin is the maximum margin from the hyperplane to the data points, which is why hinge loss is preferred for SVMs.

9. Activation Functions

An activation function in a neural network defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network.

Activation functions are a critical part of the design of a neural network.

The choice of activation function in the hidden layer will control how well the network model learns the training dataset. The choice of activation function in the output layer will define the type of predictions the model can make.

Technically, the activation function is used within or after the internal processing of each node in the network, although networks are designed to use the same activation function for all nodes in a layer.

A network may have three types of layers: input layers that take raw input from the domain, hidden layers that take input from another layer and pass output to another layer, and output layers that make a prediction.

All hidden layers typically use the same activation function. The output layer will typically use a different activation function from the hidden layers and is dependent upon the type of prediction required by the model.

Most commonly used activation functions are as follows –

- Rectified Linear Activation (**ReLU**)
- Logistic (**Sigmoid**)
- Hyperbolic Tangent (**Tanh**)

ReLU Activation Function

The rectified linear activation function, or ReLU activation function, is perhaps the most common function used for hidden layers.

It is common because it is both simple to implement and effective at overcoming the limitations of other previously popular activation functions, such as Sigmoid and Tanh. Specifically, it is less susceptible to vanishing gradients that prevent deep models from being trained, although it can suffer from other problems like saturated or "*dead*" units.

The ReLU function is calculated as follows:

- max(0.0, x)

This means that if the input value (x) is negative, then a value 0.0 is returned, otherwise, the value is returned.

Sigmoid Activation Function

The sigmoid activation function is also called the logistic function.

It is the same function used in the logistic regression classification algorithm.

The function takes any real value as input and outputs values in the range 0 to 1. The larger the input (more positive), the closer the output value will be to 1.0, whereas the smaller the input (more negative), the closer the output will be to 0.0.

The sigmoid activation function is calculated as follows:

- 1.0 / (1.0 + e^-x)

Where e is a mathematical constant, which is the base of the natural logarithm.

Tanh Activation Function

The hyperbolic tangent activation function is also referred to simply as the Tanh (also "*tanh*" and "*TanH*") function.

It is very similar to the sigmoid activation function and even has the same S-shape.

The function takes any real value as input and outputs values in the range -1 to 1. The larger the input (more positive), the closer the output value will be to 1.0, whereas the smaller the input (more negative), the closer the output will be to -1.0.

The Tanh activation function is calculated as follows:

- (e^x – e^-x) / (e^x + e^-x)

Where e is a mathematical constant that is the base of the natural logarithm.

10. Source Code Snippets

1. Data preprocessing

```
import numpy as np
import pandas as pd
import regex as re

!pip3 install nltk

import nltk

df=pd.read_csv('/content/fakenewsdataset.zip')

df

df = pd.get_dummies(df, columns=['label'], drop_first=True)

df

df.drop('title' , axis=1 , inplace=True)


def regex_operations(string):
    string = re.sub(r"[^\w(),|!?\'\`\:\-\.;\$%#]", " ", string)
    string = re.sub(r"\'s", " is", string)
    string = re.sub(r"\'ve", " have", string)
    string = re.sub(r"n\'t", " not", string)
    string = re.sub(r"\'re", " are", string)
    string = re.sub(r"\'d", " would", string)
    string = re.sub(r"\'ll", " will", string)
    string = re.sub(r"(?<=\w)\.\.\.", " ... ", string)
    string = re.sub(r"(?<=\w)\.", " . ", string)
    string = re.sub(r"(?<=\w),", " , ", string)
    string = re.sub(r"(?<=\w);", " ; ", string)
    string = re.sub(r"(?<=\w)!", " ! ", string)
    string = re.sub(r"\((?=\w)", " ( ", string)
    string = re.sub(r"(?<=\w)\)", " ) ", string)
    string = re.sub(r"(?<=\w)\?", " ? ", string)
    string = re.sub(r"\s{2,}", " ", string)
    string = re.sub(r'\W+,. ', ' ', string)
    string =  re.sub(r'<.*?>' ,' ' ,string)
    string = re.sub(r'[^\w\s]', " ",string)
    string = re.sub('[0-9\n]',' ',string)
    return string.strip()

df['text']=df['text'].apply(regex_operations)

df['text'][1]

def convert_to_lower(org):
  cleantext=org.lower()
  return cleantext
df['text']=df['text'].apply(convert_to_lower)

df['text'][1]

from nltk.stem.porter import PorterStemmer
stemming_model=PorterStemmer()

def stem_words(org):
  cleantext=[]
  for word in org:
    cleantext.append(stemming_model.stem(word))
  new=cleantext[:]
  cleantext.clear()
  return new
df['text']=df['text'].apply(stem_words)

def join_back(org):
  return "".join(org)

df['text']=df['text'].apply(join_back)

df.head()

df.to_csv('preprocessed_fakenews.csv')



sentences = df['text'].tolist()

sentences
```

2. Using TF-IDF Vectorizer

```
90
91  from sklearn.feature_extraction.text import TfidfVectorizer
92  tfidf_v=TfidfVectorizer(max_features=5000,ngram_range=(1,3))
93  x=tfidf_v.fit_transform(sentences).toarray()
94
95  x.shape
96
97  y=df['label_REAL']
98
99  y
100
101 tfidf_v.get_feature_names()[:20]
102
103 tfidf_v.get_params()
104
105 from sklearn.model_selection import train_test_split
106 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=2021)
107
108 count_df = pd.DataFrame(x_train, columns=tfidf_v.get_feature_names())
109
110 count_df.head()
```

3. Implementing Naïve Bayes model

```
111
112 from sklearn.naive_bayes import GaussianNB
113 gnb_model=GaussianNB()
114
115 gnb_model.fit(x_train, y_train)
116 gnb_pred = gnb_model.predict(x_test)
117
118 from sklearn.metrics import classification_report
119
120 print(classification_report(y_test,gnb_pred))
121
```

4. Implementing the decision tree model

```
121
122 from sklearn.tree import DecisionTreeClassifier
123
124 dt_model=DecisionTreeClassifier()
125
126 dt_model.fit(x_train, y_train)
127 dt_pred = dt_model.predict(x_test)
128
129 print(classification_report(y_test,dt_pred))
```

5. Implementing the RNN model

In [ ]:
```
!pip install plotly
!pip install nltk
!pip install spacy # For Advanced Natural Language Processing
!pip install gensim # For Unsupervised Topic Modeling And Natural Language Processing
import nltk
nltk.download('punkt')
```

```
Requirement already satisfied: plotly in /usr/local/lib/python3.7/dist-packages (4.4.1)
Requirement already satisfied: retrying>=1.3.3 in /usr/local/lib/python3.7/dist-packages (from plotly) (1.3.3)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from plotly) (1.15.0)
Requirement already satisfied: nltk in /usr/local/lib/python3.7/dist-packages (3.2.5)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from nltk) (1.15.0)
Requirement already satisfied: spacy in /usr/local/lib/python3.7/dist-packages (2.2.4)
Requirement already satisfied: thinc==7.4.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (7.4.0)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (1.19.5)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy) (2.0.5)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (0.8.2)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (4.41.1)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (2.23.0)
Requirement already satisfied: blis<0.5.0,>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (0.4.1)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy) (3.0.5)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in /usr/local/lib/python3.7/dist-packages (from spacy) (1.0.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from spacy) (56.1.0)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in /usr/local/lib/python3.7/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests<3.0.0,>=
2.13.0->spacy) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2020.1
2.5)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2.10)
Requirement already satisfied: importlib-metadata>=0.20; python_version < "3.8" in /usr/local/lib/python3.7/dist-packages (from catalogue
<1.1.0,>=0.0.7->spacy) (4.0.1)
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8" in /usr/local/lib/python3.7/dist-packages (from importlib
-metadata>=0.20; python_version < "3.8"->catalogue<1.1.0,>=0.0.7->spacy) (3.7.4.3)
```

In [ ]:
```
import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
from wordcloud import WordCloud, STOPWORDS
import nltk
import re
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
```

In [ ]:
```
import gensim
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import STOPWORDS
from tensorflow.keras.preprocessing.text import one_hot, Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Embedding, Input, LSTM, Conv1D, MaxPool1D, Bidirectional
from tensorflow.keras.models import Model
```

In [ ]:
```
df=pd.read_csv('/content/fakenewsdataset.zip')
```

In [ ]:
```
nltk.download("stopwords")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[ ]:
```
True
```

In [ ]:
```
df.head()
```

Out[ ]:

	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

In [ ]:
```
df.head()
```

Out[ ]:

	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	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

In [ ]:
```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])
```

In [ ]:
```
def preprocess(text):
    result = []
    for token in gensim.utils.simple_preprocess(text):
        if token not in gensim.parsing.preprocessing.STOPWORDS and len(token) > 3 and token not in stop_words:
            result.append(token)

    return result
```

In [ ]:
```
df['clean'] = df['text'].apply(preprocess)
```

In [ ]:
```
df
```

Out[ ]:

	Unnamed: 0	title	text	label	clean
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	[daniel, greenfield, shillman, journalism, fel...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol	Google Pinterest Digg Linkedin Reddit Stumbleu	FAKE	[google, pinterest, digg, linkedin, reddit, st

In [ ]:
```
df['clean'] = df['text'].apply(preprocess)
```

In [ ]:
```
df
```

Out[ ]:

	Unnamed: 0	title	text	label	clean
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	[daniel, greenfield, shillman, journalism, fel...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	[google, pinterest, digg, linkedin, reddit, st...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	[secretary, state, john, kerry, said, monday, ...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	[kaydee, king, kaydeeking, november, lesson, t...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	[primary, york, runners, hillary, clinton, don...
...	...	...	...	...	...
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL	[state, department, told, republican, national...
6331	8062	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE	[stand, plutocratic, pentagon, posted, wikimed...
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE	[anti, trump, protesters, tools, oligarchy, re...
6333	4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL	[addis, ababa, ethiopia, president, obama, con...
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL	[bush, suddenly, attacking, trump, matters, bu...

6335 rows × 5 columns

In [ ]:
```
list_of_words = []
for i in df['clean']:
    for j in i:
        list_of_words.append(j)
```

In [ ]:
```
list_of_words
```

Out[ ]:
```
['daniel',
 'greenfield',
 'shillman',
 'journalism',
 'fellow',
 'freedom',
 'center',
 'york',
 'writer',
 'focusing',
 'radical',
 'islam',
 'final',
 'stretch',
 'election',
 'hillary',
 'rodham',
 'clinton',
 'gone',
 'word',
 'unprecedented',
 'thrown',
 'election',
 'ought',
 'retired',
 'unprecedented',
 'nominee',
 'major',
 'political',
 'party',
 'exactly',
 'hillary',
```

In [ ]:
```
len(list_of_words)
```

Out[ ]:
```
2304425
```

In [ ]:
```
total_words = len(list(set(list_of_words)))
total_words
```

Out[ ]:
```
62148
```

In [ ]:
```
df['clean_joined'] = df['clean'].apply(lambda x: " ".join(x))
```

In [ ]:
```
df
```

Out[ ]:

	Unnamed: 0	title	text	label	clean	clean_joined
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	[daniel, greenfield, shillman, journalism, fel...	daniel greenfield shillman journalism fellow f...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	[google, pinterest, digg, linkedin, reddit, st...	google pinterest digg linkedin reddit stumbleu...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	[secretary, state, john, kerry, said, monday, ...	secretary state john kerry said monday stop pa...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	[kaydee, king, kaydeeking, november, lesson, t...	kaydee king kaydeeking november lesson tonight...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	[primary, york, runners, hillary, clinton, don...	primary york runners hillary clinton donald tr...
...	...	...	...	...	...	...
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL	[state, department, told, republican, national...	state department told republican national comm...
		The 'P' in PBS Should Stand for	The 'P' in PBS Should Stand for		[stand, plutocratic, pentagon,	stand plutocratic pentagon

In [ ]:
```
maxlen = -1
for doc in df.clean_joined:
    tokens = nltk.word_tokenize(doc)
    if(maxlen<len(tokens)):
        maxlen = len(tokens)
print( maxlen)
```

```
7967
```

In [ ]:
```
df = pd.get_dummies(df, columns=['label'], drop_first=True)
```

In [ ]:
```
df
```

Out[ ]:

	Unnamed: 0	title	text	clean	clean_joined	label_REAL
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	[daniel, greenfield, shillman, journalism, fel...	daniel greenfield shillman journalism fellow f...	0
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	google pinterest digg linkedin reddit stumbleu...	0
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	[secretary, state, john, kerry, said, monday, ...	secretary state john kerry said monday stop pa...	1
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	[kaydee, king, kaydeeking, november, lesson, t...	kaydee king kaydeeking november lesson tonight...	0
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	[primary, york, runners, hillary, clinton, don...	primary york runners hillary clinton donald tr...	1
...	...	...	...	...	...	...
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	[state, department, told, republican, national...	state department told republican national comm...	1
6331	8062	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	[stand, plutocratic, pentagon, posted, wikimed...	stand plutocratic pentagon posted wikimedia te...	0
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc	Anti-Trump Protesters Are Tools of the Oligar	[anti, trump, protesters, tools, oligarchy, re	anti trump protesters tools oligarchy reform p	0

In [ ]:
```
x=df['clean_joined']
y=df['label_REAL']
```

In [ ]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.1,random_state=2021)
```

In [ ]:
```
from nltk import word_tokenize
```

In [ ]:
```
tokenizer = Tokenizer(num_words = total_words)
tokenizer.fit_on_texts(x_train)
```

In [ ]:
```
train_sequences = tokenizer.texts_to_sequences(x_train)
test_sequences = tokenizer.texts_to_sequences(x_test)
```

In [ ]:
```
train_sequences[2]
```

Out[ ]:
```
[6,
 21640,
 2567,
 231,
 7123,
 2344,
 4420,
 1546,
 887,
 14,
 5007,
 11,
 11,
 32,
 1,
 7723,
 6585,
 69,
 842,
 6489,
 69,
```

```
        918,
        272,
        18495,
        544,
        678,
        3108,
        3767,
        596,
        478,
        2517,
        766,
        8056,
        3634]
```

In [ ]:
```
padded_train = pad_sequences(train_sequences,maxlen = 40, padding = 'post', truncating = 'post')
padded_test = pad_sequences(test_sequences,maxlen = 40, truncating = 'post')
```

In [ ]:
```
rnn_model = Sequential()
```

In [ ]:
```
rnn_model.add(Embedding(total_words, output_dim = 128))
```

In [ ]:
```
rnn_model.add(Bidirectional(LSTM(128)))
```

In [ ]:
```
rnn_model.add(Dense(128, activation = 'relu'))
rnn_model.add(Dense(1,activation= 'sigmoid'))
rnn_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
rnn_model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 128)         7954944
_________________________________________________________________
bidirectional_1 (Bidirection (None, 256)               263168
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 129
```

In [ ]:
```
total_words
```

Out[ ]:
```
62148
```

In [ ]:
```
y_train = np.asarray(y_train)
```

In [ ]:
```
rnn_model.fit(padded_train, y_train, batch_size =512,epochs =3)
```

```
Epoch 1/3
12/12 [==============================] - 14s 855ms/step - loss: 0.6843 - acc: 0.5779
Epoch 2/3
12/12 [==============================] - 10s 859ms/step - loss: 0.4245 - acc: 0.8594
Epoch 3/3
12/12 [==============================] - 10s 846ms/step - loss: 0.1643 - acc: 0.9425
```

Out[ ]:
```
<tensorflow.python.keras.callbacks.History at 0x7fde15f45290>
```

In [ ]:
```
```

Accuracy Scores:

As seen in the above picture, the accuracy was nearly equal to 94% which is an improvement from the naïve bayes and decision tree classifier.

9. Results

- As seen above , among the four implementations , the RNN model gave the best result with an accuracy of over 94% .
- The naïve bayes models had an accuracy of about 87% and Decision Tree classifier had accuracy of 82%

10. Conclusion

- Classification was performed on the Kaggle Fake news Dataset to classify news as real or fake
- The various models used for this project are naïve bayes classifiers , decision tree

  and RNN.

- Among the ones mentioned above , the most accurate results were given by the RNN model .Thus it can be concluded that neural networks have performed better than the traditional machine learning models .
- The accuracy of naïve bayes was 87 % and that of Decision Tree classifier was 82%

11.Citations and References

- https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76
- https://www.analyticsvidhya.com/blog/2020/12/fake-news-classifier-on-us-election-news%F0%9F%93%B0-lstm-%F0%9F%88%9A/
- https://en.wikipedia.org/wiki/Fake_news
- https://towardsdatascience.com/fake-news-classification-with-recurrent-convolutional-neural-networks-4a081ff69f1a