A close up of a logo

Description automatically generated

**FAKE NEWS DETECTOR**

**Team Members:**

**Arjav Kala**

**Harshit Batia**

**Oindrila Choudury**

**Mehak Sood**

**Rushabh Yogesh Gala**

**Sirisha Nookala**

**Mentored By:**

**Karan Nandwani**

**ACKNOWLEDGEMENT**

The success and outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all through the completion of our project. All that we have done is only due to the supervision and assistance that we got, and we would not forget to thank them.

We want to express our deep and sincerest gratitude to our mentor, Mr. Karan Nandwani for providing us with invaluable feedback and guidance throughout this project. His dynamic vision, sincerity and motivation has deeply inspired us. Not only did he teach us the methodology involved in building a product, but also helped us to present our work as clearly as possible. It was a great privilege and honor to work under his guidance and mentorship. We are extremely grateful for what he offered us and are thankful for his friendship, commitment and empathy. We appreciate his knowledgeful support and guidance, although he had busy schedule managing the corporate affairs.

Lastly, we are thankful to and fortunate enough to get constant encouragement, support and guidance from each other and our mentor which helped us in successfully building our product.

**TEAM INTRODUCTION**

The Goals of the team was to build and develop our product. Below are the names of the team members and their qualifications -

*Arjav Kala: MS in Business Analytics*

*Harshit Bhatia: MS in Information Management (BI and DS)*

*Mehak Sood: Dual degree MS-MIS & MBA with concentration in Finance and Business Intelligence*

*Oindrila Choudhury: MS-IM, Data Science & Business Intelligence*

*Rushabh Yogesh Gala: MS in Industrial Engineering*

*Sirisha Nookala: MS-MIS, Business Intelligence & Analytics*

**INDIVIDUAL CONTRIBUTION**

*Arjav Kala: Data Cleaning, Data Pre-Processing, Machine Learning Model Implementation*

*Harshit Bhatia: Data Cleaning, Pre-processing, Survey, ML Algorithms*

*Mehak Sood: Data Pre-Processing, Data Visualization, Project Documentation*

*Oindrila Choudhury: Deep Learning Algorithm Implementation, Customer Validation Survey,*

*Flask Application Design, AWS Hosting*

*Rushabh Yogesh Gala: Project Plan and Timeline Set-Up, Customer Validation Survey, Data*

*Visualization, Project Documentation*

*Sirisha Nookala: Deep Learning Algorithm Implementation, Customer Validation Survey,*

*Flask Application Design, AWS Hosting*

## CONTENTS

[CONTENTS 4](#_Toc47169937)

[INTRODUCTION 5](#_Toc47169938)

[GOALS 6](#_Toc47169939)

[FAKE NEWS DETECTOR: AS A PRODUCT 6](#_Toc47169940)

[PROBLEM HYPOTHESIS 7](#_Toc47169941)

[SOLUTION HYPOTHESIS 8](#_Toc47169942)

[CUSTOMER VALIDATION 8](#_Toc47169943)

[MODEL SELECTION 8](#_Toc47169944)

[MODEL EVALUATION 8](#_Toc47169945)

[UI DESIGN 9](#_Toc47169946)

[LEARNINGS 11](#_Toc47169947)

[FUTURE SCOPE 12](#_Toc47169948)

[APPENDIX 13](#_Toc47169949)

[Dataset Description 13](#_Toc47169950)

[Data Pre-processing 13](#_Toc47169951)

[K nearest neighbors (KNN) 15](#_Toc47169952)

[Logistic Regression 15](#_Toc47169953)

[Bi-directional RNN 15](#_Toc47169954)

[Long Short Term Memory 16](#_Toc47169955)

[BERT 16](#_Toc47169956)

[REFERENCES 17](#_Toc47169957)

## INTRODUCTION

"Fake news" refers to intentionally and verifiably false stories that are largely disseminated through social media networks. It can be very persuasive and therefore it is necessary to develop strategies to identify and critically assess news you read on social media.

As per the recent study by Facebook almost 30% of the news that we come across in day to day life is not accurate. The basic principle in spreading of misinformation is the amplification factor which can completely distort a small portion of falsified information into a complete article of its own. Hence it is really important that we try and cut the amplification as soon as possible in the cycle. In such try times, it is our attempt to restrict the spread of misinformation via digital medium.

Detecting fake news on social media poses several new and challenging research problems. Though fake news itself is not a new problem–nations or groups have been using the news media to execute propaganda or influence operations for centuries–the rise of web-generated news on social media makes fake news a more powerful force that challenges traditional journalistic norms. There are several characteristics of this problem that make it uniquely challenging for automated detection. First, fake news is intentionally written to mislead readers, which makes it non-trivial to detect simply based on news content. The content of fake news is rather diverse in terms of topics, styles and media platforms, and fake news attempts to distort truth with diverse linguistic styles while simultaneously mocking true news. For example, fake news may cite true evidence within the in-correct context to support a non-factual claim. Thus, existing hand-crafted and data-specific textual features are generally not sufficient for fake news detection. Other auxiliary information must also be applied to improve detection, such as knowledge base and user social engagements. Second, exploiting this auxiliary information actually leads to another critical challenge: the quality of the data itself. Fake news is usually related to newly emerging, time-critical events, which may not have been properly verified by existing knowledge bases due to the lack of corroborating evidence or claims. In addition, users’ social engagements with fake news produce data that is big, incomplete, unstructured, and noisy. Effective methods to differentiate credible users, extract useful post features and exploit network interactions are an open area of research and need further investigations.

## GOALS

Through this product, our major goal is to learn and build our knowledge about data modeling techniques from regression to deep machine learning algorithms. Learning the implementation of these algorithms and identifying its accuracies in the scenario was our major goal. The algorithms covered through this project were:

1. Logistic regression
2. KNN
3. SVM
4. BERT
5. Distil BERT
6. Combination of Distil BERT and Logistic Regression
7. LSTM
8. Bi-Directional LSTM

Through the ongoing pandemic, we have faced a situation where there was much fake news spreading through the people about the spread and transmission of COVID-19. With the product, we aimed to address the concern of fake news through the common public and implement these models to obtain higher accuracy in identifying fake news based on the context and author.

The secondary goal of this project was to learn about text visualizations and creating a word cloud representing the most used words in the fake news as well as the authors who have published such news.

## FAKE NEWS DETECTOR: AS A PRODUCT

The Fake News Detector is a web application that allows you to detect and flag news from Facebook, Twitter or a News Article as legitimate news or fake news.

The Detector uses a deep learning algorithm. The Detector has read thousands of articles from sites all over the internet and has been trained to detect relevant Fake News patterns. Our system reads the information given by us humans and learns with time to automatically flag news as Fake News based on its text and author. By doing that, we can even flag fresh news that no one saw.

## PROBLEM HYPOTHESIS

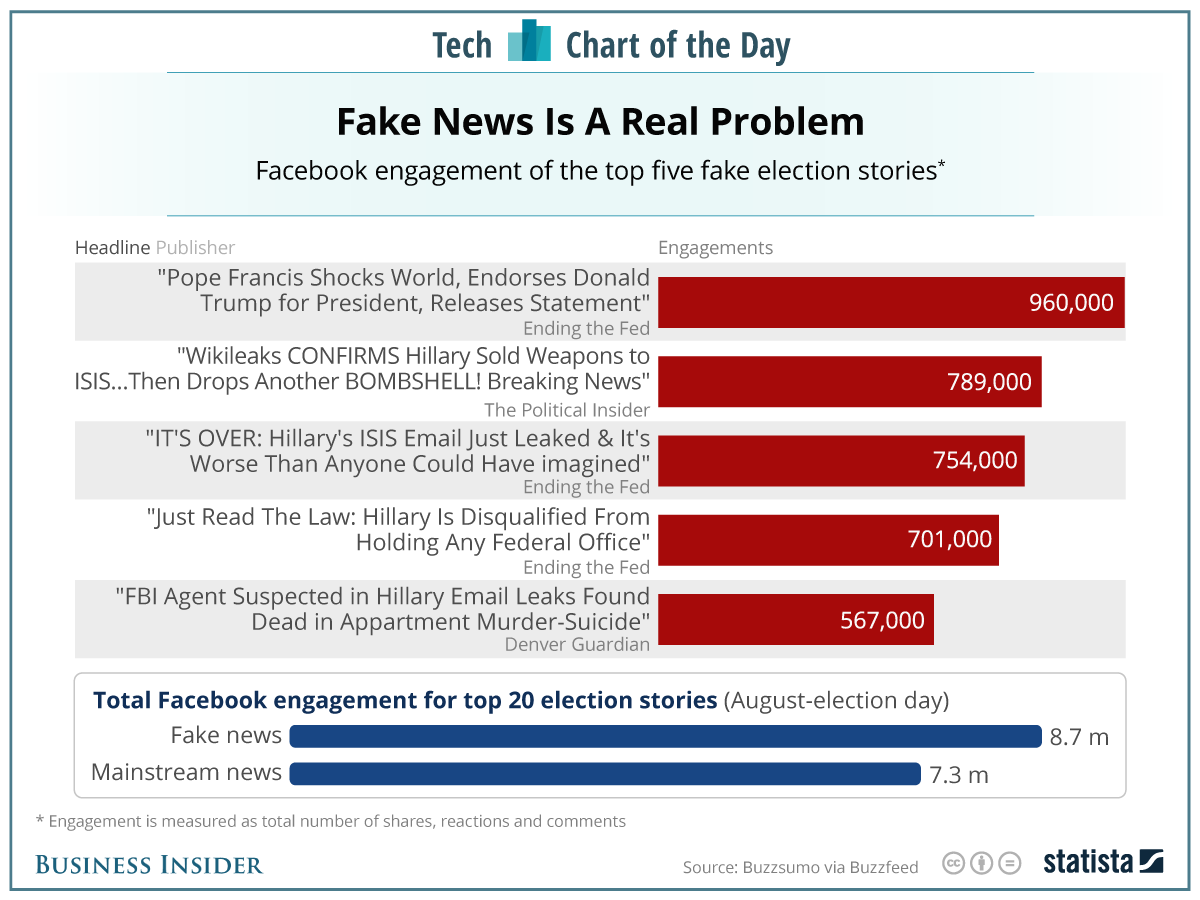
According to the News Consumption Statistics, 43% of Americans got their news often from either news websites or social media. While most adults in the age group of 24-39 preferred to consume news via online platforms.

According to our survey, 95.1% people said that they have consumed some kind of fake news in the last two months.

With the rapid increase of viewership in different channels including social media, the demand for “fake news” may be a natural byproduct of faster news cycles and increasing consumer demand for shorter-form content.

This news consumption from different channels including social media and web platforms increases their propensity to come across a fake news article. The proliferation of fake news articles can prove to be disastrous in case of a pandemic or during major elections.

For example, the Presidential Election Campaign of 2016 may be remembered for the proliferation of fake news stories. Viral news hoaxes have been around for many years, but in 2016, they exploded into the consciousness of the American public.



While there is a general awareness of the existence of “fake news,” there is widespread disagreement over what comprises “fake news.” Flagging news as “fake” can help in the spread of such news further. It is thus important to curb the spread of such news by flagging such articles.

## SOLUTION HYPOTHESIS

The Fake News Detector is a small initiative to try to make some difference in the fight against this proliferation of fake news. It is an automated fake-news detection system utilizing machine-learning models to catch subtle but consistent differences in the language of factual and false stories.

The Fake News Detector can be used as a plug-in extension for the Google Chrome Browser. This application can utilize crowdsourcing and Machine Learning to detect the Fake news. The fake news can be flagged by the users. The Machine Learning algorithms can use sentimental analysis and different parameters available in the article to predict if the news is fake or reliable.

## CUSTOMER VALIDATION

* 95.1% of users have come across fake news
* 60% have seen mis-information on social media upto 5 times a week
* SOurce and the structure of the articles were main reasons for the suspit

## MODEL SELECTION

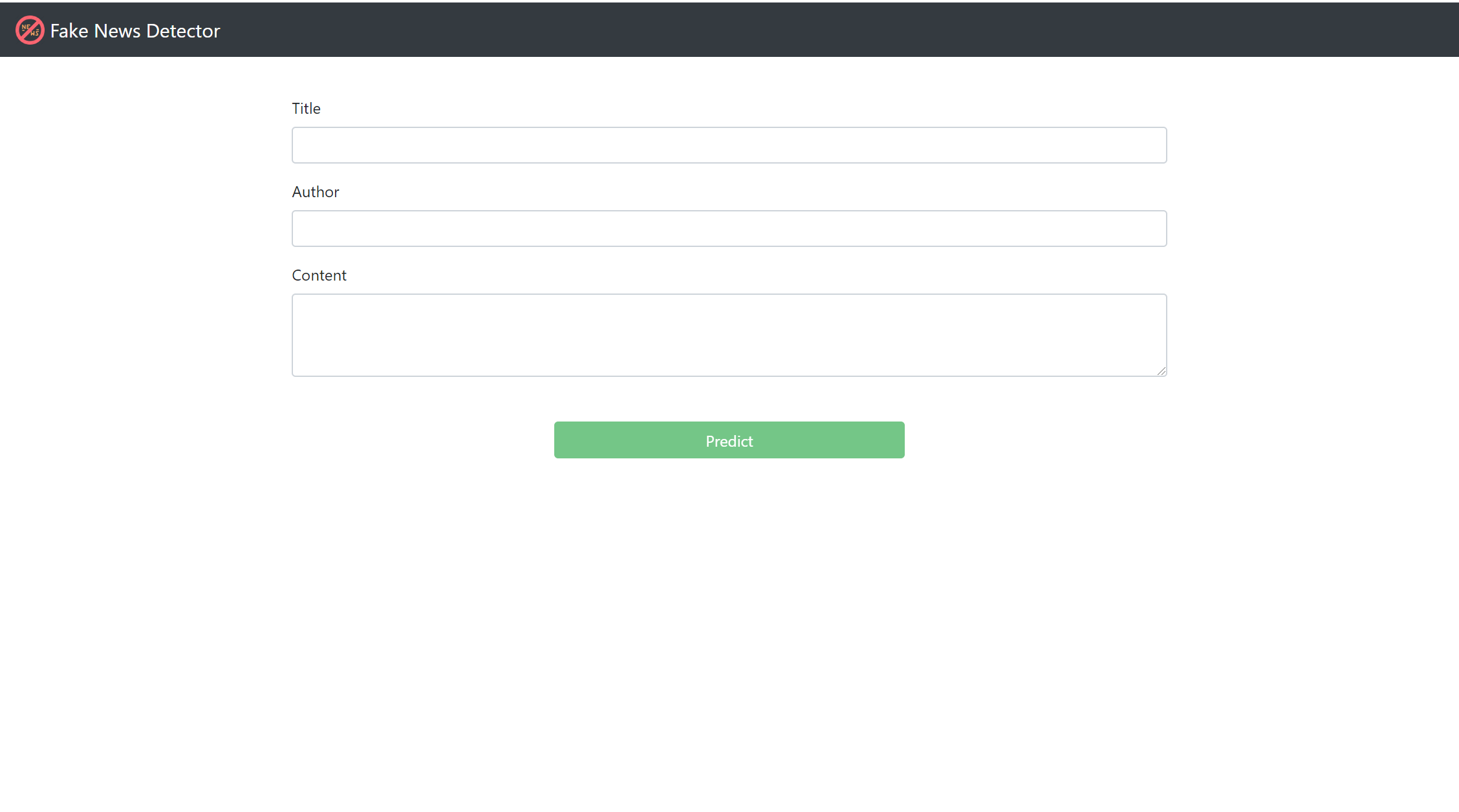
Classification being the primary goal, multiple data mining algorithms have been applied on the dataset. The baseline models being logistic regression and KNN, we’ve realized that the accuracy is saturated, and we had to shift to deep-learning algorithms for improving the accuracy. Hence, we explored BERT, DistilBERT and LSTM for improving the performance.

## MODEL EVALUATION

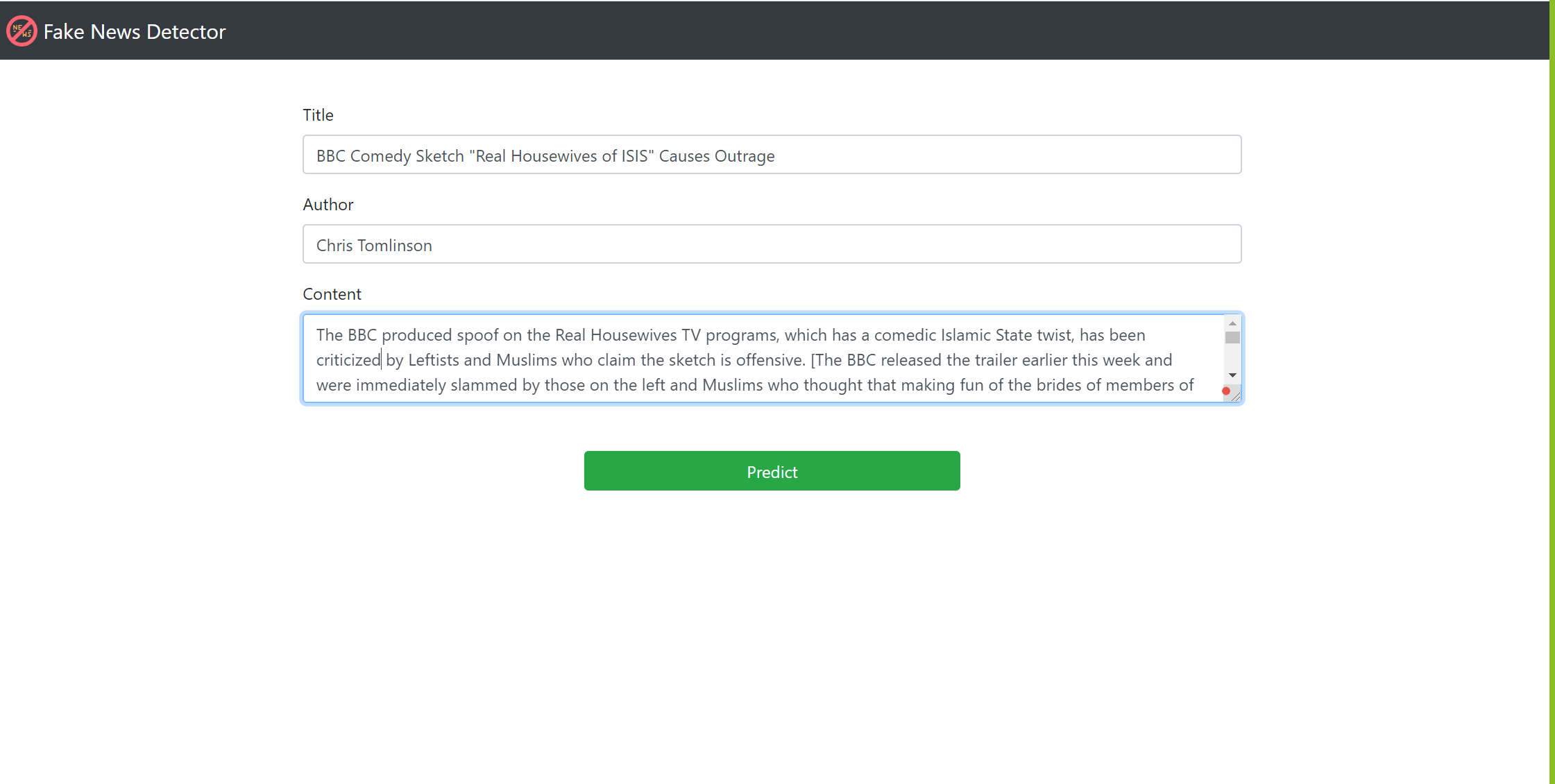
|  |  |
| --- | --- |
| **MODEL** | **ACCURACY** |
| Logistic Regression | 83 |
| KNN | 82 |
| SVM | 84 |
| BERT (without hyper-parameter tuning) | 66 |
| Distil BERT + Logistic Regression | 86 |
| LSTM | 91 |
| Bi-Directional RNN | 90 |

## UI DESIGN

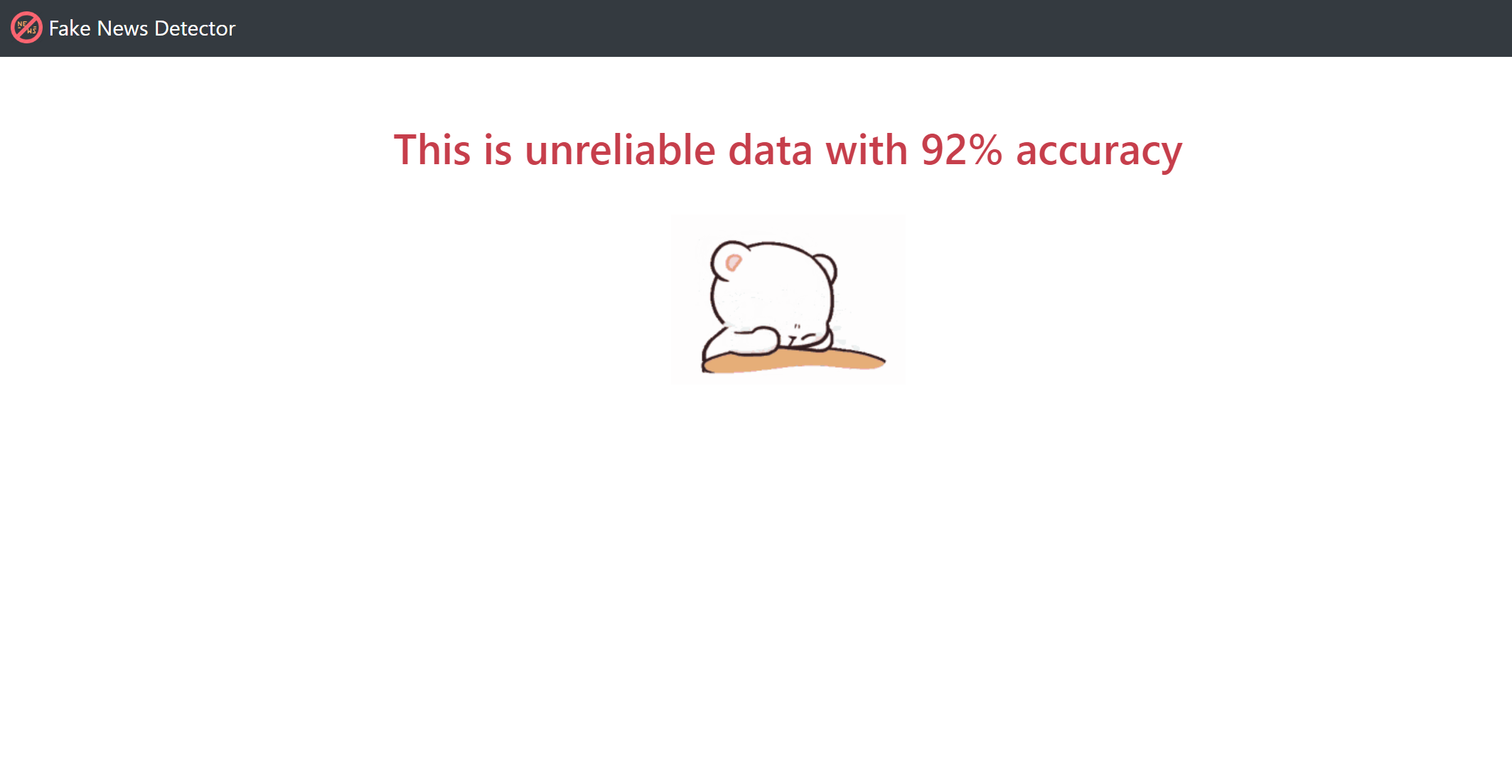
The model has been deployed through a Flask application with Bootstrap, HTML and CSS on the UI. This has been hosted publicly on an AWS EC2 Ubuntu instance using nginx and gunicorn as a web server and a web gateway. The interface provides you information about the credibility of data based on user inputs (title, author, content of the article) where the content of the article is mandatory.



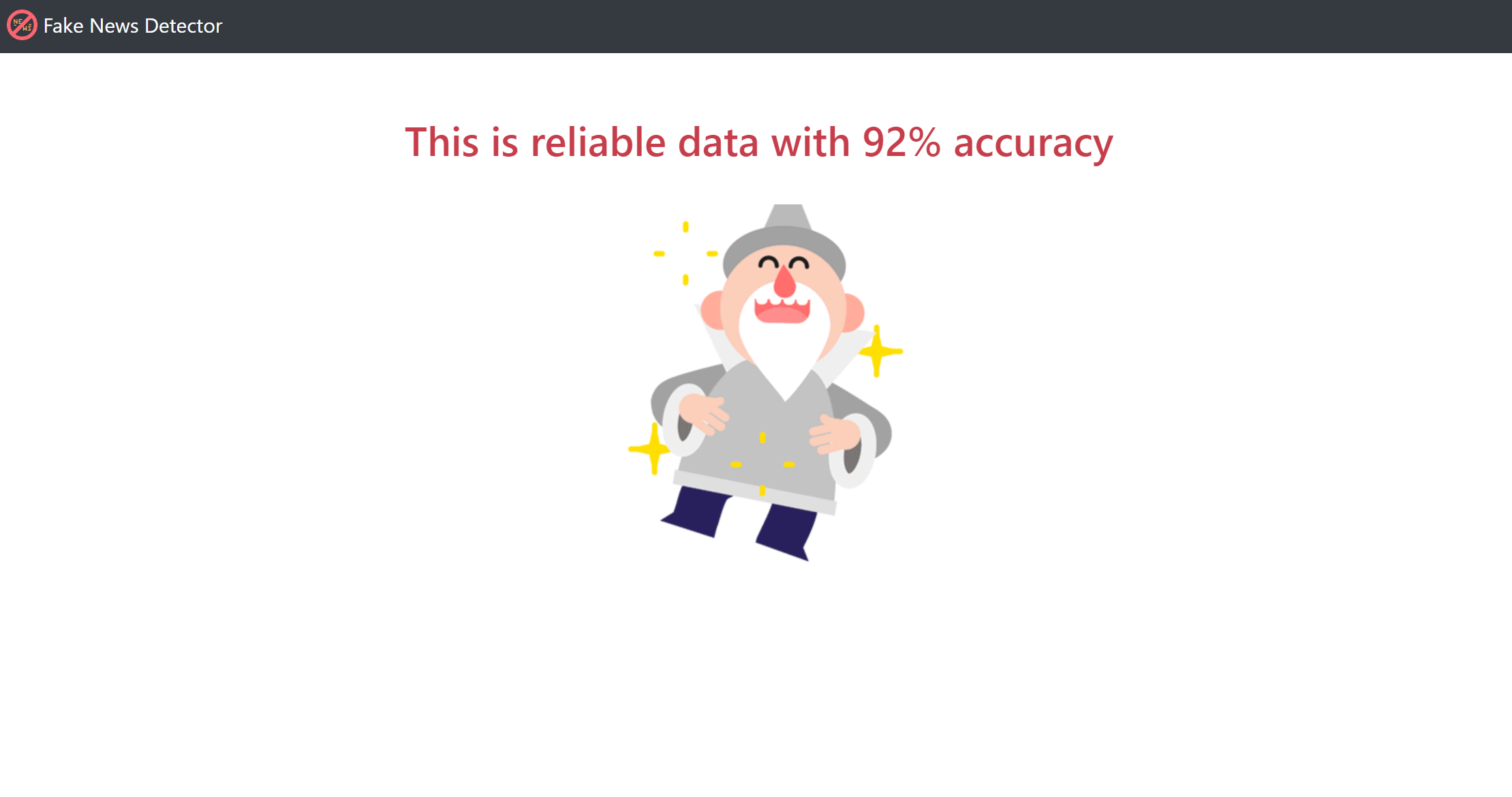
*Figure 1: The home page where the user enters input*

**

*Figure 2: The home page after entering the input*



*Figure 3: The screen that shows up when the user enters unreliable information*



*Figure 4: The screen that shows up when the user enters reliable information*

## LEARNINGS

* Virtual collaboration:
  + We managed to coordinate and collaborate multiple times a week virtually, even when most of us are in different time-zones. This taught us the importance of teamwork and collaboration
* Project Management:
  + We learnt a lot of PM skills like scheduling the project, prioritizing features, effective communication, team management, adaptability, reporting, conflict management, etc.
* Data science Algorithms:
  + We trained our model on several data science algorithms using python, hence we learned in-depth about functioning of these algorithms
* Deep learning:
  + We also learned in-depth about RNN, LSTM and distilled BERT which eventually gave better results for our dataset
* AWS:
  + We hosted the instance of our website using AWS, hence familiarizing ourselves with its concept
* Survey:
  + Inputs from other were one of the crucial factors in determining the path of our project, hence we carried detailed research about best practices in developing survey forms

## FUTURE SCOPE

The current model gives a binary output of whether the news is fake or not. This can be upgraded to a probability output with the percentage or a probability of the news being fake. The higher probability of the news, the more cautious the user about its veracity.

With the UI ready, the product can be upgraded with increases in the size of the dataset. Also, the model currently detects fake news from the data set itself. With the change in the technique and implementation, detecting fake news from new inputs would be an ideal implementation of the product.

From the point of implementation, additional modes of reading context and identification of fake news can be created for enhanced user experience. Few examples of such modes are:

1. Chrome Extension
2. Keyboard Button

These modes will comprise selecting a context from an article and the software giving the percentage or a probability of the context being scripted.

## APPENDIX

### Dataset Description

The dataset for this project has been acquired from Kaggle. The data has been collected from various news websites and contains the news about 2016 United States elections. It has a total of 5 columns which includes columns like id, title, author, text, label etc. This data has 3 categorical variables and 2 numeric variables. The dataset that we utilized contains:

• **id**: unique id for a news article

• **title**: the title of a news article

• **author**: author of the news article

• **text**: the text of the article; could be incomplete

• **label**: a label that marks the article as potentially unreliable

1: unreliable

0: reliable

### Data Pre-processing

The dataset contains a lot of junk characters and missing values. The dataset contains some columns which were not useful such as id. As the column was not giving us any additional information, we drop the ‘id’ column from the data frame. Also, while analyzing the dataset, we found out that there were many missing values in the title, author, and text column. We replaced the missing values with the empty strings.

When a user enters the information to test the credibility of the source, it may be so that the user does not have any information about the title or author of the article. In order to combat the problem, we created a new column which contains concatenation of title, author and text.

Process to prepare the textual data for model training and evaluation:

1. **Conversion to lower case**: We converted the textual data columns into lower case to make sure that the model is not case-sensitive.
2. **Tokenization:** Further, we also broke the text into individual words
3. **Stop Words Removal:** Next, All the common words such as for, in were filtered out from the data
4. **Lemmatization**: Lastly, we also removed the end of all the words to extract the root words from our dataset
5. **Vectorizer**: Different techniques have been used to convert all the token (words) into vectors

TF-IDF Vectorizer:

The word counts are a good starting point, but one issue with simple counts is that some words like “the” will appear many times, and their large counts will not be significant in the encoded vectors. TF-IDF calculates the word frequencies, and by far is the most popular method. This is an acronym that stands for Term Frequency – Inverse Document Frequency which are the components of the resulting scores assigned to each word.

* The Term Frequency (TF) summarizes how often a given word appears within a document.
* The Inverse Document Frequency (IDF) downscales words that appear a lot across documents.

The TF-IDF is word frequency score that tries to highlight the words that are more interesting, e.g., frequent in a document but not across documents. The TF-IDF Vectorizer will tokenize documents, learn the vocabulary and inverse document frequency weightings, and allow you to encode new documents. The TF-IDF penalizes the most recurring words in the document and gives less importance to those words.

Word Embeddings:

A word embedding is a learned representation for text where words that have the same meaning have a similar representation. It is this approach to representing words and documents that may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems. Each word is mapped to one vector and the vector values are learned in a way that resembles a neural network, and hence the technique is often lumped into the field of deep learning. Key to the approach is the idea of using a dense distributed representation for each word.

BERT Tokenizers:

We will use BERT to extract features, namely word and sentence embedding vectors, from text data. These embeddings are useful for keyword/search expansion, semantic search and information retrieval. For example, if you want to match customer questions or searches against already answered questions or well documented searches, these representations will help you accurately retrieve results matching the customer’s intent and contextual meaning, even if there’s no keyword or phrase overlap. Also, these vectors are used as high-quality feature inputs to downstream models. NLP models such as LSTMs or CNNs require inputs in the form of numerical vectors, and this typically means translating features like the vocabulary and parts of speech into numerical representations.

Once the data was preprocessed, we utilize a 50-50 split for our training and test data. We then trained and tested the below models –

### K nearest neighbors (KNN)

The K nearest neighbors (KNN) is a simple algorithm that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition.

A simple implementation of KNN regression is to calculate the average of the numerical target of the K nearest neighbors. Another approach uses an inverse distance weighted average of the K nearest neighbors. KNN is a lazy learner algorithm and does not learn anything in the training period. It stores the training dataset and learns from it only at the time of making real time predictions. Also, it is a non-parametric algorithm which means there are various assumptions to be met to implement KNN. But, KNN cannot work well with high dimensional data and performs poorly when the data is imbalanced. It is very sensitive to the outliers as well.

### Logistic Regression

Logistic Regression a classification algorithm, that is used where the response variable is categorical. The idea of Logistic Regression is to find a relationship between features and probability of particular outcome.  
Logistics regression is a simple and efficient algorithm with low variance. It also provides a probability score for our observations. But it doesn’t handle a large number of categorical features well and requires transformation of non-linear features.

### Bi-directional RNN

RNNs are a very powerful tool to deal with sequence data, they provide incredible memorizing capacities and are widely used in day-to-day life. They also have many extensions which enable them to address various types of data-driven problems.

Bidirectional recurrent neural networks (RNN) are just putting two independent RNNs together. The input sequence is fed in normal time order for one network, and in reverse time order for another. The outputs of the two networks are usually concatenated at each time step, though there are other options, e.g. summation. This structure allows the networks to have both backward and forward information about the sequence at every time step.

### Long Short Term Memory

LSTMs (Long Short Term Memory) were introduced to overcome the problem of short memory, they have 4 times more memory than RNNs. This model uses the notion of gates and has three gates:

* **Input gate i**: controls the flow of incoming information.
* **Forget gate f**: Controls the amount of information from the previous memory state.
* **Output gate o:** controls the flow of outgoing information

When the input and output doors are closed, activation is blocked in the memory cell.

The LSTM can model long-term sequence dependencies. They are more robust to the problem of short memory than ‘Vanilla’ RNNs since the definition of the internal memory is changed. But they increase the computing complexity compared to the RNN with the introduction of more parameters to learn. The memory required is higher than the one of ‘Vanilla’ RNNs due to the presence of several memory cells.

### BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers. is a model that knows to represent text. You give it some sequence as an input, it then looks left and right several times and produces a vector representation for each word as the output. BERT allows us to perform different tasks based on its output. So for different task types, we need to change the input and/or the output slightly.

There are two ways to work with BERT, one as with the “feature extraction” mechanism. That is, we use the final output of BERT as an input to another model. This way we’re “extracting” features from text using BERT and then use it in a separate model for the actual task in hand. The other way is by “fine-tuning” BERT. That is, we add additional layer/s on top of BERT and then train the whole thing together. This way, we train our additional layer/s and also change (fine-tune) the BERTs weights.

## REFERENCES

1. <https://windowsreport.com/fake-news-detector-pc/>
2. <https://www.theatlantic.com/ideas/archive/2019/06/fake-news-republicans-democrats/591211/>
3. <https://libraryguides.unh.edu/fakenews>
4. <https://advances.sciencemag.org/content/6/14/eaay3539>

1. <https://library.piercecollege.edu/c.php?g=598055&p=4428903>