Final Project Report

For

FactFlow: Navigating News With Certainty

Prepared by:

Name	Roll No	Branch
Aayush Aaryan	R2142210014	CSE AIML
Shaurya Srivastava	R2142210729	CSE AIML
Indure Sainath Manoj	R2142210354	CSE AIML

AI Cluster
School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand

Achala Shakya
Project Guide

Dr. Anil Kumar
Cluster Head AI

Table of Contents

Topic: FactFlow: Navigating News With Certainty		Page No
Table of Content		2
1	Introduction	3-5
	1.1 Purpose of the Project	3
	1.2 Target Beneficiary	3
	1.3 Project Scope	3-4
	1.4 References	4-5
2	Project Description	5-12
	2.1 Reference Algorithm	5
	2.2 Characteristics of Data	5-6
	2.3 Project Features	6-7
	2.4 SWOT Analysis	8-9
	2.5 User Classes and Characteristics	9-10
	2.6 Design and Implementation Constraints	10
	2.7 Design diagrams	11
	2.8 Assumption and Dependencies	11-12
3	System Requirements	12-13
	3.1 Software Interface	12
	3.2 Database Interface	13
4	Non-functional Requirements	13-14
	4.1 Performance requirements	13
	4.2 Security requirements	13
	4.3 Software Quality Attributes	13-14
5	Code and Output	14-18
	5.1 Python code snapshots	14-17
	5.2 Output and evaluation matrix	18

1. INTRODUCTION:

1.1 Purpose of the Project:

Driven by the purpose of mitigating the detrimental effects of online misinformation, this detector seeks to empower users and devices across platforms to effectively identify and combat fake news. By harnessing the capabilities of machine learning algorithms, natural language processing techniques, and data analytics, the detector aims to provide users with reliable tools to discern between credible and fabricated content. Furthermore, it aspires to enhance the resilience of digital ecosystems against the propagation of fake news, thereby fostering a more trustworthy and transparent information environment.

1.2 Target Beneficiary:

The target beneficiary of the fake news detector project would primarily be internet users, including individuals, organizations, and communities, who consume online content. Additionally, digital platforms, social media networks, and news organizations could also benefit from the detector's capabilities to combat the spread of misinformation. Ultimately, the goal is to empower all stakeholders involved in the online information ecosystem to make informed decisions and contribute to a more trustworthy digital environment.

1.3 Project Scope :

Scope of our project:

1. Detection Algorithm Development: Develop machine learning algorithms and natural language processing techniques to identify patterns and features indicative of fake news content.

2. Data Collection and Analysis: Collect a diverse dataset of news articles, social media posts, and online content for training and testing the detection algorithms. Analyze the data to extract relevant features and establish baseline metrics.

3. Detector Implementation: Build a user-friendly interface for the fake news detector that can be integrated across various platforms, including web browsers, social media platforms, and news aggregator apps.

4. Real-time Monitoring: Implement a mechanism for real-time monitoring of online content to quickly identify and flag potentially fake news articles or posts.

5. User Feedback Mechanism: Incorporate a feedback loop where users can report suspicious content and provide feedback on the detector's accuracy, thereby improving its performance over time.

6. Evaluation and Validation: Conduct thorough evaluations and validation tests to assess the detector's effectiveness, including precision, recall, and false positive rates.

7. Documentation and Deployment: Provide comprehensive documentation for users and developers on how to use and integrate the fake news detector. Deploy the detector in a production environment, ensuring scalability and reliability.

8. Community Engagement: Foster collaboration with other researchers, organizations, and stakeholders working on combating misinformation to share knowledge, resources, and best practices.

9. Continuous Improvement: Establish a framework for ongoing research and development to adapt the detector to evolving tactics used in spreading fake news and to incorporate advancements in machine learning and natural language processing technologies.

1.4 References:

1.] Citation Z Khanam et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1099 012040 DOI 10.1088/1757-899X/1099/1/012040

2.] N. F. Baarir and A. Djeffal, "Fake News detection Using Machine Learning," 2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH), Boumerdes, Algeria, 2021, pp. 125-130, doi: 10.1109/IHSH51661.2021.9378748.

3.] J. Shaikh and R. Patil, "Fake News Detection using Machine Learning," 2020 IEEE
International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), Gunupur Odisha, India, 2020, pp. 1-5, doi: 10.1109/iSSSC50941.2020.9358890.

4.] Prabha, T., Aisuwariya, T., Kiran, M., & Vasudevan, Shriram. (2020). An Innovative and Implementable Approach for Online Fake News Detection Through Machine Learning. Journal of Computational and Theoretical Nanoscience, 17, 130-135. DOI: 10.1166/jctn.2020.8639.

5.] C. De Silva and T. Halloluwa, "The Detection of Fake News through the use of a Hierarchical Convolutional Attention Network," in 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-6, doi: 10.1109/I2CT57861.2023.10126324. Keywords: Recurrent neural networks; Correlation; Social networking (online); Publishing; Bit error rate; Companies; Convolutional neural networks; Social media; Fake News; Convolutional Neural Network (CNN); Hierarchical Convolutional Attention Network (HCAN).

6.] arXiv:1910.03496 [cs.CL] (or arXiv:1910.03496v2 [cs.CL] for this version) https://doi.org/10.48550/arXiv.1910.03496

7.] Lathkar, M. (2021). Building Web Apps with Python and Flask: Learn to Develop and Deploy Responsive RESTful Web Applications Using Flask

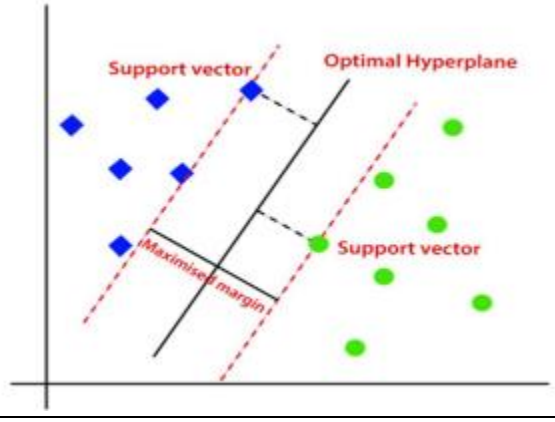
Framework. India: BPB.

2. PROJECT DESCRIPTION:

2.1 Reference Algorithm:

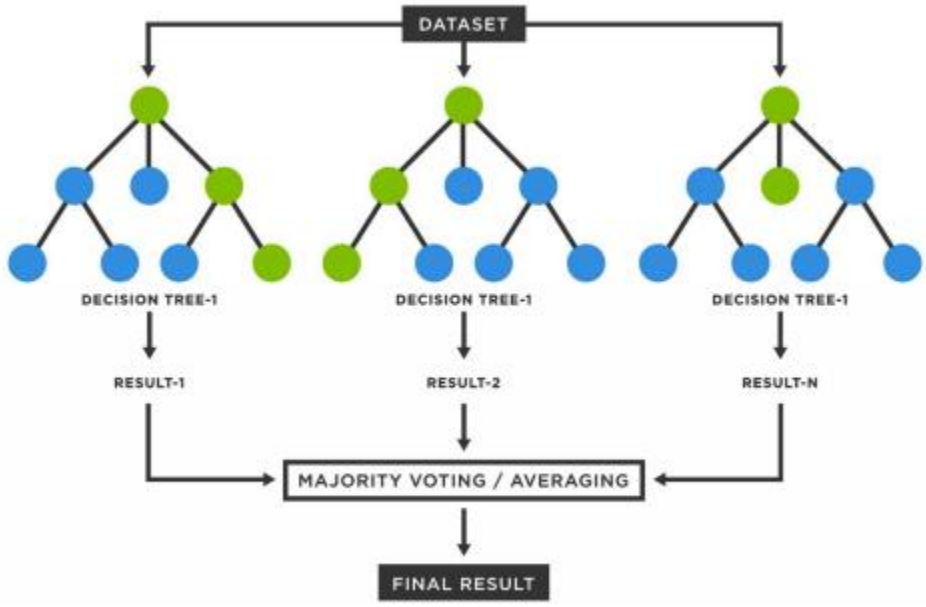
In this study, we used 5 supervised ML classification method to identify false news: LR, NB, SVM, DT, and RF, and determined which approach performed best based on accuracy. The description algorithms used are given as follows:

- **Logistic Regression:** It's a tool for categorizing binary data. For binary classification usually, Linear regression is used to create the best bit line. When two classes can be separated linearly, logistic regression is used. It is within the supervised machine learning algorithm category. It's a machine-learning-based categorization problem-solving approach. In logistic regression, a type of predictive analysis, the probability assumptions are applied. To complete a binary classification job, a linear equation is used as input, and the logistic function and log odds are used in the logistic regression model. It employs a more complicated function when compared to linear regression.
- **Naïve Bayes:** The Naive Bayes approach, a supervised machine learning methodology based on the well-known Bayes theorem, is used to tackle classification issues. It's most commonly used for text classification with a big training dataset. One of the most simple and effective classification methods is the Naive Bayes Classifier. It allows for the rapid building of machine learning models as well as effective training and testing to make speedy predictions. It's a probabilistic classifier, which implies the algorithm's whole basis is built on probabilities that have been computed, and it predicts based on an item's likelihood.
- **Support Vector Machine:** The Support Vector Machine approach aims to find a hyperplane (where N is the number of characteristics) that clearly arranges the principal elements in an N-dimensional space. There is an assortment of hyperplanes from which to isolate the two kinds of informative items. Our point is to track down the plane with the biggest edge, or distance between relevant items from the two classes.



- **Random Forest:** Random forest is a supervised ML technique. It is basically established on the outfit learning techniques where different classifiers are united to deal with an issue and to chip away at the display of the presentation of the model. Random Forest is a

classifier that calculates the dataset's predicted precision by averaging the results of many decision trees applied to different subsets of the dataset. As given in below figure we can say that Random Forest is combines two or more decision trees and gives the results.



- **Stochastic Gradient Descent Classifier:** The Stochastic Gradient Descent (SGD) classifier is a variant of the traditional Gradient Descent algorithm commonly used in machine learning for optimization tasks, particularly in the context of training classification models. Unlike traditional Gradient Descent, which computes the gradient using the entire training dataset in each iteration, Stochastic Gradient Descent computes the gradient using only a single training example (or a small subset of examples) at a time. This makes SGD much faster, especially for large datasets, as it updates the parameters more frequently.

2.2 Characteristics of Data:

1. **Relevance**: The data should be directly relevant to the problem or task at hand. Irrelevant or extraneous data can introduce noise and decrease the effectiveness of analysis or modelling.
2. **Completeness**: The dataset should contain all the necessary information required for the analysis or task. Missing values or incomplete records can hinder the accuracy of results.
3. **Quality**: High-quality data is accurate, reliable, and free from errors, inconsistencies, or duplicates. Data cleaning and preprocessing techniques may be necessary to ensure data quality.

4. **Representativeness**: The dataset should accurately represent the population or phenomenon being studied. Biases or skewed distributions can lead to inaccurate or biased results.
5. **Variety**: A diverse range of data types and attributes can provide richer insights and enable more comprehensive analysis. This includes numerical, categorical, textual, and multimedia data.

2.3 Project Features:

- **Fake News Detection Algorithm**: Develop a machine learning algorithm capable of identifying fake news using techniques such as natural language processing (NLP) and sentiment analysis.

- **Data Collection and Analysis**: Collect a diverse dataset of news articles and online content for training and testing the detection algorithm. Analyse the data to extract relevant features and patterns.

- **User Interface**: Design a user-friendly interface for the fake news detector that allows users to input news articles or URLs and receive instant feedback on their credibility.

- **Evaluation Metrics**: Implement evaluation metrics such as precision, recall, and F1-score to assess the detector's effectiveness and performance.

- **Documentation**: Provide comprehensive documentation for users and developers on how to use the fake news detector, including installation instructions, usage guidelines, and API documentation.

- **Integration**: Ensure the fake news detector can be easily integrated into various platforms, including web browsers, social media platforms, and news aggregator apps.

- **Scalability**: Design the fake news detector to be scalable, allowing it to handle large volumes of data and user requests without compromising performance.

- **Privacy and Security**: Implement measures to protect user privacy and data security, including encryption, data anonymization, and compliance with privacy regulations.

2.4 SWOT Analysis:

- **Strengths**:

- Machine Learning Expertise: Leveraging machine learning algorithms enables accurate analysis and classification of news articles.

- Potential Impact: Effective fake news detection contributes to mitigating the spread of misinformation, thereby promoting informed decision-making and societal well-being.

- Scalability: The system can be designed to scale to large datasets and high volumes of news articles for comprehensive analysis.

- Flexibility: Machine learning models can be trained and fine-tuned to adapt to evolving patterns of misinformation and new types of fake news.

- **Weaknesses**:
  - Data Quality: The effectiveness of the fake news detector heavily relies on the quality and representativeness of the training data. Poor-quality or biased datasets may lead to inaccurate classification.

  - Algorithm Complexity: Complex machine learning models may require significant computational resources for training and inference, leading to higher costs and potential scalability issues.

  - Adversarial Attacks: Fake news creators may actively try to circumvent detection algorithms through adversarial techniques, posing a challenge to the robustness of the system.

  - Interpretability: Deep learning models, in particular, may lack interpretability, making it difficult to understand and explain the decision-making process behind classification results.

- **Opportunities**:
  - Integration with Social Media Platforms: Collaborating with social media companies to integrate the fake news detector into their platforms could enhance its reach and impact in combating misinformation.

  - Continuous Improvement: Continuous monitoring and updating of the machine learning models based on user feedback and new data can improve the accuracy and effectiveness of the fake news detector over time.
  - Collaborative Research: Collaboration with researchers, journalists, and fact-checking organizations can provide valuable insights and enhance the credibility of the fake news detection system.

- Customization for Different Domains: Tailoring the fake news detector for specific domains (e.g., politics, health, finance) can improve its performance and relevance in different contexts.

- **Threats**:
  - Legal and Ethical Concerns: Implementation of the fake news detector may raise legal and ethical questions related to privacy, censorship, and freedom of speech, potentially leading to regulatory challenges or public backlash.

  - Resource Constraints: Limited resources in terms of funding, expertise, and infrastructure could hinder the development and deployment of the fake news detector.

  - Competitive Landscape: Competition from other fake news detection solutions or alternative approaches to combating misinformation may pose a threat to the adoption and success of your project.

  - Technological Advancements: Rapid advancements in technology, particularly in the generation of synthetic media (e.g., deepfakes), may pose new challenges to fake news detection algorithms, requiring continuous innovation and adaptation.

2.5 User Classes and Characteristics:

For a fake news detection project, the user classes could include:

- **General Users**: These are individuals who consume online content and are concerned about the authenticity of the information they encounter. They may use the fake news detector to verify the credibility of news articles, social media posts, or other online content before sharing or acting upon it.

- **Educators and Students**: Teachers, professors, and students may use the fake news detector as an educational tool to learn about misinformation and critical thinking skills. They can analyze news articles or social media posts in class or as part of assignments to discuss the importance of verifying information sources.

- **Journalists and Fact-Checkers**: Professionals in the journalism industry or fact-checking organizations may use the fake news detector as a tool to aid in their verification process. They can quickly assess the credibility of sources and identify potentially misleading or false information before publishing news articles or reports.

- **Social Media Platforms**: Social media platforms and online communities may integrate the fake news detector into their platforms to help users identify and report fake news

content. They can leverage the detector's capabilities to enhance content moderation efforts and promote a more trustworthy information environment.

- **Government Agencies and Policy Makers**: Government agencies, policymakers, and regulatory bodies may use the fake news detector to monitor the spread of misinformation and assess its impact on society. They can utilize the detector's insights to develop policies and initiatives aimed at combating online misinformation and promoting media literacy.

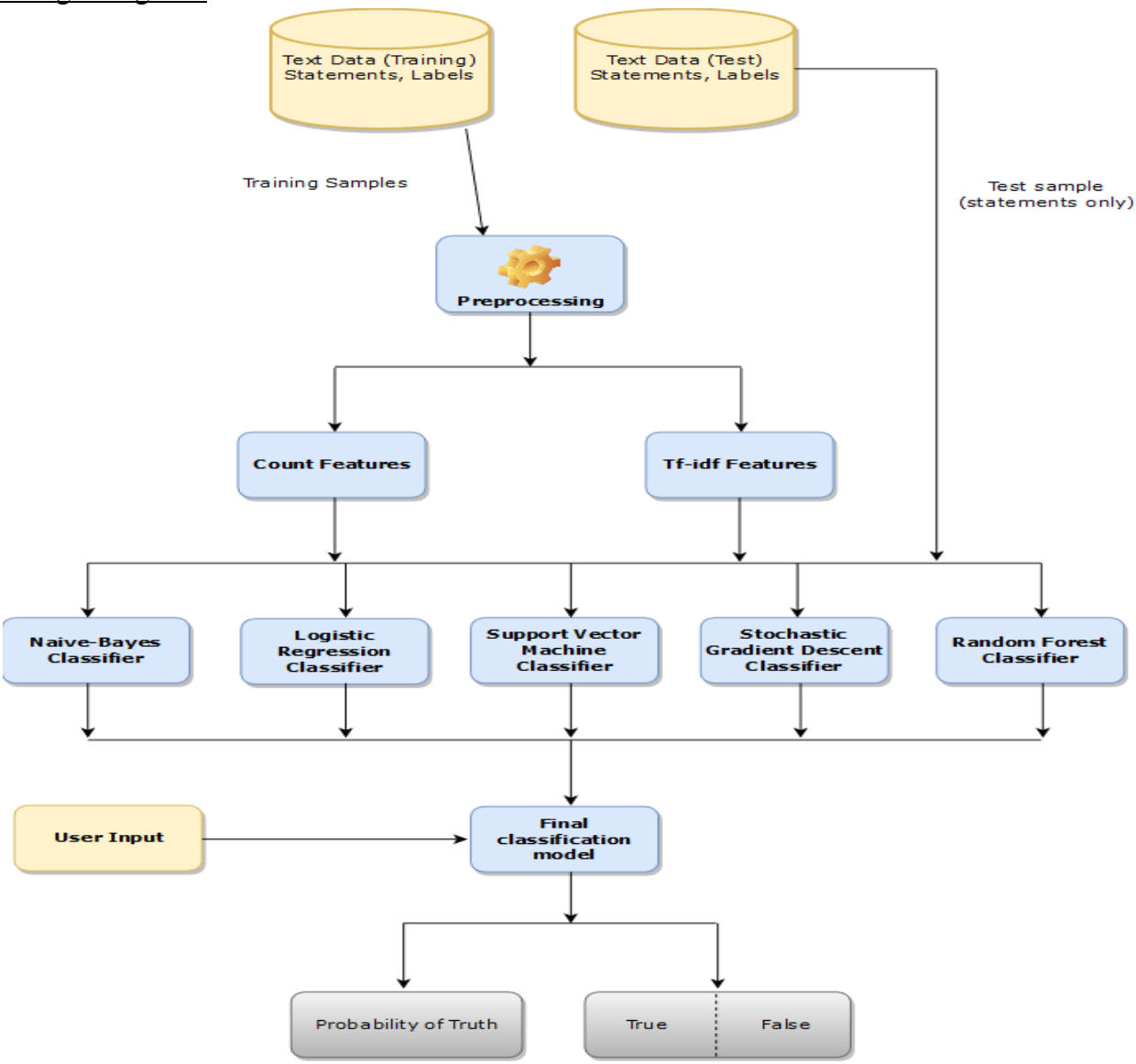
- **Researchers and Data Scientists**: Researchers and data scientists studying misinformation and online social dynamics may use the fake news detector as a tool for analysis and experimentation. They can leverage the detector's data and insights to conduct research on the spread of fake news, identify trends and patterns, and develop new detection algorithms and techniques.

2.6 Design and Implementation Constraints:

- **Interfaces to Other Applications:** Integrating with a web browser extension or social media platform APIs for news feed analysis.

- **Specific Technologies and Tools:**
  We will be using Python as our primary language, and below modules we would be using for our projects:
  - Numpy
  - Pandas
  - Sklearn
  - Itertools
  - Seaborn
  - Nltk
  - Pickle
  - Matplotlib

2.7 Design Diagrams:



2.8 Assumptions and Dependencies:

- **Assumptions**:
  - **Quality of Training Data**: The effectiveness of the fake news detection algorithm relies on the quality and diversity of the training dataset. It is assumed that the training data accurately represents a wide range of fake and credible news sources and that it is free from biases or inaccuracies.

  - **Language and Cultural Considerations**: The fake news detector may perform differently across different languages and cultures. It is assumed that the algorithm's effectiveness in detecting fake news is not significantly hindered by language barriers or cultural nuances.

- **User Engagement**: The success of the fake news detector depends on user engagement and feedback. It is assumed that users will actively use the detector, provide accurate feedback on flagged content, and report suspicious articles or posts to improve the algorithm's performance.

- **Algorithm Performance**: The effectiveness of the fake news detection algorithm may vary over time as fake news tactics evolve. It is assumed that the algorithm can adapt and improve over time to remain effective in detecting emerging patterns of misinformation.

- **Dependencies**:
  - **Data Sources**: The project depends on access to reliable and diverse datasets of news articles, social media posts, and online content for training and testing the fake news detection algorithm. Dependencies on external data sources may affect the project timeline and feasibility.

  - **Technological Infrastructure**: The project relies on access to computational resources, storage facilities, and software tools for developing and deploying the fake news detection system. Dependencies on external technologies and infrastructure may impact the scalability and performance of the system.

  - **Regulatory Compliance**: The project must comply with relevant privacy regulations and ethical guidelines governing data collection, storage, and usage. Dependencies on external regulatory frameworks may affect the project's implementation and deployment.

  - **User Engagement and Feedback**: The effectiveness of the fake news detector depends on user engagement and feedback. Dependencies on external factors such as user behaviour, preferences, and trust in the system may influence the success of the project.

3. SYSTEM REQUIREMENTS:

3.1 Software Interface:

- **Web Browsers**:
  - Support for popular web browsers: Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge.
- **Security Measures**:
  - Encryption protocols (HTTPS) for secure data transmission.
  - User authentication and authorization mechanisms.

3.2 Database Interface:

We will use preset dataset from Kaggle and edit data accordingly for this project.

- **Data Storage**: Storage of user profiles, electricity consumption data.
- **Data Privacy**: Implementation of privacy measures in compliance with relevant data protection regulations.
- User consent for data collection and analysis

4. NON-FUNCTIONAL REQUIREMENTS:

4.1 Performance requirements:

- **Response Time:** The respond time for any user request must be under 2-5 seconds.

- **Scalability:** The architecture should be scalable to accommodate future growth.

- **Reliability:** The system should have an uptime of 99.9%.

- **Resource Utilization:** The application should operate without causing significant strain on device resources.

4.2 Security requirements:

- **Data Encryption:**
  - All data transmitted between the client and server should be encrypted using HTTPS.
  - User passwords and sensitive information should be stored using strong encryption.

- **Access Control:**
  - User authentication is mandatory for accessing any system functionality.

4.3 Software Quality Attributes:

- **Usability:**
  - The user interface should be intuitive and user-friendly, requiring minimal training.
  - Provide tooltips and contextual help for users.

- **Maintainability:**

- The codebase should follow best coding practices and be well-documented.
- Implement version control for efficient code management.

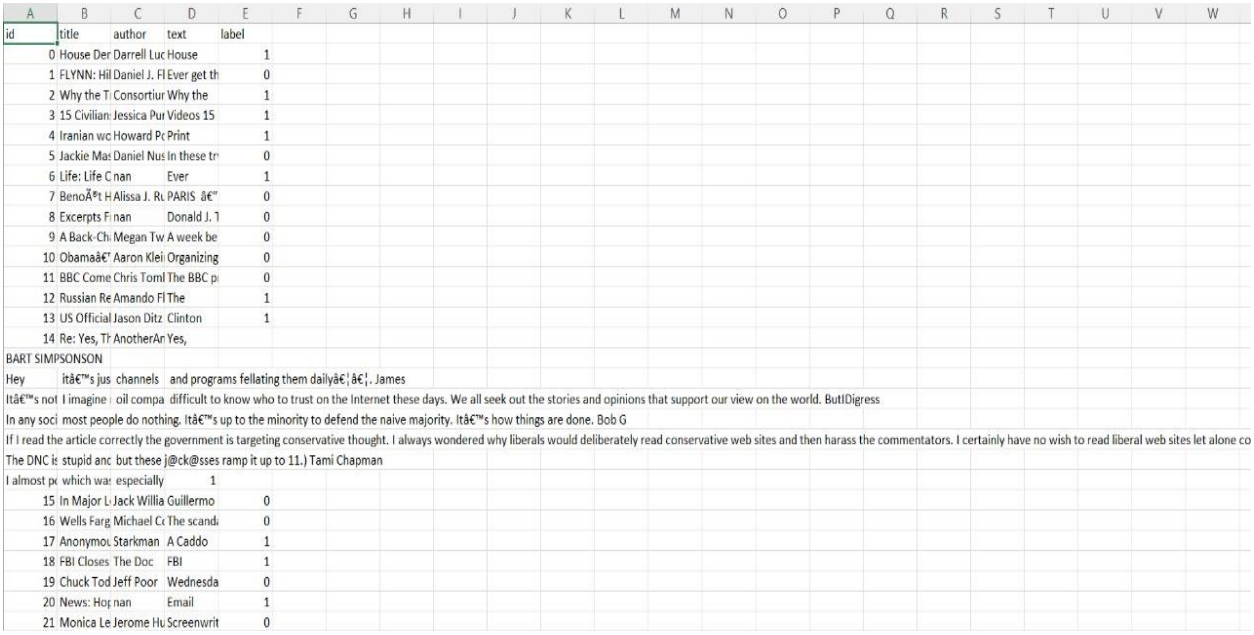
- **Reliability:**
  - Minimize system downtime by implementing robust error handling and recovery mechanisms.
  - Regularly test the system for potential issues.

- **Portability:**
  - Ensure compatibility with various devices.
  - Optimize for different screen sizes and resolutions.

5. CODE AND OUTPUT:

5.1 Python code snapshots

- **Dataset**



id	title	author	text	label
0	House Der	Darrell Luc	House	1
1	FLYNN: Hil	Daniel J. Fl	Ever get th	0
2	Why the Ti	Consortiur	Why the	1
3	15 Civilian	Jessica Pur	Videos 15	1
4	Iranian wc	Howard Pc	Print	1
5	Jackie Mas	Daniel Nus	In these tr	0
6	Life: Life C	nan	Ever	1
7	BenoÃ®t H	Alissa J. Rt	PARIS â€"	0
8	Excerpts Fi	nan	Donald J. T	0
9	A Back-Ch	Megan Tw	A week be	0
10	Obamaâ€™	Aaron Klei	Organizing	0
11	BBC Come	Chris Toml	The BBC p	0
12	Russian Re	Amando Fl	The	1
13	US Official	Jason Ditz	Clinton	1
14	Re: Yes, Th	AnotherAn	Yes,	
BART SIMPSONSON				
Hey	itâ€™s jus	channels	and programs fellating them dailyâ€¦â€¦. James	
Itâ€™s not	I imagine	oil compa	difficult to know who to trust on the Internet these days. We all seek out the stories and opinions that support our view on the world. ButIDigress	
In any soci	most people	do nothing. Itâ€™s up to the minority to defend the naive majority. Itâ€™s how things are done. Bob G		
If I read the article correctly the government is targeting conservative thought. I always wondered why liberals would deliberately read conservative web sites and then harass the commentators. I certainly have no wish to read liberal web sites let alone co				
The DNC is	stupid anc	but these	j@ck@sses ramp it up to 11.) Tami Chapman	
I almost pc	which wa	especially	1	
15	In Major L	Jack Willia	Guillermo	0
16	Wells Farg	Michael Cc	The scand	0
17	Anonymou	Starkman	A Caddo	1
18	FBI Closes	The Doc	FBI	1
19	Chuck Tod	Jeff Poor	Wednesda	0
20	News: Hoy	nan	Email	1
21	Monica Le	Jerome Hu	Screenwrit	0

- **Data Preprocessing**

DataPrep.py > ...

```
import os
import pandas as pd
import csv
import numpy as np
import nltk
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer
from nltk.stem.porter import PorterStemmer
from nltk.tokenize import word_tokenize
import seaborn as sb
```

- **Feature selection**

FeatureSelection.py > ...

```
import DataPrep
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
import nltk
nltk.download('treebank')
import nltk.corpus
from nltk.tokenize import word_tokenize
from gensim.models.word2vec import Word2Vec
```

- **Classifier and prediction**

```
import DataPrep
import FeatureSelection
import numpy as np
import pandas as pd
import pickle
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import  LogisticRegression
from sklearn.linear_model import SGDClassifier
from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import KFold
from sklearn.metrics import confusion_matrix, f1_score, classification_report
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import learning_curve
import matplotlib.pyplot as plt
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import average_precision_score
```

```
#==============================================================================
#Bag of words confusion matrix and F1 scores

#Naive bayes
# [2118 2370]
# [1664 4088]
# f1-Score: 0.669611539651

#Logistic regression
# [2252 2236]
# [1933 3819]
# f1-Score: 0.646909097798

#svm
# [2260 2228]
# [2246 3506]
#f1-score: 0.610468748792

#sgdclassifier
# [2414 2074]
# [2042 3710]
# f1-Score: 0.640874558778

#random forest classifier
# [1821 2667]
# [1192 4560]
# f1-Score: 0.702651511011
#==============================================================================
```

```
#==========================================================================================
#n-grams & tfidf confusion matrix and F1 scores

#Naive bayes
# [841 3647]
# [427 5325]
# f1-Score: 0.723262051071

#Logistic regression
# [1617 2871]
# [1097 4655]
# f1-Score: 0.70113000531

#svm
# [2016 2472]
# [1524 4228]
# f1-Score: 0.67909201429

#sgdclassifier
# [  10 4478]
# [  13 5739]
# f1-Score: 0.718731637053

#random forest
# [1979 2509]
# [1630 4122]
# f1-Score: 0.665720333284
#==========================================================================================
```

- **Final Model**
  After using different classifier, we concluded with Naïve byes with highest accuracy. So, we moved forward using naïve byes classifier.

```
train_df.head(15)
```
✓ 0.0s

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1
5	5	Jackie Mason: Hollywood Would Love Trump if He...	Daniel Nussbaum	In these trying times, Jackie Mason is the Voi...	0
6	6	Life: Life Of Luxury: Elton John's 6 Favorite ...	NaN	Ever wonder how Britain's most iconic pop pian...	1
7	7	Benoît Hamon Wins French Socialist Party's Pre...	Alissa J. Rubin	PARIS — France chose an idealistic, traditi...	0
8	8	Excerpts From a Draft Script for Donald Trump'...	NaN	Donald J. Trump is scheduled to make a highly ...	0
9	9	A Back-Channel Plan for Ukraine and Russia, Co...	Megan Twohey and Scott Shane	A week before Michael T. Flynn resigned as nat...	0
10	10	Obama's Organizing for Action Partners with So...	Aaron Klein	Organizing for Action, the activist group that...	0
11	11	BBC Comedy Sketch "Real Housewives of ISIS" Ca...	Chris Tomlinson	The BBC produced spoof on the "Real Housewives...	0
12	12	Russian Researchers Discover Secret Nazi Milit...	Amando Flavio	The mystery surrounding The Third Reich and Na...	1
13	13	US Officials See No Link Between Trump and Russia	Jason Ditz	Clinton Campaign Demands FBI Affirm Trump's Ru...	1
14	14	Re: Yes, There Are Paid Government Trolls On S...	AnotherAnnie	Yes, There Are Paid Government Trolls On Socia...	1

5.2 Output and evaluation

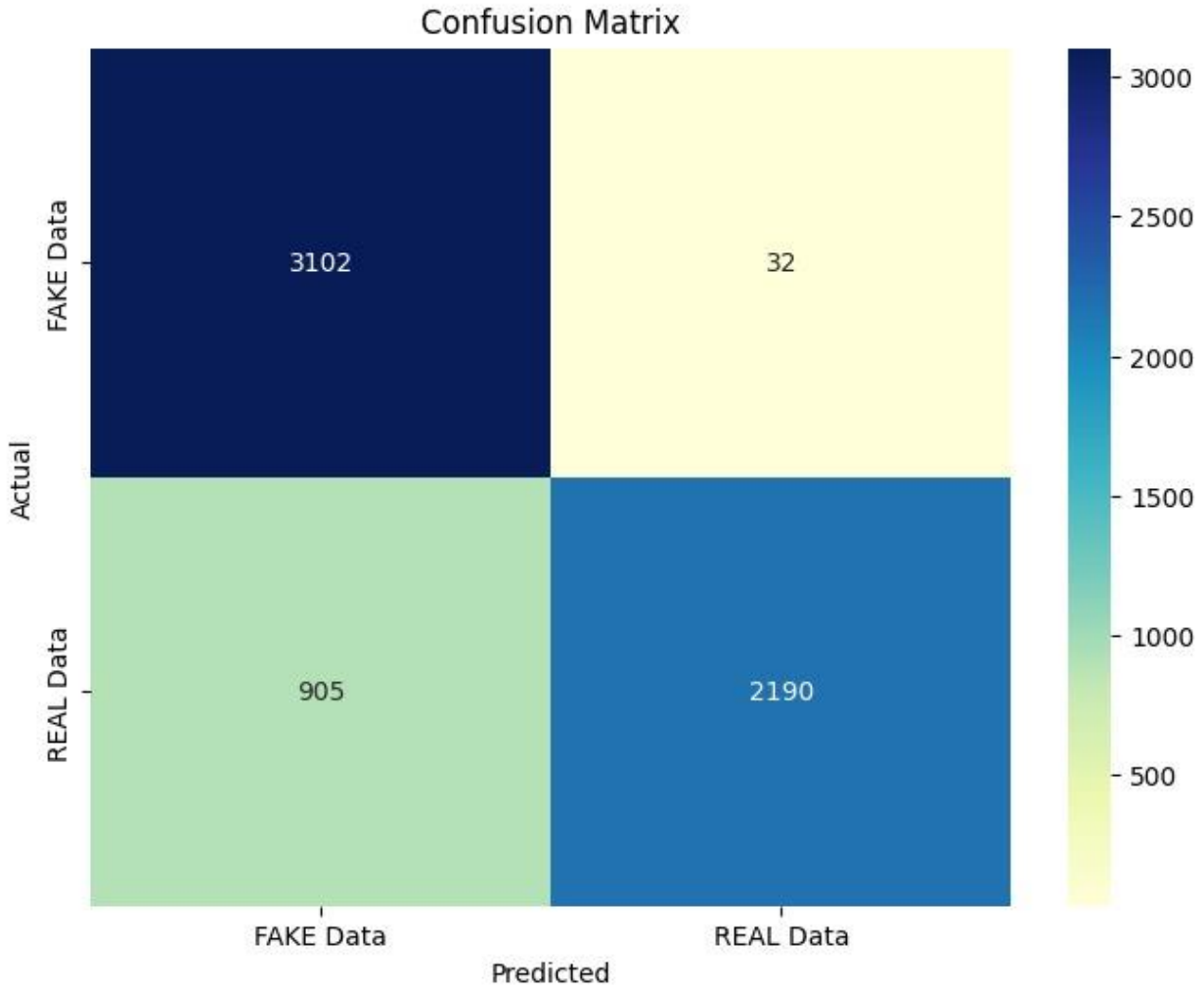
```
Y_pred = classifier.predict(tfidf_X_test)
score = metrics.accuracy_score(Y_test, Y_pred)
print(f'Accuracy: {round(score*100,2)}%')

cm = metrics.confusion_matrix(Y_test, Y_pred)

plt.figure(figsize=(8,6))
sns.heatmap(cm, annot=True, fmt='d', cmap="YlGnBu", xticklabels=['FAKE Data', 'REAL Data'], yticklabels=['FAKE Data', 'REAL Data'])
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion Matrix')
plt.show()
```

✓ 0.1s

```
Accuracy: 84.96%
```




```
input_news = "Wisconsin is on pace to double the number of layoffs this year."
prediction_result = fake_news_det(input_news, loaded_model, tfidf_v, stpwrds)
print("Prediction:", prediction_result)
```

✓ 0.0s Python

```
Prediction: Looking Fake⚠ News
```