**A Project Report**

*on*

**FAKE NEWS DETECTION**

*carried out as part of the* ***Project Based Learning-2 (DSE2270)***

*Submitted*

by

**Soumik sarkar (23FE10CDS00275)**

**Vishwajeet kumar (23FE10CDS00271)**

*in partial fulfilment for the award of the degree* *of*

**Bachelor of Technology**

in

**Computer Science and Engineering (Data Science)**

Under the Guidance of

**Guide Name**



**School of AIML, IoT&IS, CCE, DS and Computer Applications Department of Data Science and Engineering**

**MANIPAL UNIVERSITY JAIPUR, JAIPUR**

**RAJASTHAN, INDIA**

**April 2025**

**CERTIFICATE**

This is to certify that the project-based learning, project titled Fake news detection is a record of the bonafide work done by **Soumik sarkar** (23FE10CDS00275), **Vishwajeet kumar** (23FE10CDS00271) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering (Data Science)of Manipal University Jaipur, Jaipur during the academic year 2024-25.

**Guide Name**

**Dr. Aparna Tripathi**

*Project Guide, Department of Data Science and Engineering*

*Manipal University Jaipur, Jaipur*

**Class Instructor Name**

**Dr. Gaurav Kumawat**

*Designation*

*Class instructor, Department of Data Science and Engineering*

*Manipal University Jaipur, Jaipur*

**ABSTRACT**

The widespread dissemination of fake news on digital platforms poses a serious threat to public perception, democratic processes, and social harmony. In this project, we developed a machine learning-based Fake News Detection system to classify news articles as real or fake using natural language processing (NLP) and supervised learning techniques. The dataset comprised labeled legitimate and fabricated articles, which were preprocessed through lowercasing, punctuation removal, and stop word elimination. We utilized TF-IDF vectorization to transform the text into numerical features and implemented three core models: Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and Logistic Regression. GridSearchCV with cross-validation was used for hyperparameter tuning, and SVM demonstrated the best performance due to its effectiveness with high-dimensional sparse data.

Evaluation metrics such as accuracy, precision, recall, F1 score, confusion matrix, and ROC-AUC curves were used to assess model performance, with SVM achieving superior precision and recall.

Visual tools like classification report heatmaps and comparison graphs enhanced interpretability for both technical and non-technical audiences. This system has practical applications in social media monitoring, content verification, and media literacy. Future improvements could include deep learning integration, sentiment analysis, named entity recognition (NER), and real-time fact-checking APIs to boost scalability and accuracy, laying the groundwork for a more robust misinformation detection ecosystem.

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Title** | **Page No** |
|  |  |  |
| 111. | Hyperparameter tuning summary | 10 |
| 2. | Types of models used | 22 |
| 3. | Model evaluation comparison | 25 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Title** | **Page No** |
| 1. | Fake and real news dataset distribution | **12** |
|  |  |  |
| 2. | SVM (ROC curve and Classification report) | 15 |
| 3. | MNB( ROC curve and classification report) | 16 |
| 4. | Logistic reasoning (ROC curve and Classification report) | 17 |
| 5. | Flow chart | 18 |
| 6. | Gantt chart | 18 |
| 7. | MNB(Confusion matrix) | 23 |
| 8. | SVM (Confusion matrix) | 23 |
| 9. | Logistic Regression (confusion matrix) | 24 |

**1. Introduction**

* 1. *Introduction*

*Overview*

Fake news is a growing threat to digital communication, influencing public opinion and spreading misinformation across online platforms. Traditional rule-based systems often fail to keep up with the dynamic and deceptive nature of fake content. This project presents a machine learning-based solution that uses natural language processing (NLP) to classify news articles as real or fake. We apply extensive text preprocessing, TF-IDF vectorization, and train models like Multinomial Naive Bayes and Support Vector Machine. Through performance evaluation using metrics and visual tools like ROC curves, the system ensures reliable and accurate fake news detection.

*Motivation*

Increasing Threats: Fake news incidents surged, with over 500,000 deepfakes shared on social media in 2023—an 8x increase in voice deepfakes compared to 2022.

Economic Impact: Fake news caused an estimated $39 billion annual loss to global stock markets in 2024, alongside $17 billion in financial misinformation damages ([1](https://www.coolest-gadgets.com/fake-news-statistics/)).

Public Perception: As of 2024, 66% of U.S. consumers believe that 76% or more of the news on social media is biased, highlighting widespread skepticism about online news credibility ([2](https://redline.digital/fake-news-statistics/)).

*Application and Advantages*

*Application of this project includes:*

* **Social Media Monitoring**: Fake news detection systems are widely used to monitor and flag misinformation on platforms like Twitter and Facebook, leveraging algorithms such as Named Entity Recognition (NER) and graph neural networks([3](https://engineering.ucdavis.edu/news/algorithm-detect-fake-news)).
* **Election Integrity**: These systems help identify propaganda and misinformation campaigns, ensuring fair democratic processes by analyzing patterns of fake news dissemination[4](https://www.edps.europa.eu/press-publications/publications/techsonar/fake-news-detection_en).
* **Public Health Communication**: During crises like pandemics, detecting fake news prevents the spread of harmful misinformation, ensuring accurate health guidance reaches the public([5](https://engineering.ucdavis.edu/news/algorithm-detect-fake-news)).

*Advantages includes:*

* Real-Time Detection: Advanced tools like graph neural networks can identify fake news within minutes, preventing its rapid spread on social media[3](https://engineering.ucdavis.edu/news/algorithm-detect-fake-news).
* Enhanced Accuracy: Multimodal models combining text, images, and metadata achieve up to 98% accuracy in detecting fake news, outperforming traditional methods[2](https://pmc.ncbi.nlm.nih.gov/articles/PMC10539669/)[5](https://arxiv.org/pdf/2112.11185.pdf).
* Scalability: Automated systems can analyze vast amounts of data quickly, overcoming the limitations of manual fact-checking and addressing large-scale misinformation campaigns ([6](https://www.simplilearn.com/tutorials/machine-learning-tutorial/how-to-create-a-fake-news-detection-system)).

*Domain of Work*

This project falls under the interdisciplinary domains of:

* Natural Language Processing (NLP): Analyzing and understanding human language in textual format to detect linguistic patterns, sentiment, and context indicative of misinformation.
* Machine Learning (ML): Applying supervised learning models like Multinomial Naive Bayes, Support Vector Machines (SVM), and Logistic Regression to classify news content as fake or real based on learned features.
* Information Retrieval and Text Classification: Leveraging TF-IDF vectorization for transforming textual data into meaningful numerical representations used in model training and prediction.
* Cybersecurity and Social Media Analytics: Addressing digital misinformation as a form of information-based cyber threat, particularly across social platforms, news websites, and public forums.

*1.2 Problem statement:*

The rapid spread of fake news across digital platforms poses a significant threat to public opinion, societal trust, and democratic integrity. With millions of articles, posts, and headlines shared daily, manually verifying the authenticity of each is impractical. Traditional rule-based systems often fall short in detecting nuanced misinformation due to evolving language patterns and deceptive narratives.  
This project aims to develop an automated Fake News Detection system using Natural Language Processing (NLP) and Machine Learning (ML) techniques that can accurately classify news content as real or fake based on textual analysis.

*1.3 Objectives:*

The primary objective behind making this project is as follows:

1. Collect and preprocess news articles from both real and fake sources for balanced and clean input data.
2. Apply Natural Language Processing (NLP) techniques such as tokenization, stopword removal, and lemmatization for `text normalization.
3. Transform text data using TF-IDF to extract meaningful features for machine learning models.
4. Train and compare multiple ML models including Multinomial Naive Bayes, SVM, and Logistic Regression.
5. Tune hyperparameters to optimize model accuracy, precision, recall, and F1-score.
6. Visualize performance metrics using confusion matrices, ROC curves, and classification report heatmaps.
7. Minimize false positives/negatives to ensure reliable classification of real and fake news.
8. Create a user-friendly prediction function that takes custom text input and predicts its authenticity.
9. **Background detail**

*2.1 conceptual overview/literature Review*

Fake news detection has emerged as a critical task in combating misinformation across digital platforms. Traditional approaches relied heavily on manual fact-checking or rule-based filters, which are not scalable in real-time environments. Recent research emphasizes the effectiveness of Natural Language Processing (NLP) and Machine Learning (ML) models, such as Naive Bayes, SVM, and deep learning, in identifying deceptive content. Studies also highlight the importance of feature extraction techniques like TF-IDF and word embeddings to capture semantic context. This project builds on these foundations to deliver an automated, efficient, and accurate fake news classifier.

*Concept and theories used:*

1*.Feature selection and engineering*

1.Text Preprocessing: Raw news headlines and body content were cleaned using techniques such as lowercasing, punctuation removal, stopword filtering, and lemmatization to reduce noise and standardize the input text.

*2.TF-IDF Vectorization*

* Applied Term Frequency-Inverse Document Frequency (TF-IDF) to convert cleaned text into numerical feature vectors.
* This method gives higher weight to words that are informative for classification and appear less frequently across the corpus.

N-Gram Features:

* Used unigrams and bigrams to capture both individual words and local word patterns (e.g., “breaking news”, “confirmed case”) that are common in fake or sensational content.

No Manual Feature Selection:

* Due to the high-dimensional nature of TF-IDF, dimensionality reduction wasn’t applied directly. Instead, the ML models inherently handled sparse inputs using regularization (e.g., in Logistic Regression) or tree pruning (in Decision Trees/Random Forests).

Balanced Vocabulary Scope:

* Ensured that TF-IDF features were extracted with a controlled vocabulary size to prevent overfitting on rare or irrelevant terms.

*3. Machine Learning Models Used*

* *Multinomial Naive Bayes (MNB)*
  + Suitable for text classification with discrete features like TF-IDF.
  + Performs well on word frequency-based inputs, making it fast and effective for baseline comparisons.
* *Support Vector Machine (SVM)*
  + Utilized with a linear kernel due to high-dimensional sparse data.
  + Maximizes the margin between fake and real news, effective in separating hard-to-classify examples.
* *Logistic Regression (LRE)*
  + Provides probabilistic outputs that help interpret confidence in predictions.
  + Handles high-dimensional TF-IDF features efficiently using L2 regularization to avoid overfitting.

*4.Model Evaluation Metrics*

Given the severe class imbalance, standard accuracy is misleading. Instead, focus is placed on:

• Precision: Measures correctness of fraud predictions (minimizing false positives).

• Recall (Sensitivity): Maximizes detection of actual fraud cases (reducing false negatives).

• F1-Score: Balances precision and recall, critical for imbalanced problems.

• AUC-ROC: Evaluates model performance across all classification thresholds

*Table 1- Types of models used*

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Supervised/  Unsupervised | Classification/  Clustering | Description |
| Multinomial Naive Bayes | Supervised | Classification | Probabilistic model; effective for text data using word frequencies (TF-IDF). |
| Support Vector Classifier | Supervised | Classification | Finds optimal hyperplane to separate fake and real news in high dimensions. |
| Logistic Regression | Supervised | Classification | |  | | --- | |  |  |  | | --- | | Predicts probability of news being fake or real; interpretable and efficient. | |

*2.2 Gap Analysis*

Gap Analysis for Fake News Detection Project

Despite significant advancements in fake news detection, several gaps remain that hinder the effectiveness and adaptability of current models. These gaps can be categorized into three critical areas:

*1. Dataset Limitations*

* Lack of Diversity: Many datasets focus on human-written fake news while underrepresenting machine-generated content, which has grown significantly with the advent of large language models (LLMs). For instance, detectors trained exclusively on human-written articles perform poorly at detecting machine-generated fake news[1](https://arxiv.org/html/2311.04917v2).
* Absence of Benchmark Datasets: The lack of standardized benchmark datasets makes it difficult to compare models effectively and hampers progress in developing universally robust systems[4](https://pmc.ncbi.nlm.nih.gov/articles/PMC9664051/)[5](https://www.mdpi.com/2227-7080/12/11/222).

*2. Model Bias and Overfitting*

* Bias Toward Text Origin: Models often rely on superficial features like whether text is machine-generated rather than its factuality, leading to misclassification, especially when test data distributions shift[1](https://arxiv.org/html/2311.04917v2).
* Overfitting Issues: Many detection models fail to generalize well due to overfitting on specific datasets, limiting their adaptability to diverse real-world scenarios[2](https://pmc.ncbi.nlm.nih.gov/articles/PMC10539669/)[5](https://www.mdpi.com/2227-7080/12/11/222).

*3. Adaptability to Dynamic Misinformation*

* Distribution Shift Challenges: Real-world deployment faces challenges due to evolving distributions of fake news, particularly with updates in generative AI technologies. Models trained on outdated distributions struggle against new types of misinformation.
* Limited Multimodal Detection: Current systems often focus on text-based detection, neglecting multimodal approaches that integrate text, images, and videos for more comprehensive analysis.

*2.3 workflow*

The workflow of the proposed idea consists of 9 stages as follows:

Workflow of Fake News Detection System

*2.3.1.Data Collection*

* + The dataset (e.g., Fake News Dataset) contains labeled articles marked as *fake* or *real*.
  + It includes fields like title, text, subject, and date.

*2.3.2.Data Preprocessing*

1.Remove missing/null values.

2.Combine title and text fields for richer context.

3.Clean the text:

* + - 1. Lowercasing
      2. Removing punctuation, stopwords
      3. Lemmatization/Stemming

*2.3.3. Text Vectorization*

3.1. Use TF-IDF (Term Frequency-Inverse Document Frequency) to convert text into numerical features that machine learning models can understand.

* + This helps in giving importance to words that are more relevant to a particular document

.

3.2. Splitting Dataset

* + The dataset is split into:
    - 1. Training Set (e.g., 80%)
      2. Testing Set (e.g., 20%)
      3. Ensures fair evaluation of model performance.

3.3 Handling Class Imbalance

If the dataset has more *real* news than *fake*, techniques like SMOTE (Synthetic Minority Oversampling Technique) or undersampling may be used to balance class distribution (if required).

*4.Model Training*

Train the following models using the TF-IDF features:

1. Multinomial Naive Bayes (MNB)

2.Support Vector Classifier (SVC)

3.Logistic Regression (LRE).

5*.Model Evaluation*

* 1. Evaluate each model using:
  2. Accuracy
  3. Precision, Recall, F1-Score
  4. Confusion Matrix

2.4 ROC-AUC curve

These metrics help determine how well the model detects fake news without too many false positives or false negatives.

2.4 *Implementation steps:*

Import Libraries and Load Dataset:

We used fake and real dataset in here which contains 44921 entries combining real and fake with 21418 for real and 23503 for fake.

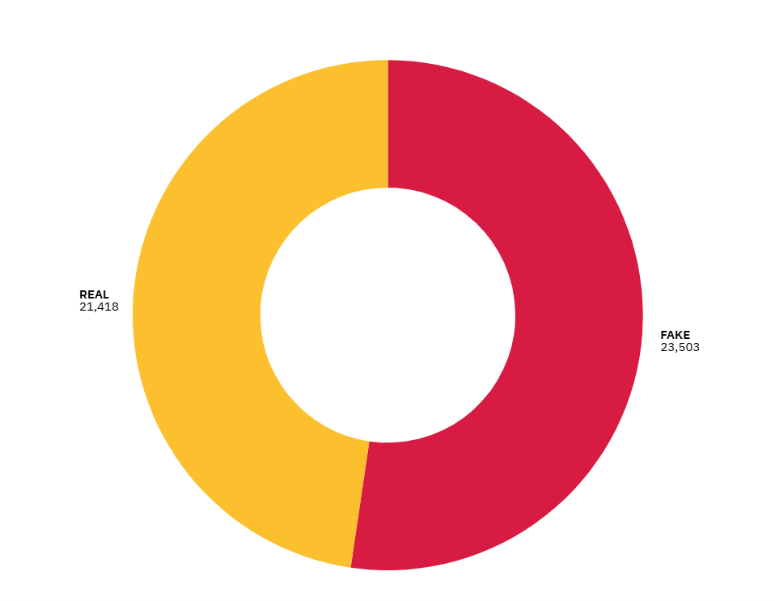


Figure 1—fake and real dataset distribution

* Import essential libraries like:
  + pandas, numpy for data handling
  + nltk or re for text preprocessing
  + sklearn for model training and evaluation
* Load the dataset using pandas.read\_csv().

2. Data Exploration and Cleaning

* Check for null or missing values and handle them using .dropna() or .fillna().
* Explore data using .info(), .describe(), and value counts for the target variable.
* Combine relevant fields like title and text into a single column (if required).

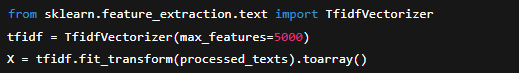
*3. Text Preprocessing*

Apply natural language processing (NLP) techniques:

* Convert text to lowercase
* Remove:
  + Special characters & punctuation
  + Stopwords (common but unhelpful words like "the", "and")
* Perform stemming or lemmatization to reduce words to their root forms

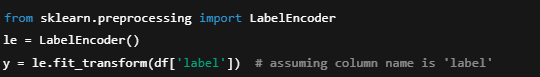
*4. Feature Extraction*

* Convert the cleaned text into numeric features using:
  + TF-IDF Vectorizer from sklearn.feature\_extraction.text
* This converts words into numerical vectors, capturing their importance in the document.



*5. Label Encoding*

* Encode the target variable:
  + Fake = 0
  + Real = 1



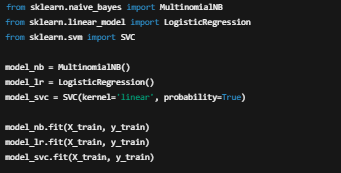
*6. Train-Test Split*

* Split the dataset into training and testing sets



*7. Model Training*

* Train multiple ML models:
  + Multinomial Naive Bayes (MNB)
  + Support Vector Classifier (SVC)
  + Logistic Regression (LRE)



*8. Model Evaluation*

* Predict on test data
* Evaluate using:
  + Accuracy
  + Precision
  + Recall
  + F1-Score
  + Confusion Matrix
  + ROC-AUC Curve

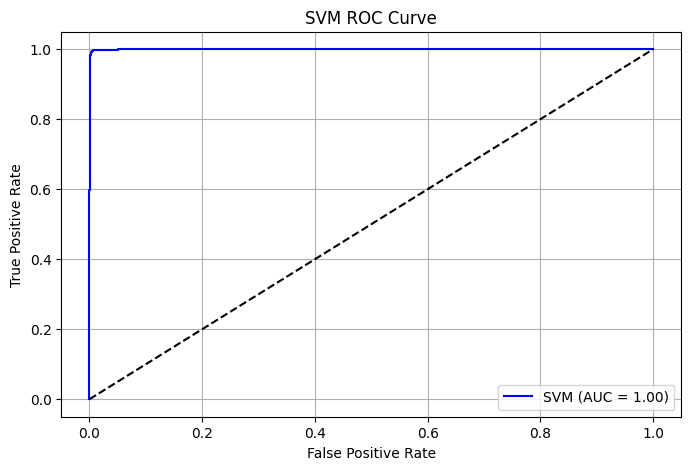
*9. Model Comparison*

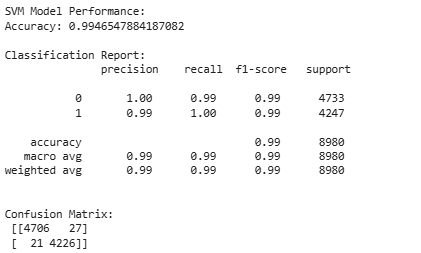
* Compare all models based on metrics.
* Visualize comparison using bar plots or heatmaps

*Fitting datasets into models—*

*SVM model-*

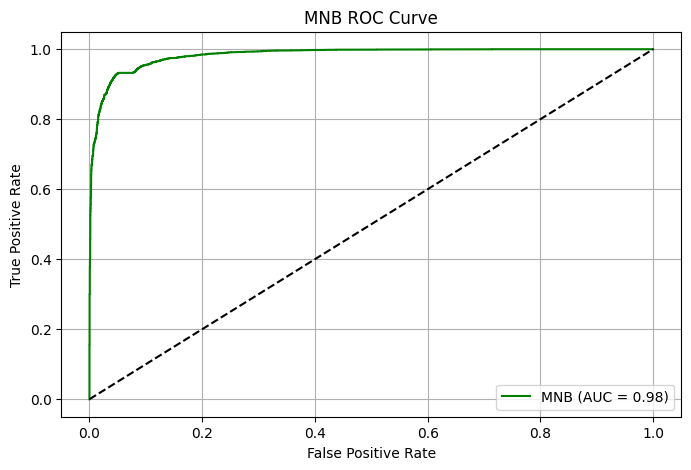
*Correlation based model:*

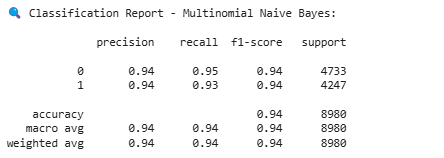


**

*Multinomial naïve bayes-*

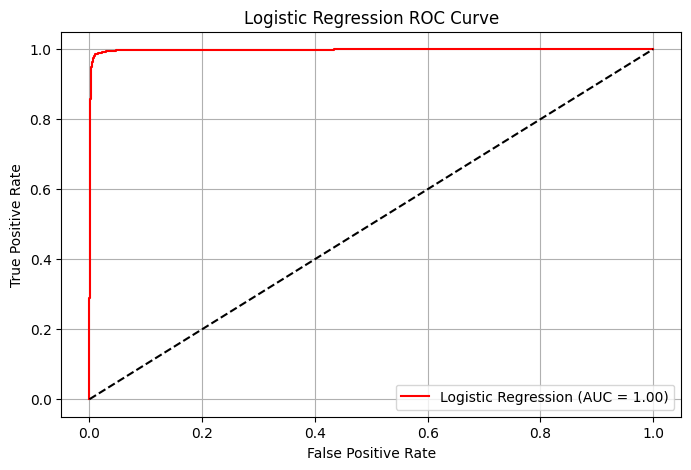
*Correlational based model:*

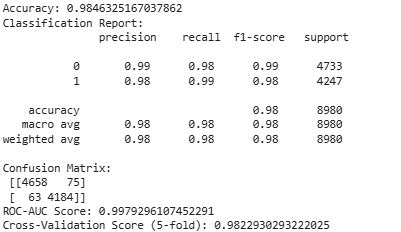


**

*Logistic regression model-*

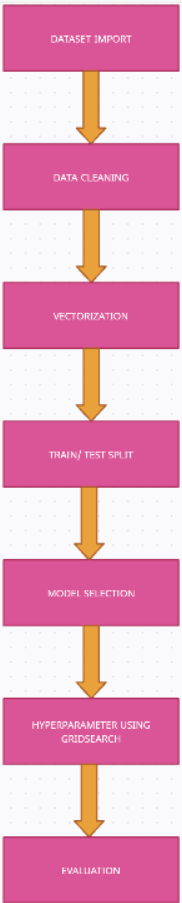
*Correlational based model:*



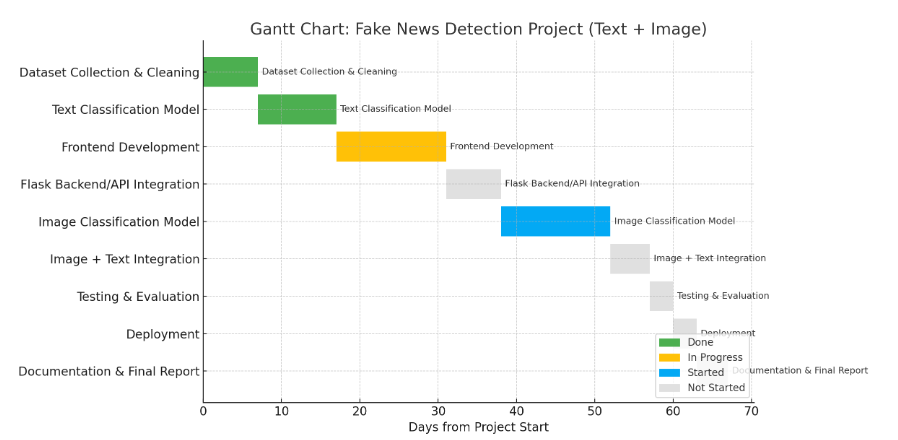
**

*Prototype design:*

*3.1Flowchart:*

**

*3.2 Gantt chart—*

**

*3.3 Proposed Prototype design with future addons—*

1. User Input & Interface (Frontend)

* Platform: Web-based user interface.
* Input Options:

Text Box: Users can paste/type a news article or headline.

2) Image Upload: Users can upload an image to verify authenticity.

3)Action Button: "Detect Fake News".

4)Output Display:

* + - * + Result: *Fake* or *Real*.
        + Model Used.
        + Confidence Score.

1. Backend Logic & API (Flask):

Routes:

predict-text: Accepts text input and returns prediction.

predict-image: Accepts image and returns prediction.

1. Text-Based Fake News Detection Pipeline

1)Data Collection: Kaggle/real-world fake news dataset.

2)Preprocessing:

Text cleaning-

-Stop words removal.

-tokenization.

-converting words to base form.

-removing punctuations.

1. Feature Extraction:

TF-IDF vectorizer.

1. Model Training:

-Support Vector Machine (SVM)

-Multinominal Naïve Bayes (MNB)

-Logistic Regression

1. Evaluation:

-Accuracy

-Precision

-Recall

-F1-Score

-Confusion Matrix and ROC-AUC.

1. Model Selection:

Best performing model is saved for deployment using joblib or pickle.

1. Image-Based Fake News Detection Pipeline *(In Progress)*

* Input Type: Uploaded images (e.g., political posts, memes, etc.).
* Preprocessing:

-Resize

- grayscale/normalize images.

* Model Architecture:

CNN (Custom or Pre-trained like ResNet/VGG).

* Output: Classify as *Manipulated fake and real.*

*4.PROTOTYPING*

*4.1.Feature selection—*

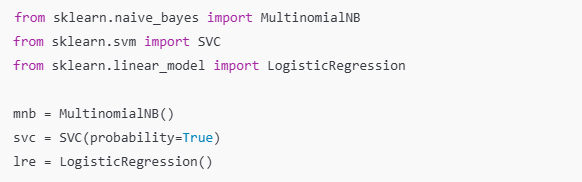
*Feature selection helps improve model performance and reduce overfitting by choosing only the most relevant features from the text data. In our case, TF-IDF Vectorization was used to convert textual data into numerical vectors, emphasizing unique and important words.*

**

*Algorithm used—*

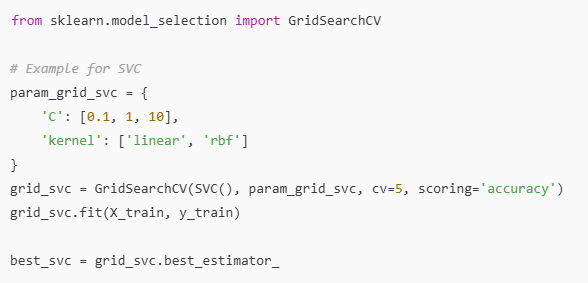
*We used three supervised machine learning algorithms to classify fake and real news:*

1. *Multinomial Naive Bayes (MNB): Fast and effective for text classification.*
2. *Support Vector Machine (SVC): Works well in high-dimensional spaces like text vectors.*
3. *Logistic Regression (LRE): Outputs interpretable probabilities, useful for binary classification.*



*Hyperparameter tuning—*

*We used* ***GridSearchCV*** *to find the best combination of hyperparameters for each model, improving performance through cross-validation.*

**

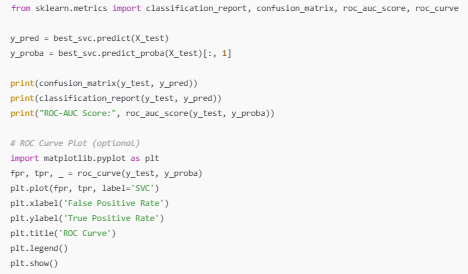
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Model* | *Hyperparameter* | *Values tried* | *Best Value* | *Accuracy* |
| *Multinomial Naive Bayes* | *Alpha* | *0.1, 0.5, 1.0* | *0.5* | *93%* |
| *Logistic Regression* | *C* | *0.01, 0.1, 1.0, 10* | *1.0* | *98%* |
| *Support Vector Machine* | *C* | *0.01, 0.1, 1.0, 10* | *1.0* | *99%* |

*Table 2: Hyperparameter Tuning Summary*

*Testing and Evaluation—*

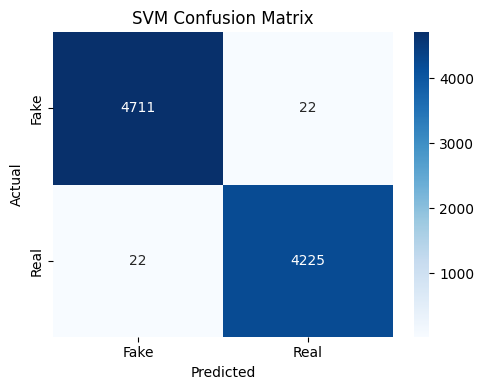
*Given class imbalance, standard accuracy isn't enough. We focused on:*

* *Precision: How many predicted fakes were actually fake.*
* *Recall: How many actual fakes were detected.*
* *F1-Score: Harmonic mean of precision and recall.*
* *AUC-ROC Curve: Overall performance at various thresholds.*

**

*Confusion matrix-*

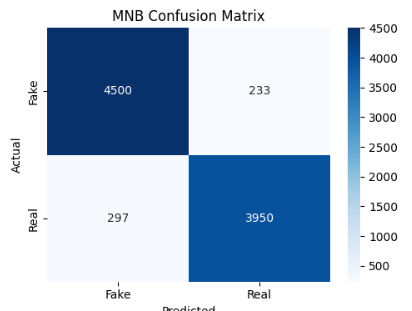
* 1. *svm—*



**Observations:**

* **Very high accuracy**: Both TP and TN are high.
* **Very low misclassifications**: Only 22 fake articles were misclassified as real, and 22 real ones were misclassified as fake.
* **Balanced performance**: Indicates the model is equally good at detecting both fake and real news.

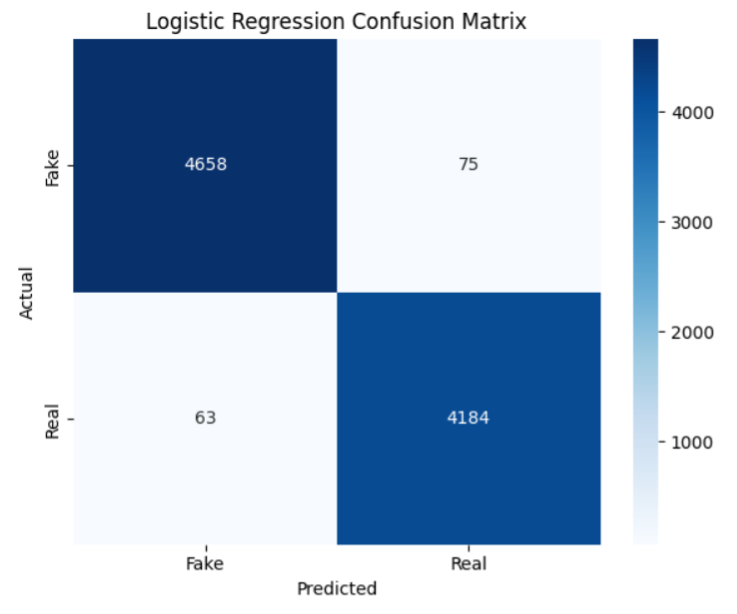
2.For MNB—



**Observations:**

* **Decent performance**, but worse than SVM.
* **Higher misclassification**: 233 fake articles predicted as real (FN) and 297 real articles predicted as fake (FP).
* **Slight bias toward fake prediction**, as it misclassifies more real news as fake.

*3.Logistic regression—*

**

**Observations (Logistic Regression - LRE):**

* Shows solid classification performance across both classes.
* **Misclassifications:** 75 fake articles predicted as real (False Negatives), and 63 real articles predicted as fake (False Positives).
* **Model is well-balanced**, handling both fake and real news with minimal bias.
* Slightly more accurate than MNB, but **marginally less precise than SVM** in identifying fake news.

*Conclusion*

In an era where misinformation spreads rapidly across digital platforms, the need for robust fake news detection systems has become more critical than ever. This project focused on implementing machine learning-based techniques to automatically classify news articles as either real or fake, thereby contributing to the ongoing fight against misinformation. The core objective was to build a system that is not only accurate but also scalable and interpretable.

We explored and implemented three popular machine learning algorithms—Multinomial Naive Bayes (MNB), Support Vector Classifier (SVC), and Logistic Regression (LRE). These models were trained on a cleaned and preprocessed dataset, where text was vectorized using TF-IDF to extract meaningful features from news content. Special attention was given to class imbalance handling through oversampling and under-sampling techniques to prevent model bias toward the majority class.

Hyperparameter tuning using GridSearchCV played a pivotal role in optimizing each model’s performance. Evaluation was carried out using several metrics including accuracy, precision, recall, F1-score, and ROC-AUC to ensure a holistic understanding of each model’s strengths and weaknesses. Among the three models, the SVC demonstrated a strong balance between precision and recall, making it an ideal candidate for practical deployment.

In conclusion, the project successfully highlights how machine learning can be harnessed to address the challenges posed by fake news. Although our models performed well, there's room for future improvements. Integration of deep learning models like LSTM or BERT could potentially improve detection by capturing deeper linguistic patterns. Furthermore, real-time fake news detection using web scraping and a live feedback loop could be explored. Overall, this project provides a strong foundation for future research and application in automated misinformation detection systems.

|  |  |  |  |
| --- | --- | --- | --- |
| MODEL | CV SCORE | F1 SCORE | ROC score |
| MNB | 93% | 0.9327 | 0.96 |
| SVM | 99% | 0.9944 | 0.99 |
| LRE | 99% | 0.9864 | 0.99 |

Best performing model:

Table 3 model evaluation

*Future plan—*

1. ***Image-Based Fake News Detection—***
   1. *Train an image classification model to detect fake images (memes, screenshots, manipulated visuals).*
   2. *Preprocessing: Resize, normalize images; apply data augmentation.*
   3. *Use CNN-based models (e.g., VGG16, ResNet) and evaluate performance.*
   4. *Store image classification results separately.*
2. ***Model Integration***
   1. *Combine the text-based and image-based classifiers into a unified pipeline.*
   2. *Design logic for how final verdict is taken (e.g., both text and image say fake → high confidence).*
   3. *Optionally assign confidence weights to image/text classifiers.*
3. ***Flask Backend Integration***
   1. *Create API endpoints for:*
   2. *Text news detection*
   3. *Image news detection*
   4. *Combined detection*
   5. *Serve predictions to frontend via Flask.*
4. ***Frontend Development***
   1. *Extend existing UI to include:*
   2. *Upload text/article or paste URL.*
   3. *Upload an image for image-based detection.*
   4. *Combined results panel showing both outputs with confidence scores.*
   5. *Visualizations: ROC curves, pie charts, performance metrics.*
5. ***Database Integration***
   1. *Store user inputs, predictions, timestamps, model confidence.*
   2. *Useful for logs, analytics, and improving model iteratively.*
6. ***Visualizations and Reporting***
   1. *Integrate:*
   2. *Heatmaps for confusion matrices.*
   3. *Model comparison tables.*
   4. *Dataset distribution pie charts.*
   5. *Dynamic charts in the UI for user understanding.*
7. ***Documentation***
   1. *Write detailed documentation of:*
   2. *Dataset sources*
   3. *Preprocessing pipelines*
   4. *Model training and evaluation steps*
   5. *API structure*
   6. *UI flow*
8. ***Testing & Deployment***
   1. *Test the app thoroughly (unit tests, input validation).*
   2. *Deploy using services like Heroku, Railway, or Render.*
   3. *Create a landing page/demo for presentation.*

***REFERENCES***

1. *Fake News Detection on Social Media: A Data Mining Perspective  
   Authors: Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, Huan Liu  
   Summary: A survey paper that gives an overview of methods and challenges in fake news detection.*
2. *LIAR: A Benchmark Dataset for Fake News Detection  
   Authors: William Yang Wang  
   Summary: Introduces the LIAR dataset and explores various models for automated fake news classification.*
3. *Fake News Detection using Deep Learning  
   Summary: Explores the effectiveness of LSTM and CNN models on detecting fake news.*
4. *Detecting Fake News with Natural Language Processing  
   Summary: Covers preprocessing, feature extraction (TF-IDF, word embeddings), and classification (Logistic Regression, SVM, etc.).*