AI Data Science Internship Project Report

Project Title: *Fake News Detection using Machine Learning*

N K MOHAMMED RANEESH
RAJIV GANDHI INSTITUTE OF TECHNOLOGY,KOTTAYAM
mhdraneeshpop123@gmail.com

June 20, 2025

---

1. Problem Statement

This project aims to tackle the growing issue of fake news by building a machine learning model that can classify news articles as real or fake based on their content. In today's digital era, misinformation spreads rapidly through social media and online platforms, influencing public opinion and decision-making. Detecting and flagging fake news automatically is crucial to maintain information integrity, combat propaganda, and support informed decision-making.

2. Dataset Description

- Source: The dataset was obtained from Kaggle: Fake and Real News Dataset
- Number of Instances and Features:
  Fake.csv: 23,000 fake news articles
  True.csv: 21,000 real news articles
  Combined: 44,000 total instances
- Key features: title, text, subject, date
- Preprocessing Steps:
  Combined the title and text columns to form the full content.
  Removed missing values and unnecessary columns (date, subject).
  Labeled data: 0 for fake news, 1 for real news.
  Applied text vectorization using TF-IDF for feature extraction.
  Data was split into training and testing sets (80/20)

3. Methodology

- 1. Data Preprocessing:

  Loaded Fake.csv and True.csv and combined them.

  Assigned labels: 0 for fake news, 1 for real news.

  Combined the title and text columns to create a unified content column.

  Removed missing values to ensure clean data.

  Split the data into training (80

  Applied TF-IDF Vectorization to convert text into numerical feature vectors while removing common stopwords.

- 2. Models Used:

  Logistic Regression: A linear classifier effective for binary classification.

  Multinomial Naive Bayes: Well-suited for text classification tasks based on word frequency.

- 3. Training Setup:

  Used default hyperparameters for both models for baseline performance.

  Trained on TF-IDF-transformed data.

  Evaluated models using:

  Accuracy Score

  Confusion Matrix

  Classification Report (Precision, Recall, F1-score)

  Visual Plots of confusion matrices and accuracy comparison

4. Code Snippets

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

fake = pd.read_csv("Fake.csv")
real = pd.read_csv("True.csv")

fake["label"] = 0
real["label"] = 1
df = pd.concat([fake, real], ignore_index=True)

df['content'] = df['title'] + "␣" + df['text']
df = df[['content', 'label']].dropna()

X_train, X_test, y_train, y_test = train_test_split(df['content'
    ], df['label'], test_size=0.2, random_state=42)
```

Listing 1: Data Preprocessing

```
vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

Listing 2: TF-IDF Vectorization

```
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report,
   confusion_matrix, accuracy_score

lr = LogisticRegression()
lr.fit(X_train_tfidf, y_train)
lr_preds = lr.predict(X_test_tfidf)

nb = MultinomialNB()
nb.fit(X_train_tfidf, y_train)
nb_preds = nb.predict(X_test_tfidf)

print("Logistic␣Regression␣Accuracy:", accuracy_score(y_test,
   lr_preds))
print(confusion_matrix(y_test, lr_preds))

print("Naive␣Bayes␣Accuracy:", accuracy_score(y_test, nb_preds))
print(confusion_matrix(y_test, nb_preds))
```

Listing 3: Logistic Regression and Naive Bayes

5. Results and Evaluation

Classification Performance Comparison Table

Table 1: Classification Metrics for Fake News Detection Models

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.98	0.98	0.98	0.98
Naive Bayes	0.93	0.93	0.93	0.93

Confusion Matrix Visualizations

To better understand the classification results, confusion matrices were generated for both models. These plots help visualize true positives, false positives, true negatives, and false negatives.

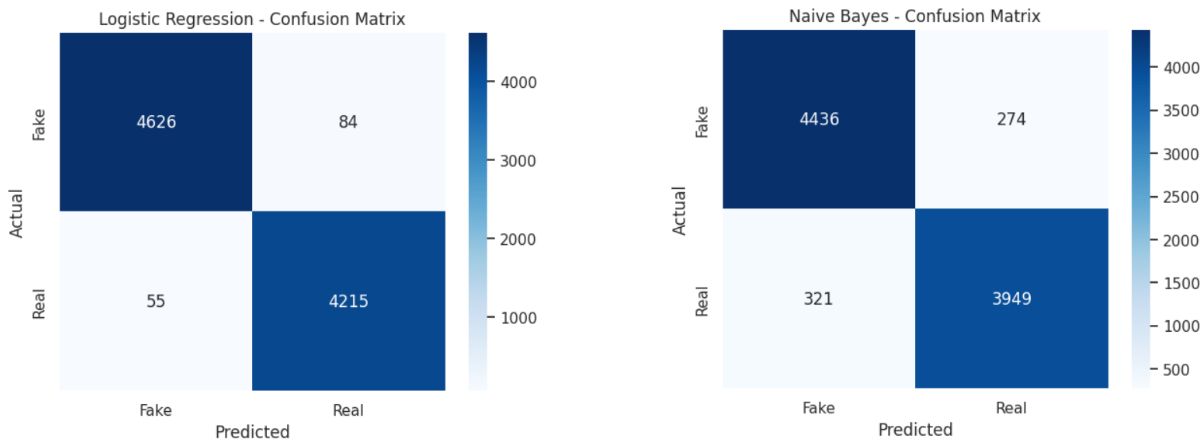

Figure 1: Confusion Matrices: Logistic Regression and Naive Bayes

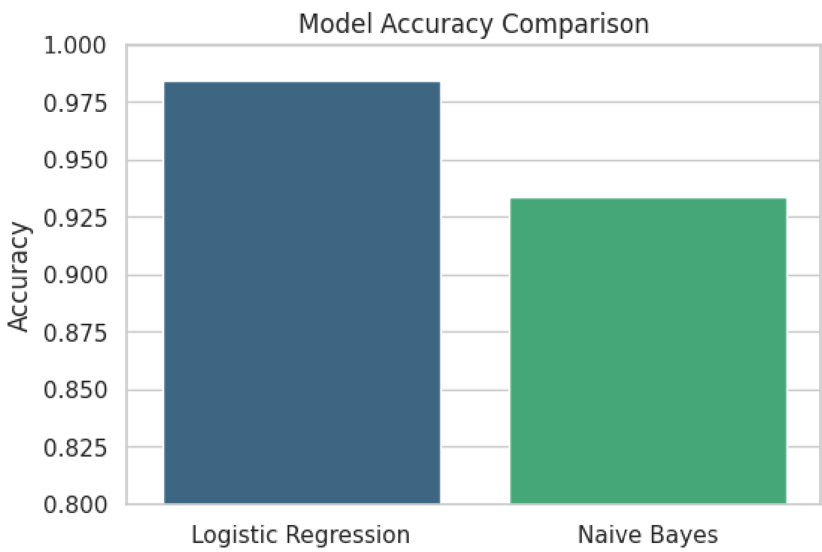

Figure 2: Model Accuracy Comparison

6. Challenges Faced

- **Handling Textual Data for Machine Learning**
  Challenge: The raw dataset contained long-form unstructured text split across `title` and `text` columns.
  Solution: Combined both columns into a new `content` feature to retain more context for classification, improving model understanding of news semantics.

- **Preprocessing and Vectorizing Large Text Data**
  Challenge: TF-IDF vectorization produced a high-dimensional sparse matrix, which can lead to memory usage issues or overfitting.
  Solution: Used `TfidfVectorizer(stop_words='english', max_df=0.7)` to:
  - Remove common English stopwords
  - Ignore very frequent terms (appearing in more than 70% of documents)

  - Reduce dimensionality while keeping meaningful features

- **Choosing the Right Classification Models**
  Challenge: Selecting models that perform well on sparse, high-dimensional data.
  Solution: Tried both Logistic Regression and Multinomial Naive Bayes, which are known to work well on text classification. This allowed you to compare linear vs. probabilistic approaches effectively.

- **Evaluating Model Performance Beyond Accuracy**
  Challenge: Accuracy alone doesn't tell the full story, especially when misclassifying fake news has serious implications.
  Solution: Printed and analyzed:
  - Confusion matrices
  - Precision, recall, and F1-scores using `classification_report()`
  - Created heatmaps and accuracy comparison plots to visualize differences between models

- **Working in Google Colab Environment**
  Challenge: Uploading and handling local datasets (like `Fake.csv` and `True.csv`) in a cloud notebook.
  Solution: Used `files.upload()` from `google.colab` to upload CSV files directly into the notebook for preprocessing and training.

7. Conclusion and Future Scope

Conclusion:
In this project, we successfully developed a machine learning-based fake news detection system using Natural Language Processing techniques. By combining the `title` and `text` fields of news articles, and applying TF-IDF vectorization, we trained and evaluated two classification models: Logistic Regression and Multinomial Naive Bayes. Among them, Logistic Regression achieved the highest accuracy of 98.45%, demonstrating its strong capability in identifying fake news. Through metrics such as precision, recall, F1-score, and confusion matrices, we were able to thoroughly assess the effectiveness of the models.

Future Scope:

- **Deep Learning Models:** Implement advanced models like LSTM, BERT, or Transformer-based architectures to better understand context and semantics.
- **Feature Enrichment:** Incorporate metadata such as publication date, source credibility, and user comments to improve model reliability.
- **Multilingual Detection:** Extend the model to detect fake news in multiple languages using language-specific preprocessing and embeddings.
- **Real-time Deployment:** Build a web-based or mobile application that can classify incoming news articles in real-time.
- **Model Interpretability:** Integrate explainable AI (XAI) techniques to make the predictions more transparent and trustworthy.

8. References

- Dataset Source: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-
- Scikit-learn Documentation: https://scikit-learn.org/stable/documentation.html
- Text Mining with TF-IDF: Manning, C.D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- Research Paper: Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H. (2017). *Fake News Detection on Social Media: A Data Mining Perspective*. ACM SIGKDD Explorations.