📄 Final Report: News Article Classification (Fake or Real)

1. Introduction

Fake news has become a major concern in the age of digital information. With social media allowing the rapid spread of misinformation, it is crucial to have automated tools that can help detect and classify fake content. In this project, we build a machine learning-based fake news classifier using natural language processing techniques to distinguish between real and fake news articles.

2. Abstract

This project uses the Fake and Real News Dataset from Kaggle. The text from news articles is vectorized using TF-IDF, and a Passive Aggressive Classifier is trained for classification. The model achieved an accuracy of **99.45%**. A simple **Streamlit** web app is also developed, allowing users to input news text and receive real-time predictions.

3. Tools and Technologies

- **Python** (Jupyter Notebook)
- **Libraries**: Pandas, Scikit-learn, Joblib, Streamlit
- **Model**: PassiveAggressiveClassifier
- **Data**: Fake.csv and True.csv (Kaggle)

4. Methodology

- **Data Preprocessing**: Labeled fake and real articles, shuffled and split into train-test sets.
- **Vectorization**: Used TF-IDF to convert text into numerical features.
- **Model Training**: Trained a PassiveAggressiveClassifier on the processed data.

- **Evaluation**: Achieved 99.45% accuracy. Confusion matrix showed low false positive/negative rates.
- **Deployment**: Built a user-friendly Streamlit interface to test the model live.

5. Results

The model performed very well on the test data. Sample testing in the Streamlit app gave fast and accurate results for many cases. Some real articles may still be misclassified due to dataset bias (e.g., Indian news not seen in training data).

6. Conclusion

The project demonstrates how simple NLP techniques combined with machine learning can be effective for fake news detection. With a small dataset and efficient algorithm, the model achieved high accuracy. The added Streamlit interface makes it easy to test and demonstrate in real-time.

7. Future Scope

- Use a more diverse dataset with recent news
- Add advanced models (e.g., BERT)
- Improve the interface with options like file upload
- Deploy the app to Streamlit Cloud