**Twitter Fake News Detection Using Machine Learning**

**1. Abstract**

The widespread dissemination of fake news on social media platforms like Twitter has become a major concern, influencing public opinion and spreading misinformation rapidly. This project aims to build a machine learning model utilizing logistic regression to detect fake news on Twitter. A dataset sourced from Kaggle, spanning news articles published in India between 2021 and 2022, is used for training and evaluation. Preprocessing techniques, including text cleaning and TF-IDF vectorization, were employed to transform the news content into numerical data for the logistic regression model. After addressing class imbalance with the SMOTE technique, the model achieved an accuracy of 54%, demonstrating a basic ability to classify news as "True" or "False." Performance metrics such as accuracy, precision, recall, and F1-score were computed to assess the model's effectiveness.

**2. Introduction**

The proliferation of fake news poses a significant threat to societies worldwide, undermining the credibility of legitimate information and spreading misinformation. Twitter, a widely used social media platform, is a hotbed for the distribution of such news, where information spreads rapidly, often without verification. Machine learning techniques provide a scalable solution to this problem by automatically identifying fake news based on patterns and features derived from the text.

This project focuses on applying machine learning, specifically logistic regression, to detect fake news in Indian Twitter data. Using a dataset containing news articles from 2017 to 2022, the model is designed to classify news articles as "True" or "False." This report details the methodology, including data preprocessing, feature extraction, model training, evaluation, and results.

**3. Previous Work**

Several approaches have been developed to combat the issue of fake news detection. Traditional methods like manual fact-checking are accurate but not scalable for large datasets. Keyword detection and sentiment analysis provide insights but often fail to capture the nuanced patterns within fake news. Machine learning algorithms, such as Naive Bayes, Decision Trees, and Support Vector Machines (SVM), have been extensively researched for this purpose.

Previous studies demonstrate that while SVMs and Decision Trees perform well on balanced datasets, they may struggle with real-world imbalanced datasets, where fake news is often less frequent than true news. Logistic regression, though a simpler model, offers interpretability and works well for binary classification tasks, making it a suitable choice for this project. In addition, handling the class imbalance using SMOTE ensures a more balanced and representative model during training.

**4. Problem Statement & Objective**

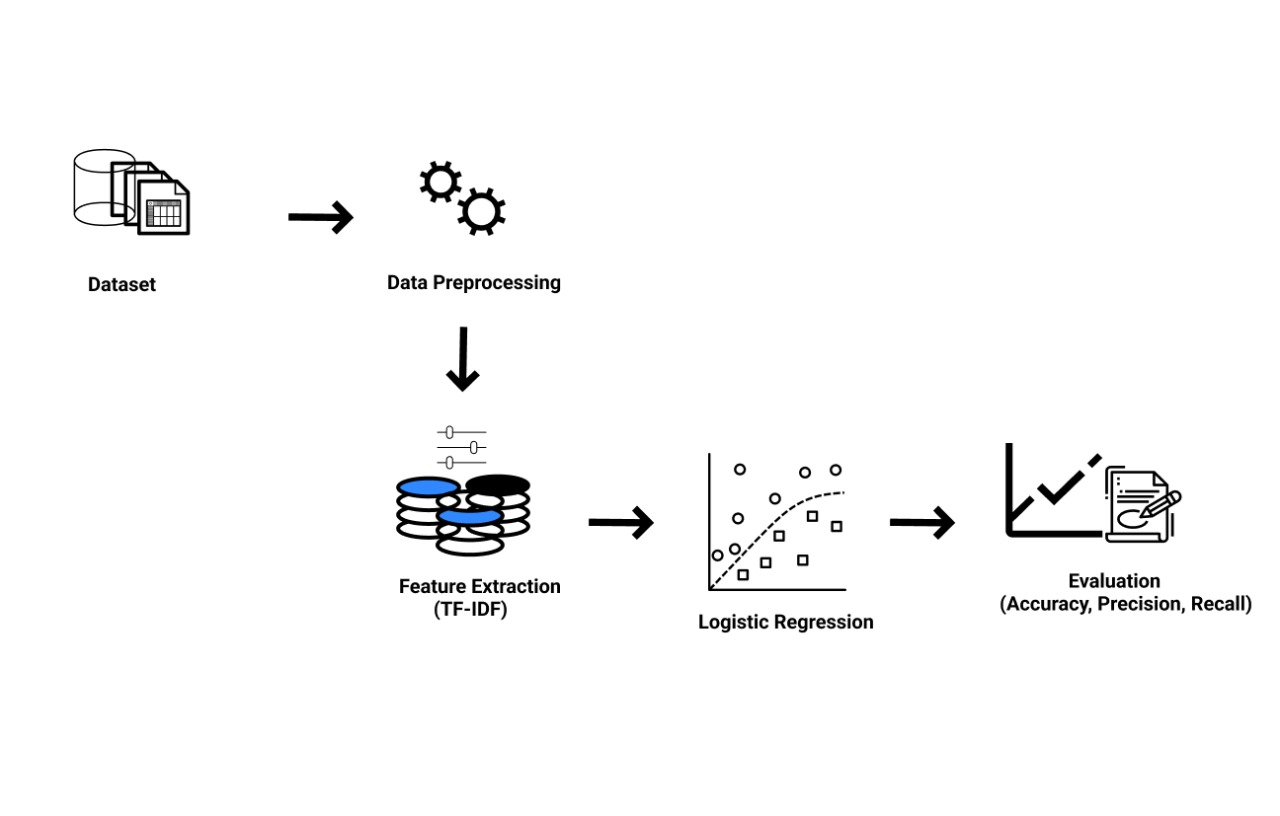
The objective of this project is to develop a machine learning model capable of detecting fake news on Twitter, focusing specifically on Indian news articles. By leveraging logistic regression and addressing the class imbalance using SMOTE, the project seeks to classify news articles as either "True" or "False." The problem is rooted in the growing spread of misinformation on social media platforms and the need for an automated solution to mitigate its effects.

**5. System Model**

The system follows a structured pipeline, starting from data collection, preprocessing, and feature extraction to model training, evaluation, and prediction. Below is the breakdown of the system architecture:

1. **Data Collection**: The dataset was sourced from Kaggle, containing Indian news articles with labelled classes indicating whether the news is true or false.
2. **Data Preprocessing**: The raw text is preprocessed by removing stopwords, digits, and special characters. Tokenization and lemmatization are performed to standardize the text data.
3. **Feature Extraction**: TF-IDF (Term Frequency-Inverse Document Frequency) is used to convert the preprocessed text into numerical vectors, which serve as inputs for the machine learning model.
4. **Model Training**: Logistic regression is used to train the model, which is ideal for binary classification problems. The class imbalance is addressed using the SMOTE technique to ensure that the model does not favour the majority class.
5. **Evaluation**: The model is evaluated using various metrics, including accuracy, precision, recall, and F1-score. A confusion matrix is also computed to visualize the classification results.

The diagram below illustrates the system flow :

**6.System Requirements**

To successfully implement and run this project, we will need **Python 3.10** or a higher version. There are two ways we can execute the project:

**Option 1: Online (Google Colab)**

We can run the project online using [Google Colab](https://colab.research.google.com/), a cloud-based platform that doesn't require any local installation.

**Option 2: Local Setup (on our system)**

Alternatively, we can run the project locally using an IDE such as PyCharm, VS Code, or Jupyter Notebook. For this, we need to ensure that our system meets the following **minimum hardware requirements**:

* **Processor**: Intel Core i5 or higher.
* **Memory**: 8 GB RAM or more (depending on the dataset size).
* **Storage**: At least 2 GB of free space for project files and dependencies.

**Required Python Libraries:**

The following Python libraries are necessary for the successful execution of the project:

* **Pandas**: For data manipulation and analysis.
* **NumPy**: For numerical operations and efficient handling of arrays.
* **scikit-learn**: To implement machine learning algorithms, train the model, and evaluate its performance.
* **nltk (Natural Language Toolkit)**: For natural language processing tasks, such as text preprocessing, tokenization, and cleaning.
* **imblearn (imbalanced-learn)**: To manage imbalanced datasets using the SMOTE (Synthetic Minority Over-sampling Technique) algorithm.
* **joblib**: For saving and loading the machine learning model, allowing us to avoid retraining the model each time the code is executed.

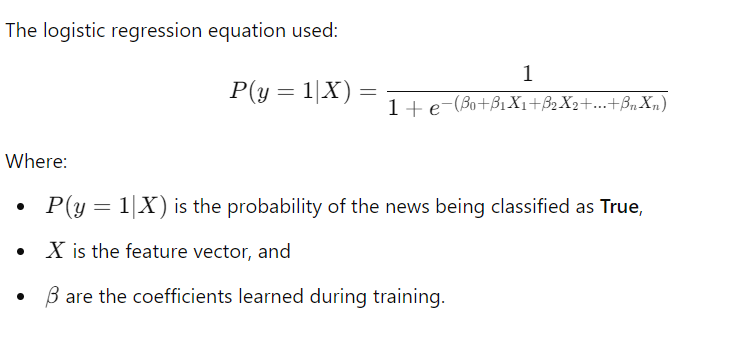
We will install these libraries in our Python environment using pip install, and verify that our Python version is up-to-date to ensure compatibility.

**7. Algorithm**

The project uses **Logistic Regression**, a widely known statistical model that predicts the probability of a binary outcome. In this case, the two possible outcomes are whether the news article is classified as **True** or **False**.

Steps:

1. **Preprocessing**: Clean and tokenize the text, remove stopwords, special characters, and digits.
2. **Feature Extraction using TF-IDF**: Text is transformed into numerical vectors by applying the TF-IDF technique.
3. **SMOTE**: Apply Synthetic Minority Over-sampling Technique to balance the dataset, ensuring the model has sufficient data for both classes.
4. **Model Training**: Logistic regression is applied, optimizing for binary classification.
5. **Evaluation**: Accuracy, confusion matrix, and classification report (precision, recall, F1-score) are computed to evaluate the model's performance.

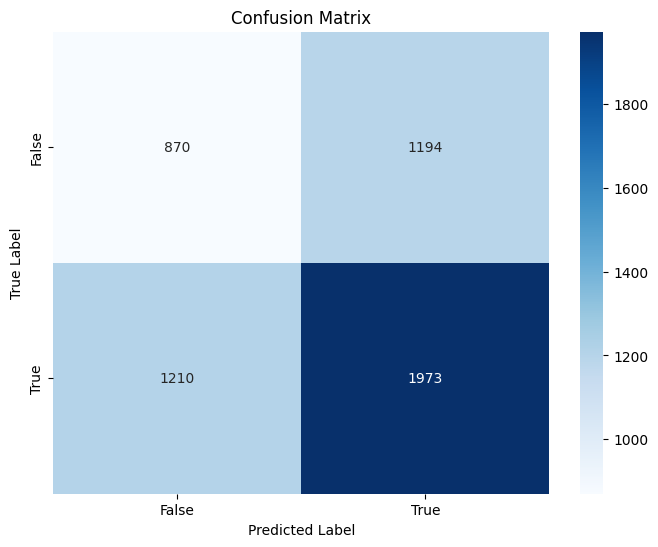


**8. Results**

The model achieved an accuracy of 54.18%, with the following results from the evaluation metrics:

**Accuracy**: The model’s overall accuracy is **54.18%**, which is the proportion of correct predictions over the total number of predictions.

**Confusion Matrix**: A table that helps visualize the performance:



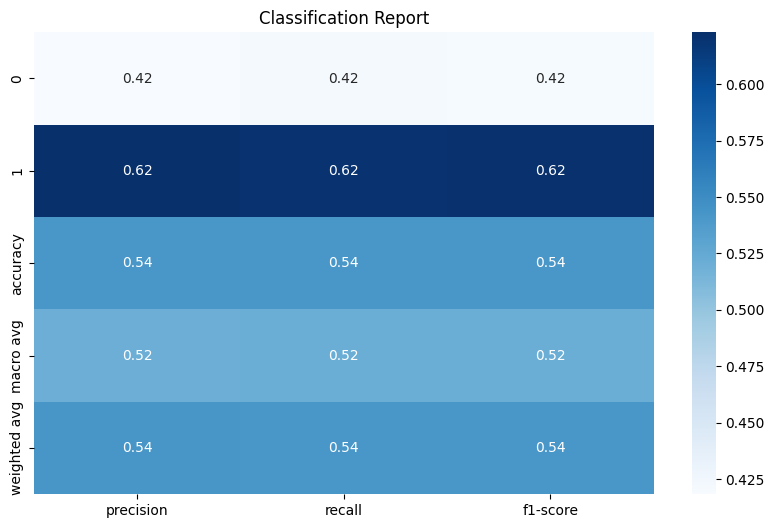
* True Positive (TP): Correctly predicted fake news as fake news
* True Negative (TN): Correctly predicted real news as real.
* False Positive (FP): Incorrectly predicted real news as fake (also called a Type I error).
* False Negative (FN): Incorrectly predicted fake news as real (also called a Type II error).

Confusion Matrix:

* True Positive: 1973
* True Negative: 870
* False Positive: 1194
* False Negative: 1210

**Classification Report**: This includes:

* Precision: The proportion of true positives out of all predicted positives.
* Recall: The proportion of true positives out of all actual positives.
* F1-Score: The harmonic mean of precision and recall, providing a balanced measure.



Classification Report:

* Precision (True): 0.62
* Recall (True): 0.62
* F1-Score (True): 0.62
* Precision (False): 0.42
* Recall (False): 0.42
* F1-Score (False): 0.42

**9. Conclusion**

This project explored the use of machine learning, particularly logistic regression, for detecting fake news . The dataset comprised Indian news articles, and after applying SMOTE and TF-IDF, the logistic regression model was able to classify news articles with an accuracy of 54%. While this accuracy is modest, the project demonstrates the feasibility of using machine learning for fake news detection.

**Future Work:**

Model Improvement: Explore more complex algorithms like Random Forest, Neural Networks, or deep learning models to improve accuracy.

Larger Dataset: Collect more comprehensive datasets to help the model generalize better.

Real-time Detection: Implement the model in real-time scenarios where fake news detection is critical.