Fake News Detection

Project Overview

Build and evaluate supervised machine learning models to classify news articles as FAKE (0) or REAL (1) based on textual content, helping reduce the spread of misinformation.

- **Learning Type:** Supervised Learning
- **Task:** Binary Text Classification
- **Target Variable:** `class`
  - `0` → Fake News
  - `1` → Real News

Dataset Description

Datasets Used

- **Fake.csv** → Fake news articles
- **True.csv** → Real news articles
- Kaggle / Online News Repositories

Dataset Shape

Dataset	Rows	Columns
Fake News	23,481	5
Real News	21,417	5
Total	~44,898	5

Features

Feature	Description
title	News headline
text	Full article content
subject	Topic/category (optional)
date	Publication date (optional)
class	Target label (0 = Fake, 1 = Real)

Project Workflow

Step 1: Import Required Libraries

Load essential libraries for:

- Data manipulation
- Text preprocessing
- Feature extraction
- Model training & evaluation

Step 2: Load the Datasets

- Load `Fake.csv` and `True.csv`

Step 3: Assign Class Labels

```
Fake News → class = 0
Real News → class = 1
```

This converts the problem into a binary classification task.

Step 4: Dataset Validation

- Check number of rows and columns
- Verify class balance
- Ensure no label leakage

Step 5: Manual Testing Data

Before merging:

- Create **manual testing samples**
- Assign class labels
- Remove these samples from training data

Step 6: Merge Datasets

Combine fake and real datasets into a single DataFrame:

```
data_merge = Fake + Real
```

Shuffle the data to remove ordering bias.

Step 7: Drop Unnecessary Columns

Remove:

- `subject`
- `date`

Step 8: Text Cleaning & Preprocessing

Create a reusable text-cleaning function:

- Lowercase text
- Remove URLs
- Remove punctuation & special characters
- Remove stopwords
- Apply stemming or lemmatization

Step 9: Feature & Target Separation

- **X:** Cleaned text (title + article body)
- **y:** Class label

Step 10: Train–Test Split

- Split data into:
  - **75% Training**
  - **25% Testing**

Step 11: Text Vectorization (TF-IDF)

Convert raw text into numerical features using TF-IDF:

- Learns important words
- Reduces impact of common, meaningless terms
- Limits vocabulary to top 5,000–10,000 features

This is the core NLP transformation step.

Model Development

Train and compare **four classifiers**:

Model 1: Logistic Regression

- Strong baseline for text classification
- Fast and interpretable

Model 2: Decision Tree

- Captures non-linear patterns
- High risk of overfitting

Model 3: Gradient Boosting Classifier

- Combines weak learners
- Better generalization than single trees

Model 4: Random Forest

- Ensemble of decision trees
- Reduces variance

Model Evaluation

Each model is evaluated using:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix
- Classification Report

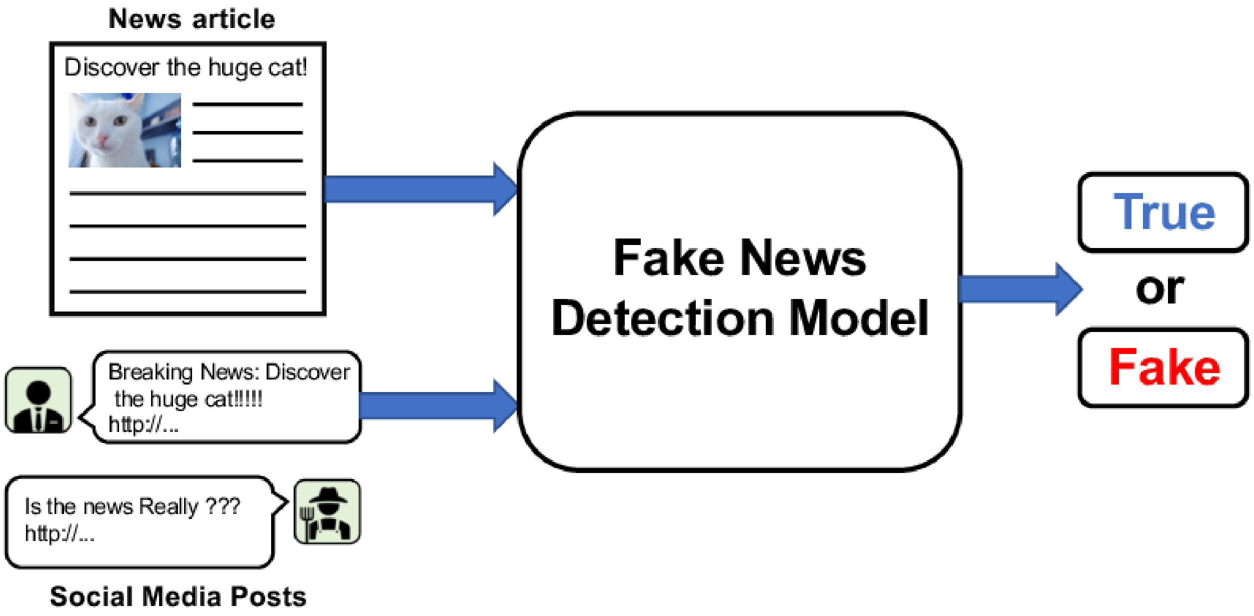
Evaluation Rule:

Do NOT trust training accuracy.
Test performance + F1-score matter most.

Fake News Prediction (Real-World Testing)

- Accept raw news text as input
- Apply the same preprocessing & TF-IDF pipeline
- Output prediction:
  - `FAKE NEWS`
  - `REAL NEWS`

This step proves your model actually works outside the dataset.


News article
Discover the huge cat!
Breaking News: Discover the huge cat!!!!! http://...
Is the news Really ??? http://...
Social Media Posts
Fake News Detection Model
True
or
Fake