📰 Fake Disaster News Detection - NLP + ML Project

Project Summary:

This project classifies whether a news tweet is **real disaster news** or **fake/irrelevant** using NLP and Machine Learning.
It uses a **Logistic Regression** model trained on **TF-IDF features** and deployed via a **Flask web app**.

Technologies Used:

Category	Tool / Library
Programming	Python 3.10+
NLP	spaCy, Gensim (Word2Vec), `re`
ML Models	Logistic Regression (Best), Naive Bayes, SVM
Feature Vectors	TF-IDF (🏆 Best), Word2Vec
Visualization	Matplotlib, Seaborn, WordCloud
Deployment	Flask
Model Saving	Pickle

Project Structure:

```
project/
├── train.csv                Training dataset
├── test.csv                 Test dataset
├── my_submission.csv        Submission file
│
├── fakenewsdet.ipynb        Complete analysis and training notebook
│
├── model/
│   ├── fake_news_model.pkl  Saved best model (LogReg + TF-IDF)
```

```
|   └── tfidf_vectorizer.pkl  Fitted TF-IDF vectorizer
|
├── app.py                    Flask backend
├── templates/
|   └── index.html            Web UI
├── static/
|   └── style.css             Web styling
```

Overview:

1. 🧹 Preprocessing

- Used **spaCy** to:
    - Lowercase
    - Remove punctuations, digits, URLs
    - Lemmatize
    - Remove stopwords
    - Tokenize

2. 📊 EDA

- Visualized class distribution
- WordClouds for both real and fake tweets

3. ✨ Feature Engineering

- Used **TF-IDF Vectorizer** (5000 features)
- Also tested **Word2Vec** with average embedding (100-dim)

4. 🧠 Model Training & Evaluation

- Trained multiple models:
    - Logistic Regression (🏆 Best)
    - Multinomial Naive Bayes
    - Linear SVM
- Evaluated using:
    - Accuracy, F1-score
    - Confusion Matrix
    - ROC Curve, PR Curve
    - Misclassified tweet examples

5. 🔍 Hyperparameter Tuning

- Performed **GridSearchCV** on:
    - Logistic Regression (C, solver, max_iter)
    - Naive Bayes (alpha, fit_prior)
    - SVM (C, max_iter)

✅ Final model: **Logistic Regression with TF-IDF**
📈 Accuracy: **~79.4%**
📊 Best F1-Score after tuning

Saving Artifacts

```
pickle.dump(best_lr_tfidf, open("model/fake_news_model.pkl", "wb"))
pickle.dump(tfidf_vectorizer, open("model/tfidf_vectorizer.pkl",
"wb"))
```

Web App (Flask)

- Accepts tweet text input
- Preprocesses and vectorizes using TF-IDF
- Predicts label using Logistic Regression
- Displays result with styling
- Keeps session history

Run using:

```
python app.py
```

Example Test Cases

✅ Real Disaster

```
Fire breaks out in a Mumbai hospital, dozens rescued
Flash floods hit Uttarakhand, emergency declared
```

❌ Fake / Irrelevant

```
This Monday feels like a train wreck 🚂
My WiFi is down again. Total disaster!
```

Final Takeaways

- **TF-IDF > Word2Vec** on small datasets
- **Logistic Regression** outperformed other models after tuning
- Can be improved with:

- More labeled data
- Advanced embeddings (e.g., BERT)
- Better handling of figurative language