**DevelopersHub Corporation: AI/ML Internee**

**Name: Namala Aftab DHC-888**

**Task 1: Fake News Detection Project Report**

**1. Approach**

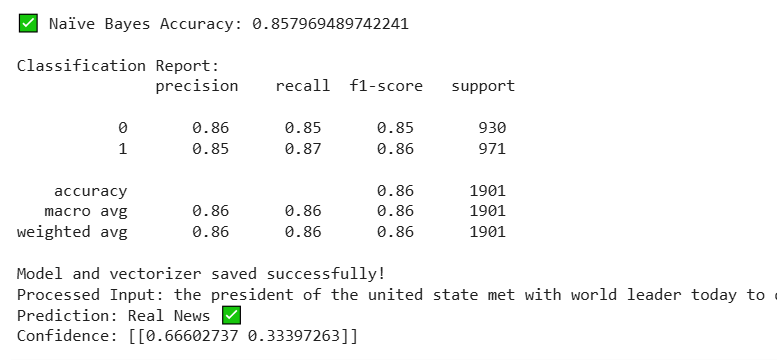
* **Data Loading & Cleaning**:  
  The dataset was loaded from a CSV file (news.csv). Text data was cleaned using preprocessing steps
* **Text Vectorization**:  
  The cleaned text was transformed **TF-IDF Vectorizer**.
* A **Multinomial Naïve Bayes** classifier was chosen. The data was split 70-30 into training and test sets.
* **Prediction**:  
  A prediction function was implemented to "Real News " or "Fake News .
* **Model Saving**:  
  The trained model and vectorizer were saved as .pkl files (nb\_model.pkl and vectorizer.pkl) using pickle.
* **Frontend Deployment**:  
  A basic HTML form and a Flask-based backend (app.py) were created to allow users to test the model via a local web app.

**2. Challenges Faced**.

* **Deployment**: Initially, there was confusion about submitting .pkl files, HTML files, and integrating with Flask.

**3. Model Performance**

* **Accuracy**: 85.8% on the test set.

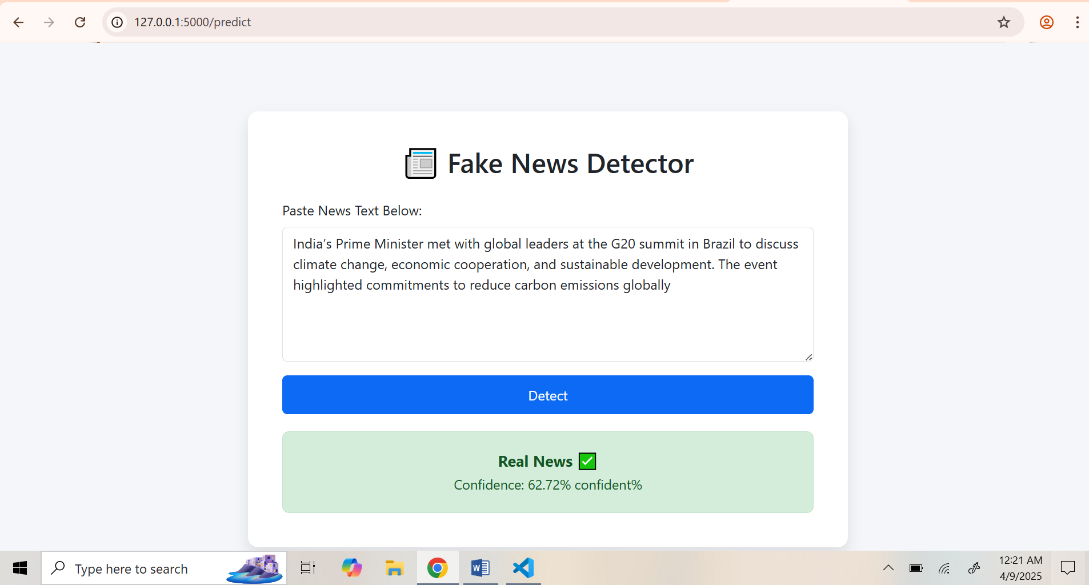


**Deployment:**

* The model is **locally deployed** using **Flask**.
* It accepts news text via a form and returns the classification and confidence score.

**Instructions to Run:**

* Make sure app.py, nb\_model.pkl, vectorizer.pkl, and templates/index.html are in the same project folder.
* Install requirements:
  + pip install flask nltk scikit-learn pandas
* Run the Flask app:
  + python app.py
* Visit http://127.0.0.1:5000 in your browser to test the app.

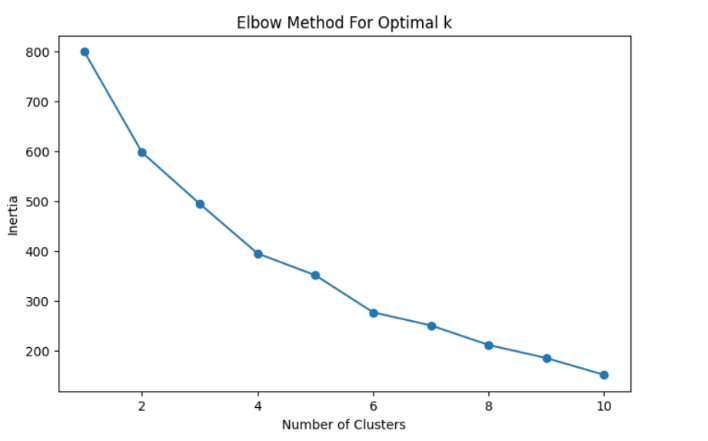


**Task 2: Customer Segmentation Project Report**

**1. Approach**

**1. Data Exploration & Preprocessing**

* Loaded the Mall Customer dataset containing Age, Gender, Annual Income, and Spending Score.
* Encoded the Gender column using Label Encoding.
* Dropped CustomerID as it wasn’t useful for clustering.
* Scaled the features using StandardScaler to normalize values before clustering.

**2. Clustering Technique**

* Applied the **K-Means clustering algorithm** to segment customers.
* Used the **Elbow Method** to find the optimal number of clusters (k = 5).
* Trained the model and assigned a cluster label to each customer.

**3. Visualization & Interpretation**

* Plotted Annual Income vs. Spending Score to visualize customer clusters.
* Analyzed each segment’s behavior based on income and spending patterns.

## **2. Challenges Faced**

* **Interpretation:** Understanding what each cluster represented required comparing multiple features and validating assumptions.

## **3. Model Performance & Improvements**

* The clustering model successfully grouped customers into 5 distinct behavior-based segments:
  + Low income, high spending
  + High income, low spending
  + Average earners/spenders
  + Low income, low spending
  + High income, high spending (VIPs)
* **Improvement Opportunities:**
  + Include more features like product preferences or visit frequency.

**Task 3: IMDB Movie Revie Sentiment Analysis Project Report**

1. **Approach Used**

* **Preprocessing**:
  + Text data was cleaned using preprocessing steps
* **Feature Extraction**:
  + Applied **TF-IDF Vectorization** with a max of 5000 features to transform text into numerical form.
* **Model Training**:
  + Used **Logistic Regression** due to its simplicity and effectiveness for binary classification.
  + Split data into **80% training** and **20% testing** using train\_test\_split.
* **Evaluation**:
  + Assessed model performance using **accuracy** and **F1-score**

1. **Challenges Faced**  
   When loading the saved model and vectorizer in Flask, I ran into version mismatch issues and file path confusion. Took me a while to realize they just needed to be in the same folder as app.py.
2. **Model Performance & Improveents**

* **Accuracy**: ~89%
* **F1-Score**: ~0.89
* For a basic model, performance was strong.
* Potential improvements:
  + Use advanced models (e.g., BERT, LSTM)

1. **Deployment Instructions**

* A simple **Flask app** was built for user interaction:
  + Open app.py in a terminal.
  + Run using: python app.py
  + Visit http://127.0.0.1:5000 in your browser.
  + Enter a review and get sentiment instantly.

