**Task-5**

**Introduction**This Python-based Fake News Detection System aims to classify news articles as either "Real" or "Fake" using machine learning techniques. It leverages the Naive Bayes classifier with a Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction approach. The system incorporates a graphical user interface (GUI) built with tkinter that allows users to input news articles and receive classification results. The application also includes data preprocessing steps to clean and prepare text data for training and evaluation.**Key Libraries and Dependencies**

* **Pandas (pd):** Used for data manipulation and loading CSV files.
* **Scikit-learn (sklearn):** For text vectorization (TF-IDF), model training (Naive Bayes), and performance evaluation.
* **Natural Language Toolkit (NLTK):** For text preprocessing (stopwords, lemmatization).
* **Langdetect:** Used to detect the language of each article, ensuring only English articles are considered.
* **Tkinter:** Provides the GUI for the fake news detection interface.
* **Matplotlib and Seaborn:** For plotting the confusion matrix to visualize model performance.

**Key Features**

* **Data Preprocessing:** The system preprocesses the news articles by removing HTML tags, punctuation, and numbers, converting the text to lowercase, removing stopwords, and lemmatizing the words.
* **Language Detection:** The system detects the language of the articles and keeps only English articles for training and classification, ensuring the model deals with a uniform language dataset.
* **TF-IDF Vectorization:** The text data is transformed into numerical features using TF-IDF to capture the importance of words within the news articles, which serves as input to the machine learning model.
* **Naive Bayes Classification:** A Multinomial Naive Bayes classifier is trained on the TF-IDF features to distinguish between fake and real news articles. The classifier is evaluated based on metrics such as accuracy, precision, recall, and F1-score.
* **GUI for User Interaction:** The system has a user-friendly GUI, allowing users to input a news article and receive a classification (Real or Fake) along with confidence in the prediction.

**Code Structure**

* **NLTK Data Setup:** The function download\_nltk\_data ensures that the necessary NLTK datasets (stopwords and WordNet lemmatizer data) are available before the model starts training.
* **Data Preprocessing:** The load\_and\_preprocess\_data method loads the dataset, removes non-English articles, and cleans the text using a custom function clean\_text. The cleaning process includes:
* Removing HTML tags, punctuation, and numbers.
* Converting to lowercase
* Removing stopwords.
* Lemmatizing words.
* **Model Training**The train\_model method:
* Uses TfidfVectorizer to convert the cleaned text into numerical features.
* Splits the data into training and test sets using train\_test\_split.
* Trains a Multinomial Naive Bayes classifier and evaluates its performance using accuracy, classification report, and confusion matrix.
* **Evaluation and Confusion Matrix**The confusion matrix visually represents the performance of the model by plotting actual versus predicted results.
* **Text Classification**The classify\_text method allows the system to classify new text inputs. It cleans the input text, vectorizes it using the trained TF-IDF model, and then predicts whether the text is real or fake news.
* **GUI Setup**The application has a simple GUI with a text input box for users to paste news articles and a button to trigger the detection process. The result is displayed with the prediction and confidence level.

**Application Workflow**

* **Data Preprocessing:** The system loads the CSV file, cleans the data, detects language, and preprocesses the text for training.
* **Model Training:** The system uses TF-IDF vectorization and trains a Naive Bayes classifier to predict fake news.
* **User Interaction:** Through the GUI, users can input text, and the system classifies it as either real or fake news, showing the confidence level of the prediction.
* **Evaluation:** The system provides classification metrics like accuracy, confusion matrix, and a detailed classification report.

**Strengths**

* **Comprehensive Preprocessing:** The system effectively cleans the text data, making it ready for model training.
* **Accurate Classification:** By using the Naive Bayes algorithm, the system achieves good performance in distinguishing between fake and real news.
* **User-Friendly Interface:** The GUI makes it easy for non-technical users to interact with the system.
* **Clear Evaluation:** The system outputs classification reports and visualizes the confusion matrix, providing insight into model performance.

**Conclusion**The Fake News Detection System provides an effective solution for classifying news articles as real or fake using machine learning and natural language processing techniques. The use of a GUI enhances user interaction, making the system accessible to non-technical users. With further improvements, this system could be deployed in real-world scenarios to help combat the spread of misinformation.