# Real-Time Fake News Detector Machine

**Group members:**
**Shadman Ameen – 0759520**
**Sadman Sakib Mridul – 0773056**

## Power of TF-IDF Vectorization in Text Data Analysis with Machine Learning

### Data Set Review

Fake news VS Real news

| | |
|---|---|
| **Title** | **The headline or title of the fake news article.** |
| **Text** | **The main body of the fake news article containing the deceptive or misleading content.** |
| **Subject** | **The subject or topic of the fake news article.** |
| **Date** | **The date of publication or creation of the fake news article.** |
| **Format** | **Tabular format represented as a table.** |
| **Size** | • **Fake News: 23,481 rows and 5 columns.**<br>• **Real News: 21,417 rows and 5 columns.** |

### Exploratory Data Analysis (EDA):

- ✓ Import Data from CSV: using *read_csv()*
- ✓ Creating a Separate Column for Class Assignment to differentiate between fake and real news
- ✓ Merging Data Sets: using *concat()* function
- ✓ Data Cleaning and Exploration: using the *isnull().sum()*

### Refining Text Data:

- ❑ Convert to Lowercase:
- ❑ Remove Square Brackets and Content
- ❑ Remove Non-Word Characters
- ❑ Remove URLs
- ❑ Remove HTML Tags
- ❑ Remove Punctuation
- ❑ Remove Newlines
- ❑ Remove Words Containing Digits

### Applying Machine Learning Functions to Text Data:

- ❖ Splitting Data: divide the refined data into input features (x) and target labels (y).
- ❖ Model Training: split the data into randomized training and testing sets by allocating a certain percentage (75/25)
- ❖ Model Selection and Training: logistic regression, decision tree classifiers, gradient boosting classifiers, and random forest classifiers
- ❖ Evaluation and performance metrics over model accuracy, precision, recall, F1-score and overall effectiveness

### TF-IDF Vectorization for Text Data Analysis

A statistical metric called TF-IDF is used to assess a word's significance inside a document in relation to a group of documents.

**Two factors** are used to determine the TF-IDF score

**Term Frequency (TF):** The frequency of a term (word) in a document. It represents the proportion of the word (w) that appears in document (d) to all of the words in the documents. We are able to measure a word's frequency in the document with this straightforward expression.
For instance, the TF ratio of the word "the" would be (2/6) if the sentence had six words and two instances of "the."

$$tf(w,d) = \frac{occurence\ of\ w\ in\ document\ d}{total\ number\ of\ words\ in\ document\ d}$$

**Inverse Document Frequency (IDF):** The inverse of the frequency of the term across all documents in the corpus. IDF determines a word's significance inside a corpus D. Most commonly used terms, such as "of," "we," and "are," are essentially meaningless. It is computed by dividing the number of documents that contain the word by the total number of documents in the corpus.

$$idf(w,D) = ln(\frac{total\ number\ of\ documents\ (N)\ in\ corpus\ D}{number\ of\ documents\ containing\ w})$$

**Term Frequency Inverse Document Frequency (TFIDF)**

The product of term frequency and inverse document frequency is known as TF-IDF. It gives a word that appears frequently in a text and infrequently in the corpus more weight.

**Transforming Text Data with TF-IDF:**

- **Fit-Transform Training Data:** fitting the *TfidfVectorizer* to the training data (*x_train*) to compute the IDF parameters necessary for calculating TF-IDF scores. transform the training data into a TF-IDF matrix (*xv_train*), where each row represents a document, and each column represents a unique word in the vocabulary.
- **Transform Testing Data:** transform the testing data (*x_test*) into a corresponding TF-IDF matrix (*xv_test*).

### Machine Learning classifiers

**Logistic Regression**

To calculate the likelihood that an observation will fall into a specific class.

A binary classification to restrict predictions between 0 and 1

**Decision Tree Classifier**

Hierarchical nodes that indicate choices made in response to input features

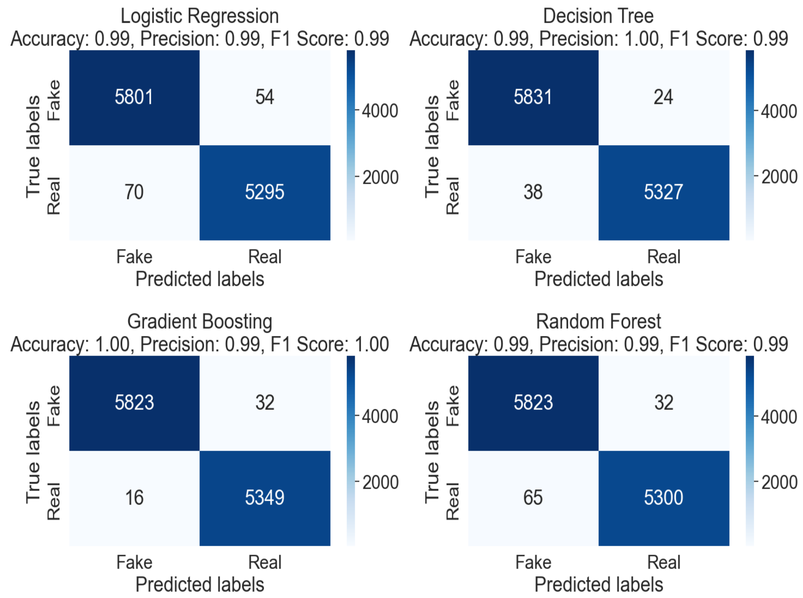
**Gradient Boosting Classifier**

Training several weak learners (usually decision trees) one after the other to fix mistakes made by the previous learners.

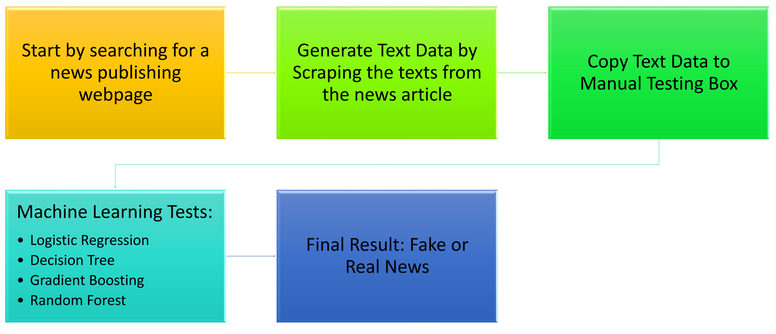
**Random Forest Classifier**

Several decision trees are trained separately on arbitrary subsets of the training set, and then the predictions are combined by voting or averaging.

### Findings and Results:




### Practical implementation




### Key Insights:

1. Deep Learning Approaches:
   - Mridha et al. (2021) emphasize the efficacy of deep learning, guiding our selection of appropriate architectures and evaluation metrics.
   - Shishah et al. (2021) propose novel BERT-based frameworks for enhanced detection leveraging linguistic features and neural-based learning.
2. Ensemble Learning and Neural Networks:
   - Aslam et al. (2021) present ensemble models, highlighting the potential of deep learning in achieving high accuracy.
3. Linguistic Feature-Based Learning:
   - Anshika et al. (2021) emphasize the incorporation of linguistic features for improved detection, resonating with our project's goals.
4. Multimodal Approach and Dual Attention Mechanisms:
   - Singh et al. (2020) advocate for multimodal analysis, aligning with our exploration of stock market data's impact on detection accuracy.
   - Shiwen Ni et al. (2021) introduce MVAN, showcasing the significance of dual attention mechanisms for real-time detection.
5. Algorithmic Evaluation and Selection:
   - Katsaros et al. (2019) and Taha et al. (2024) evaluate machine learning algorithms, offering insights into performance and methodology.