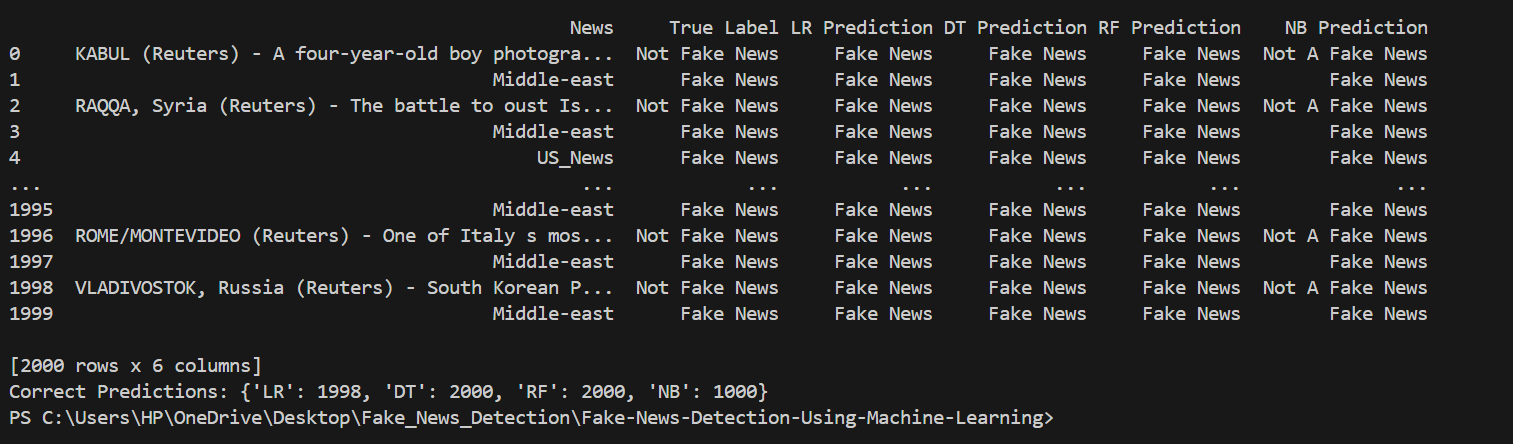
Fake News Detection Results

# Word Count Feature



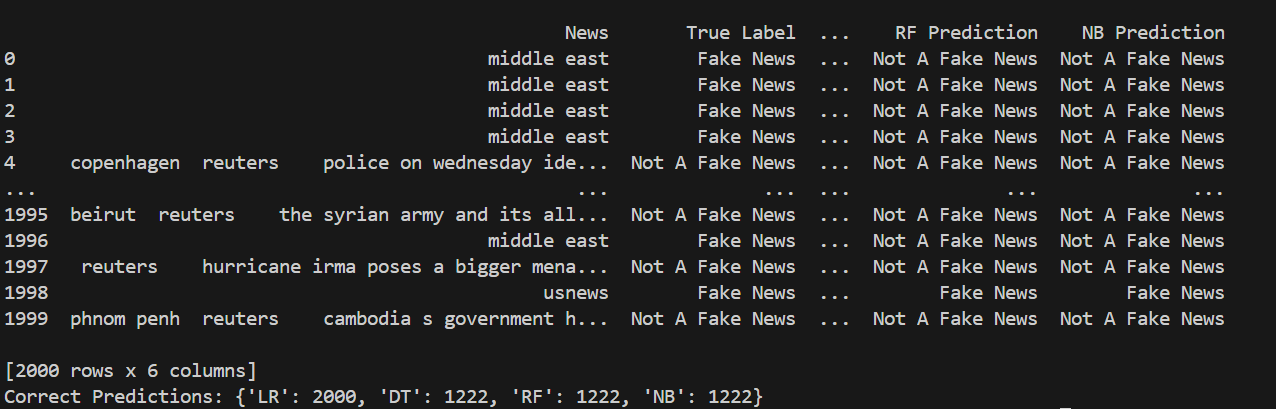
**LR = 1998/2000 Correct** , **99.9 % Accuracy**

**DT = 2000/2000 Correct , 100 % Accuracy**

**RF = 2000/2000 Correct , 100 % Accuracy**

**NB = 1000/2000 Correct , 50 % Accuracy**

# N-Gram Count Feature



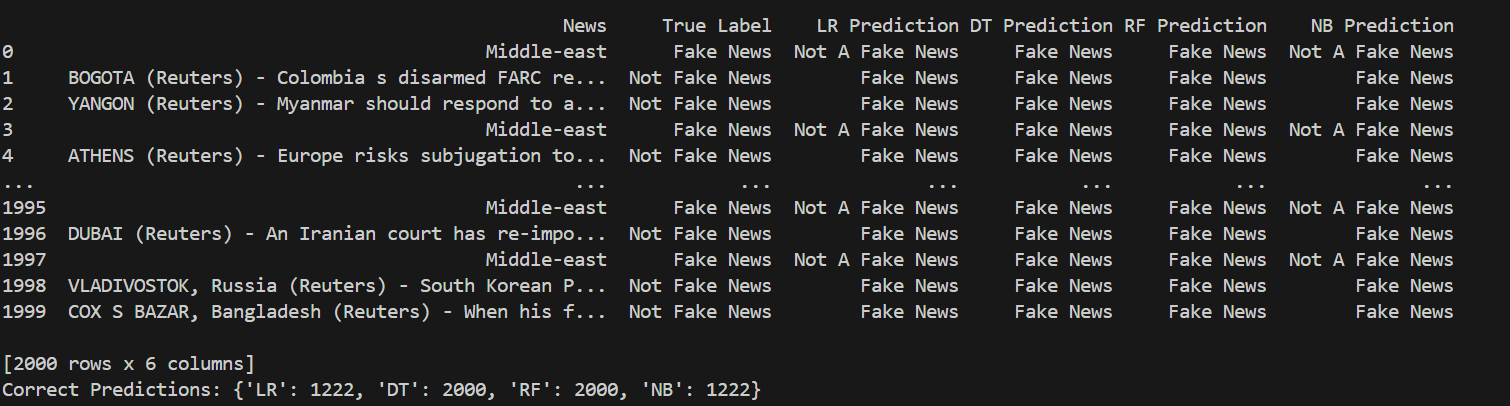
**LR = 2000/2000 Correct** , **100 % Accuracy**

**DT = 1222/2000 Correct , 61 % Accuracy**

**RF = 1222/2000 Correct , 61 % Accuracy**

**NB = 1222/2000 Correct , 61 % Accuracy**

# TF-IDF Feature



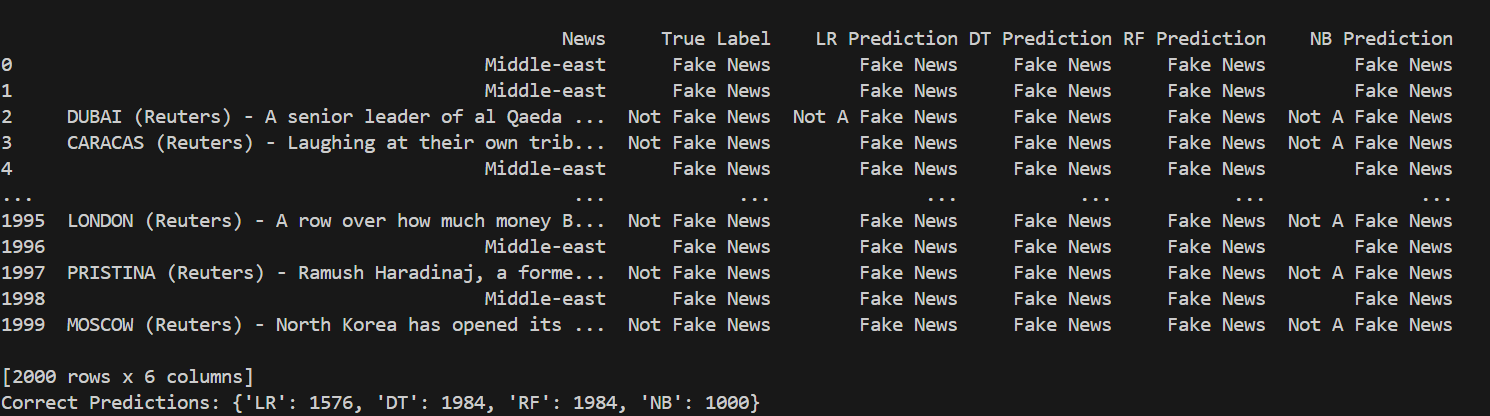
**LR = 1222/2000 Correct**, **61 % Accuracy**

**DT = 2000/2000 Correct, 100 % Accuracy**

**RF = 2000/2000 Correct, 100 % Accuracy**

**NB = 1222/2000 Correct ,61 % Accuracy**

# Sentiment Analysis Feature



**LR = 1576/2000 Correct** , **78.8 % Accuracy**

**DT = 1984/2000 Correct , 99.2 % Accuracy**

**RF = 1984/2000 Correct , 99.2% Accuracy**

**NB = 1000/2000 Correct , 50% Accuracy**

# All Features combined



**LR = 2000/2000 Correct** , **100 % Accuracy**

**DT = 2000/2000 Correct , 100 % Accuracy**

**Fake News Detection Project: Overview**

**Objective:**The main objective of the work was to compare different feature extraction methods and various classifiers for fake news detection. The motive was that among these, which feature-classifier combination would work best in terms of accuracy/performance for this particular task.  
  
**Feature Extraction Techniques**:

**1. Word Count**: It is calculated as the number of words in one news article.

**2. N-Gram Count**: It considered the sequence of 'n' words frequency bigrams, trigrams, etc.

3**. TF-IDF** - Term Frequency-Inverse Document Frequency: This feature tells how important a word is to a document in a collection.

4**. Sentiment Analysis**: It catches the overall sentiment-positive or negative-of the news article.  
  
**Classifiers Used:  
1. Logistic Regression  
2. Decision Tree  
3. Random Forest  
4. Naive Bayes**  
  
**Results Analysis:**

**The results are visualized below:**

• **Word Count Feature**: Decision Tree and Random Forest classifiers gave an accuracy of 100%, with Logistic Regression following closely behind at almost perfect accuracy. Naive Bayes did rather poorly, only getting 50% correct.

• **N-Gram Count Feature**: Logistic Regression managed full accuracy, but other classifiers, such as Decision Tree, Random Forest, and Naive Bayes all attained 61%.

**• TF-IDF Feature:** Decision Tree and Random Forest continued with perfection at 100%, while Logistic Regression and Naive Bayes were much more modest in their results, 61%.

**• Sentiment Analysis Feature**: Decision Tree and Random Forest were again among the best, this time scoring 99.2%. Logistic Regression had an accuracy rate of 78.8%, while Naive Bayes was again constant at 50%.

**Conclusion:**

**Best features**:

**Word Count and TF-IDF worked very well with the Decision Tree and Random Forest Classifiers in fake news detection, returning 100% accuracy multiple times on the two mentioned features.**

The best performance among all the classifiers for different features was that of Decision Tree and Random Forest.

Logistic Regression performed really very well with the N-Gram Count feature, but it had pretty low performance with the rest of the features.

In most of the feature sets, Naive Bayes underperformed, having achieved highest accuracy 61% achieved with N-Gram Count and TF-IDF features.

For fake news detection, the best general combination appears to be either a Decision Tree or Random Forest classifier combined with Word Count or TF-IDF features.

**Future work**

As we saw we got 100% accuracy with all combined features, so in future we can use combinations of features as a input to the model and test on more articles while also keeping the CPU, Memory Usage in mind.