**Ain Shams University**
**Faculty of Computer & Information Sciences**
**Scientific Computing Department**

# Sentiment Analysis of Movie Reviews

**Natural Language Processing (NLP)**
**T053**

| No | Name | ID |
|---|---|---|
| 1 | أحمد عبدالجواد رمضان عبدالجواد | 20191700047 |
| 2 | خالد شريف عبداللطيف عبد المجيد عزام | 20191700225 |
| 3 | سعد وليد سعد على ابوحسن | 20191700284 |
| 4 | يوسف محمد جمعه محمد | 20191700785 |
| 5 | عمر أحمد الشناوى عبدالحميد | 20191700397 |

# Data Preparation

- # Dataset

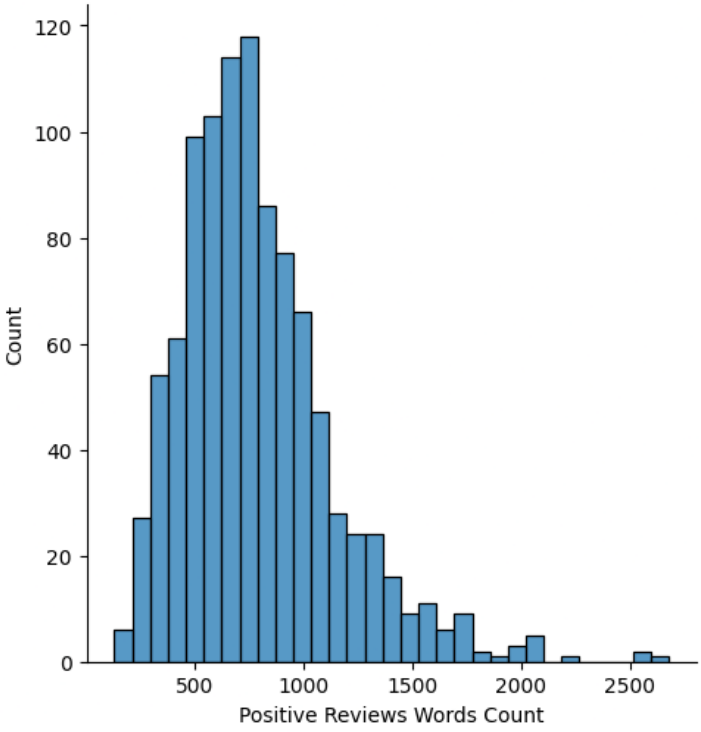
**For the movies reviews we have used the polarity dataset here is the description of the dataset:**

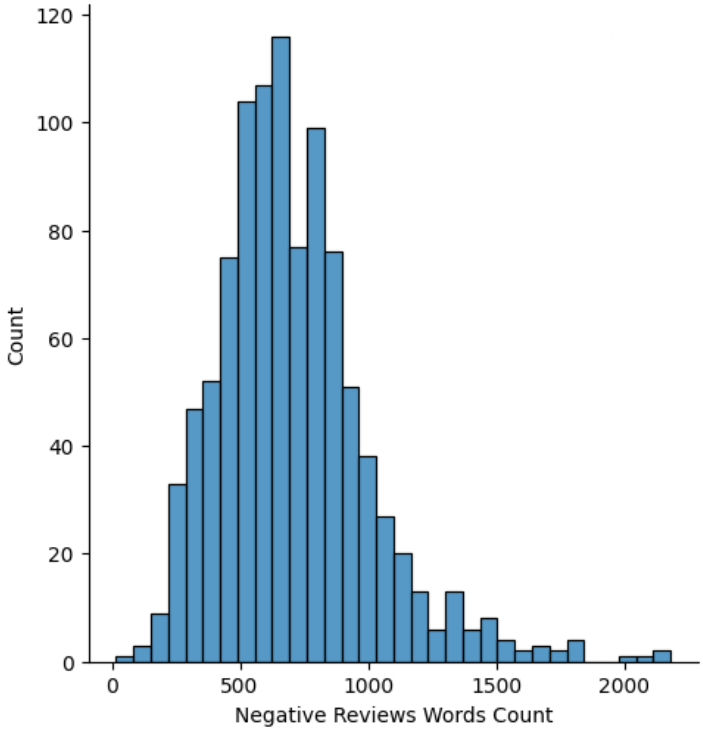
| | |
|---|---|
| **Number of classes** | **2 (positive – negative)** |
| **Number of Reviews** | **2000 (1000 positive – 1000 negative)** |

**We split the data into 70% for training and 30% for testing and here is what a random sample of a movie review from the dataset would look like:**

**“countries and legal systems that take the rule of law principle seriously , had forbidden judges and juries to make judgements in all matters that could involve them personally . luckily, movie reviewers aren't burdened with such legislation . otherwise, small pool of very special movies would be forever ignored by this reviewer . in case of star wars , 1977 science fiction epic by george lucas , the consequences would be even more severe , because that film is very special for tens of millions , if not hundreds of millions of fans . “**

**And this is the distribution of the number of words in the review in each class.**



# Preprocessing

**As we saw in the random review above there are a lot of problems in the dataset like: Non-alphanumeric values such as punctuation marks (‘ “ . , ;) and open and closing parenthesis .. etc. a lot of unimportant words “Stopwords”. So, we did the following:**

- ## Remove Non-Alphanumeric characters.
  **At first we removed all the Non-Alphanumeric characters in the dataset using the following Regex: (?![\s])\W+**

- ## Tokenization
  **Then we did Tokenization which is the process of converting a string into a sequence of tokens which could be a single word or number.**

- ## Remove Stopwords
  **Then we removed the Stopwords from the reviews, they are commonly used words that do not add much meaning to the sentence and doesn’t help in the task of classification. Like (a, an, in, and).**

- ## Lemmatization
  **The next step is Lemmatization which is the process of getting the root of the word, in other words the process of normalizing different inflected forms of the same word into a single word.**

- ## Join Tokens
  **The last step of the preprocessing is the reconstruct the movie review as a string by joining the preprocessed tokens together, to extract features from them.**

**Before we extract features from the reviews, we create a data frame for the reviews and assign a numeric label for each movie review 1 for positive review and 0 for negative review.**

# Feature Extraction

**For the feature extraction we use term frequency-inverse document frequency (TF-IDF), which is a statistical measure that is used to evaluate the importance of a word in a document in a collection of documents.**

**TF =** $\frac{Number\ of\ times\ term\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document} = (1 + log\ tf_{t,d})$

**IDF =** $log\ \frac{Total\ numbers\ of\ documents}{Number\ of\ documents\ with\ a\ given\ term\ in\ it} = log_{10}(N/df_t)$

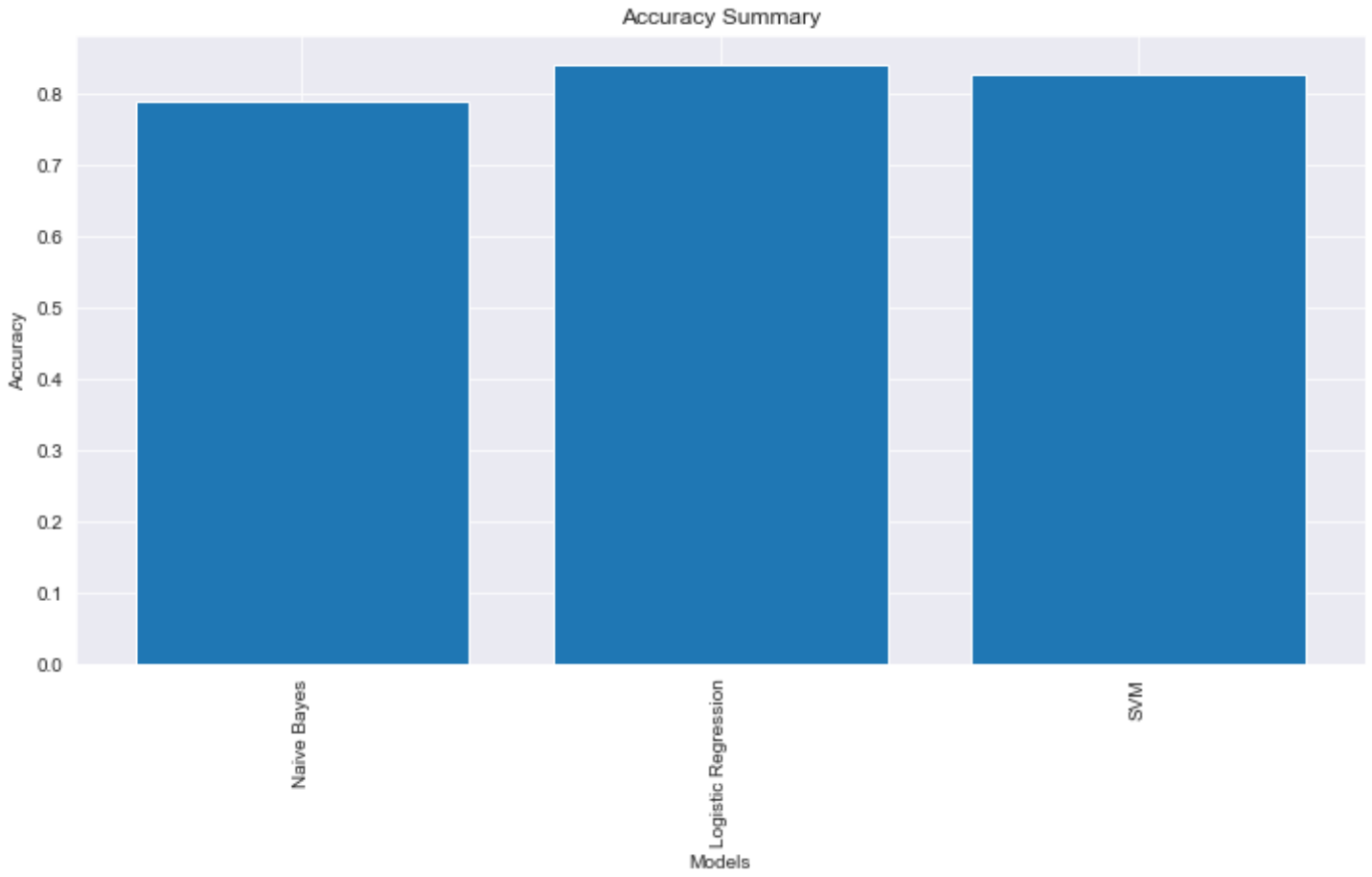
**Then,**

**The TF-IDF is calculated by multiplying TF and IDF,** $(1 + log\ tf_{t,d}) * log_{10}(N/df_t)$ **, and we use that measure as features and feed it to the classifiers.**

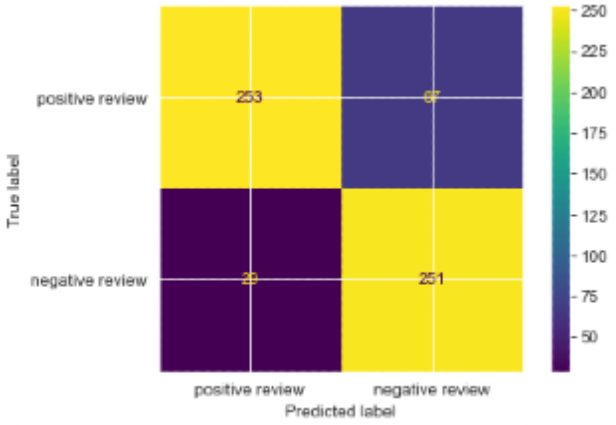
# Classification

**We have tried multiple classifiers with TF-IDF features:**

| _____ | Naïve Bayes | Logistic Regression | SVM |
|---|---|---|---|
| **TF-IDF** | **0.79** | **0.84** | **0.82** |




**And here is the detailed classification report and confusion matrix of the highest accuracy model which is logistic Regression:**




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Positive | 0.90 | 0.79 | 0.84 | 320 |
| Negative | 0.79 | 0.90 | 0.84 | 280 |
| accuracy | | | 0.84 | 600 |
| macro avg | 0.84 | 0.84 | 0.84 | 600 |
| weighted avg | 0.85 | 0.84 | 0.84 | 600 |

Logistic Regression Classifier Accuracy: 0.84