**Project Name:** Sentiment Analysis

**Github Link:** https://github.com/projectsforstudents2022/Sentiment\_Analysis.git

**Why was this project created?**

Movie reviews are an important way to gauge the performance of a movie. What provides us with a more in-depth qualitative understanding of the film's various components is a collection of movie reviews. A textual movie review identifies the film's good and poor elements, and a more in-depth examination of the review might reveal if the film, as a whole, lives up to the reviewer's expectations. Machine learning's main field of study in sentiment analysis aims to extract subjective data from textual reviews.

**What problem is it solving?**

In this project, we utilize sentiment analysis to examine a collection of movie reviews to determine the reviewers' general feelings toward the film, such as whether they enjoyed it or despised it. In order to forecast the review's general polarity, we plan to use the word relationships in the review.

**Entire explanation of project**

* **PROPOSED APPROACH**

The main goal of this research is to analyze a movie review's textual content in order to determine its underlying sentiment. We employed three techniques to extract valuable elements from the review text that might be applied to training: N-Gram modeling, TF-IDF modeling, and Bag of Words. The main objective of this research is to categorize reviews as favorable or unfavorable. We therefore investigated various classification models for this job using the feature representations mentioned above. We utilized models ranging from the straightforward Logistic Regression to the cutting-edge SVM Classifier.

We also trained the above mentioned feature representations on the Naive Bayes' Classifier, which is typically employed in text mining in conjunction with Bag of Words and N-Gram Modelling. We also trained a k-Nearest Neighbors model to categorize the reviews in accordance with how similar they were. We employed sklearn modules by adjusting their settings rather than changing how they were implemented, therefore we won't discuss their theory in this paper. Before using the above feature representations for training classifiers, we tried reducing the size of the representation set by using PCA on it. However, it did not result in a significant improvement for us because the feature vector was only decreased by 15%, thus we did not take those reductions into account.

Algorithm for creating next word prediction model :

**Step 1:** Import Libraries & Load Dataset

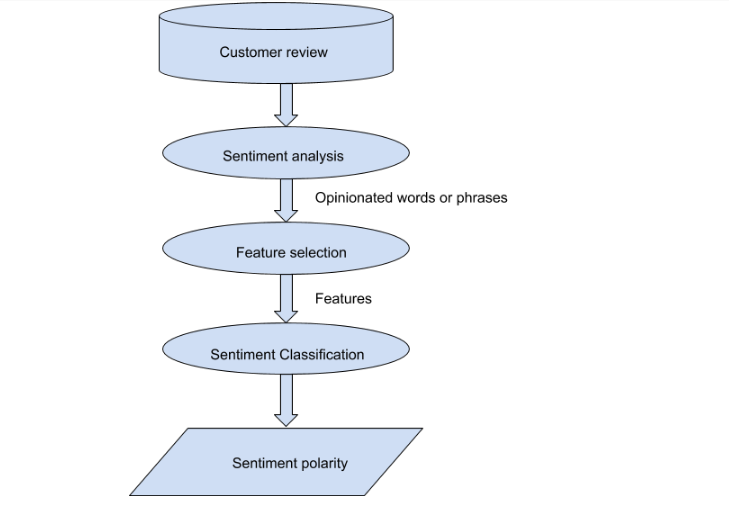
**Step 2:** Data Preprocessing

**Step 3:** Stop Words Removal

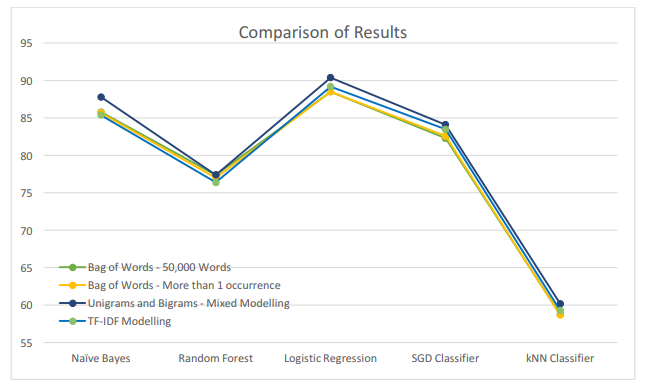
**Step 4:** Build Logistic Regression Model

**Step 5:** Train Model

**Step 6:** Testing & Visualization



* **DATA FLOW DIAGRAM**
* **RESULT**



* **CONCLUSION**

We may deduce from the findings above that the best logistic regression model for our problem statement uses a feature set that combines bigrams and a combination of ungrams. One thing to note is low accuracy with Random Forest classifiers. This may be the result of decision trees being overfit to the training set of data. Also, low accuracy of k nearest Classifiers shows us that people have varied writing styles and kNN Models are not suited to data with high variance.