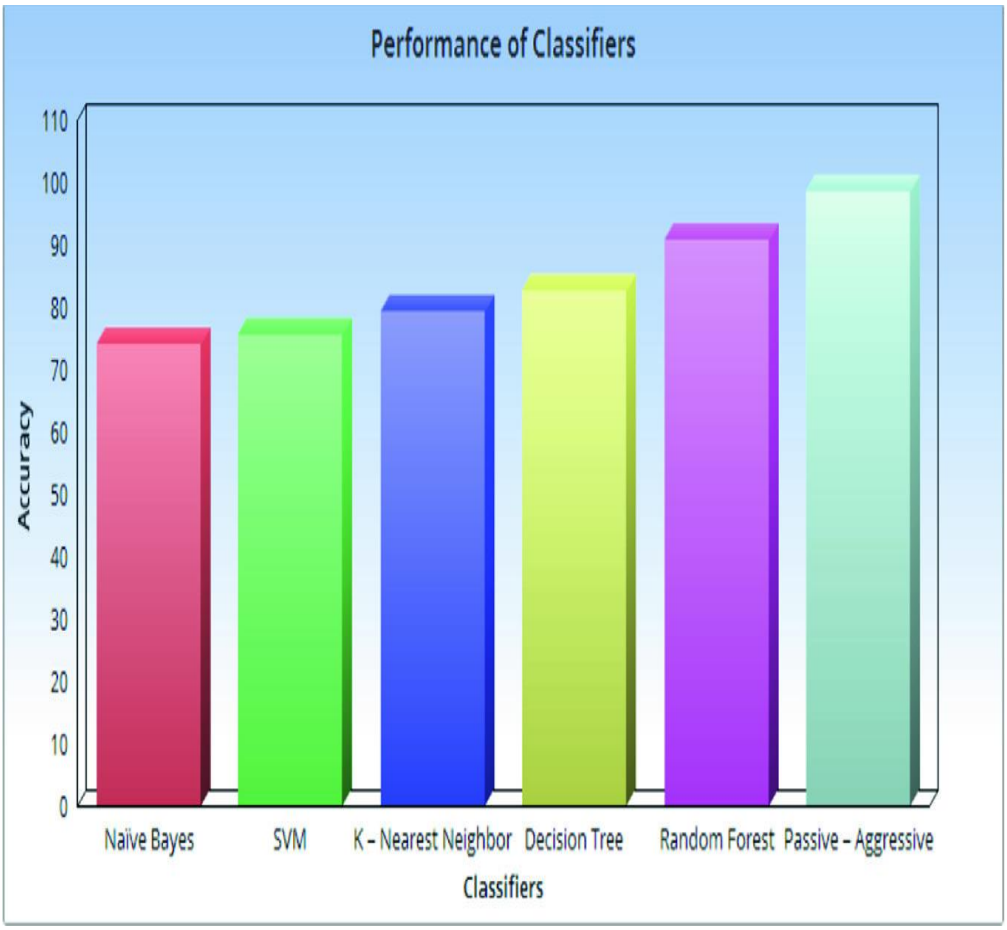
REPORT ON MODEL

Methodology used: This advanced python project of text data we are Using sklearn, to build a Tf-idf Vectorizer on our dataset. Then, we initialize a Passive Aggressive Classifier and fit the model. In the end, the accuracy score tells us how well our model performs.



TfidfVectorizer

- **TF (Term Frequency):** The number of times a word appears in a document is its Term Frequency. A higher value means a term appears more often than others, and so, the document is a good match when the term is part of the search terms.

- **IDF (Inverse Document Frequency):** Words that occur many times a document, but also occur many times in many others, may be irrelevant. IDF is a measure of how significant a term is in the entire corpus.

The TfidfVectorizer converts a collection of raw documents into a matrix of TF-IDF features.

Passive Aggressive Classifier:

- Passive Aggressive algorithms are online learning algorithms. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting. Unlike most other algorithms, it does not converge. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector.

Result and Output:

- We took a dataset, implemented a TfidfVectorizer, initialized a PassiveAggressiveClassifier, and fit our model.
- We got an accuracy of 64.29% with this model.

Limitations of model:

- It computes document similarity directly in the word-count space, which may be slow for large vocabularies
- It assumes that the counts of different words provide independent evidence of similarity.
- It makes no use of semantic similarities between words.
- **TF-IDF** is based on the bag-of-words (BoW) model, therefore it does not capture position in text, semantics.