



SENTIMENT ANALYSES OF MOVIE REVIEWS

TEAM ID: T005

TEAM MEMBERS:

دنيا إيهاب عبد البديع علي	20201700245
سما محمد العشري السيد	20201700359
هبة محمد صالح عقيلي	20201700961
مريم محمد محمود احمد	20201700818
اية محمد عثمان حسن	20201700173

INTRODUCTION:

What can we say about the success of a movie after it is released? Are there certain companies (Pixar?) that have found a consistent formula? Given that major films costing over \$100 million to produce can still flop, this question is more important than ever to the industry. Can we predict which movie reviews are positive or negative, whether they are a commercial success?

Using machine learning algorithms: Logistic Regression, SVM, KNN, and Random Forest we can predict whether this movie review is positive or negative.

DATASET PREPROCESSING:

- **TOKENIZATION:**
 - It's used to split the text into individual words or tokens.
- **STOP WORD REMOVAL:**
 - The tokens are filtered to remove stop words, which are common words that do not carry much meaning in determining sentiment.
- **PUNCTUATION REMOVAL:**
 - Each token is checked character by character, and any punctuation characters are removed. This is achieved by iterating over the characters of each token using a list comprehension.
- **LEMMATIZATION:**
 - The lemmatization process reduces words to their base or root form to capture their essential meaning, depending on the part-of-speech tag, the word is lemmatized using the appropriate POS argument ('n' for noun, 'v' for verb, 'a' for adjective, 'r' for adverb). If the tag does not match any of these, the default lemmatization is performed, assuming the word is a noun.

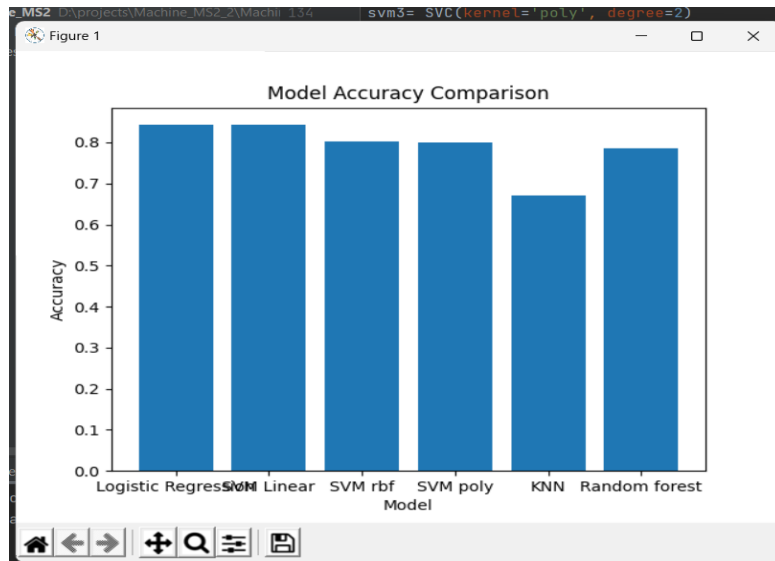
FEATURE EXTRACTION:

Using TF-IDF is a common technique in text analysis tasks, including sentiment analysis. It converts textual data into numerical features that can be fed into machine learning models for classification or regression.

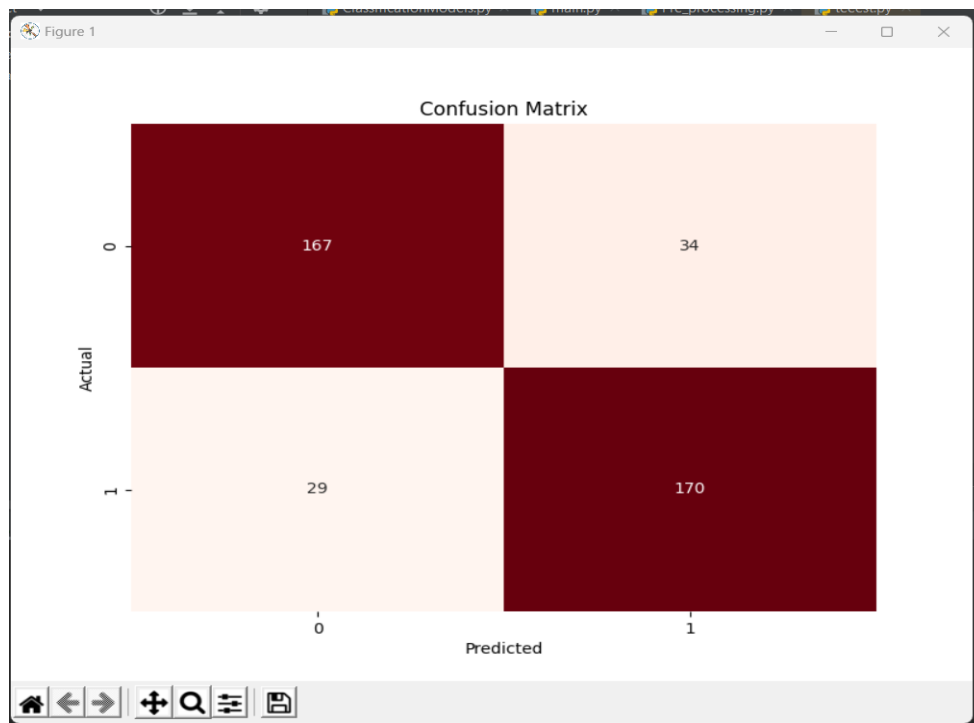
CLASSIFICATION MODELS:

- **Logistic regression:** it's used for the categorical data as the target (output) data positive or negative, it models the relationship between input features (TF-IDF vectors in this case) and the binary sentiment labels with accuracy is 0.8425.
- **SVM with linear kernel function:** it's used to find the best hyperplane that separates 2 classes with the maximum margin with accuracy 0.8425.
- **SVM with RBF kernel:** it's used to map the input data to a higher dimensional space and finds the hyperplane that separates the classes with accuracy 0.8025.
- **SVM with polynomial kernel:** it's used to find nonlinear decision boundaries and can capture curved decision boundaries with accuracy 0.8.
- **Random forest:** can provide insights into feature importance, which can be valuable in sentiment analysis by identifying the most influential words or features for sentiment classification with accuracy 0.7925.
- **KNN:** it is a non-parametric algorithm that classifies data based on the majority vote of its neighbors. It can be effective for sentiment analysis as it considers the similarity between data points.

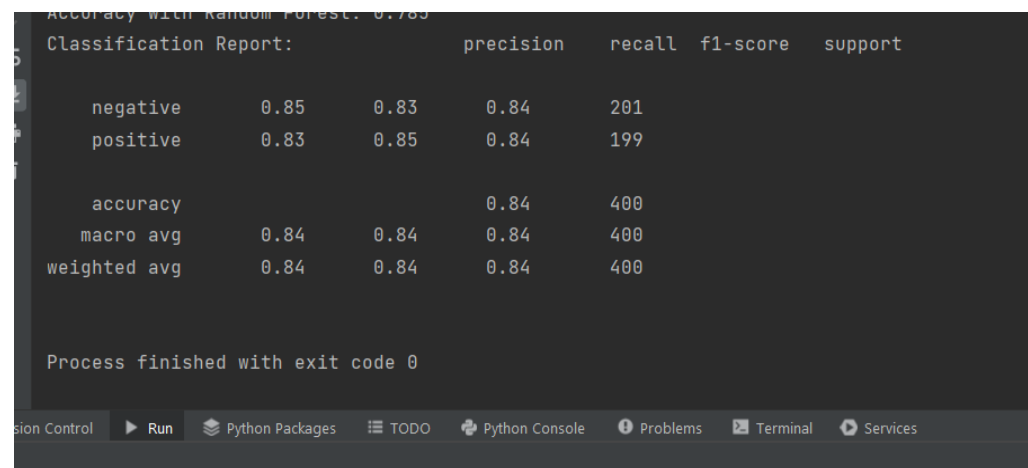
ACCURACY VISUALIZATION



CONFUSION MATRIX



CLASSIFICATION REPORT



The image shows a terminal window from a Jupyter Notebook. The output displays a classification report for a model, likely a Random Forest, with an accuracy of 0.785. The report includes precision, recall, f1-score, and support for both negative and positive classes, as well as overall accuracy and average metrics. The terminal window has a dark theme and a standard toolbar at the bottom.

```
Accuracy with Random Forest: 0.785
Classification Report:
```

			precision	recall	f1-score	support
negative	0.85	0.83	0.84	201		
positive	0.83	0.85	0.84	199		
accuracy			0.84	400		
macro avg	0.84	0.84	0.84	400		
weighted avg	0.84	0.84	0.84	400		

Process finished with exit code 0

Terminal toolbar: Run, Python Packages, TODO, Python Console, Problems, Terminal, Services