CE807- Text Analytics

Name – Akshay Janagond

Registration Id – 2322460

Overview

- Task 1: Model Selection
- Task 2: Design and Implementation of Classifiers
- Task 3: Analysis and Discussion
- Task 4: Summary
- Conclusion

Task 1: Model Selection

- **Selected Models**
- Unsupervised Model: Latent Dirichlet Allocation (LDA)
- Discriminative Classifier: Support Vector Machine (SVM)

Justification of Choices

Latent Dirichlet Allocation (LDA):

Summary: LDA is a technique used to find out topics of a text in a corpus of documents. The documents can be seen as a combination of topics and each topic seen as a distribution over words in the vocabulary. It is specifically helpful when applied to the search for thematic structure when there is no labeled data available at all.

Suitability: Thus, the LDA has been chosen, as it would perform topic modeling, which is an important task of dissecting the underlying themes of the textual data. It is highly flexible so one can easily disentangle topics and keywords that are associated with each topic.

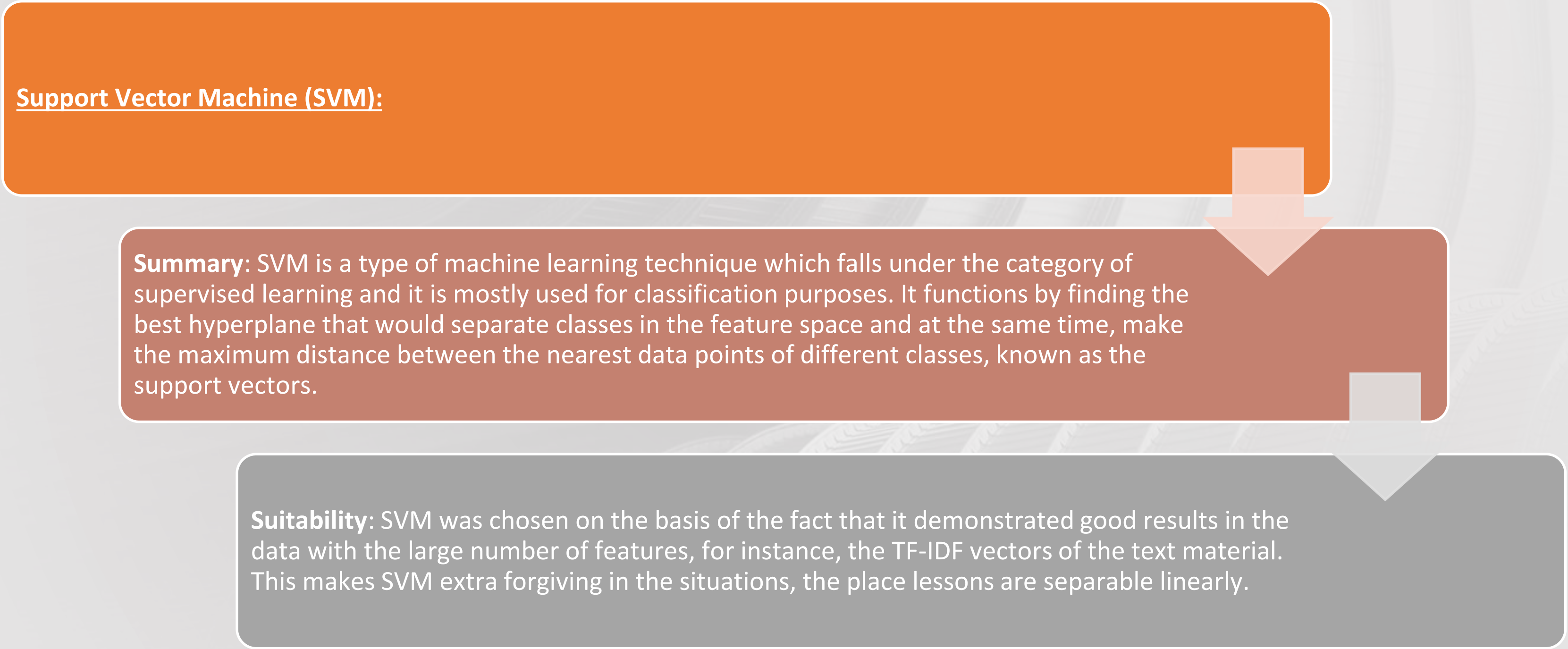
Justification of Choices

Latent Dirichlet Allocation (LDA):

Strengths: As a result of this, LDA is useful in exploring and characterizing data that has not been tagged or classified in any way. It is well scalable to large data sets of data and is relatively very efficient.

Limitations: The limitation of LDA is that it requires specification of the number of topics which may be a challenge especially for cases as it is assumed the topic number is known beforehand. It also presupposes that topics are governed by Dirichlet distribution and this might not be the real distribution in most cases.

Justification of Choices



Support Vector Machine (SVM):

Summary: SVM is a type of machine learning technique which falls under the category of supervised learning and it is mostly used for classification purposes. It functions by finding the best hyperplane that would separate classes in the feature space and at the same time, make the maximum distance between the nearest data points of different classes, known as the support vectors.

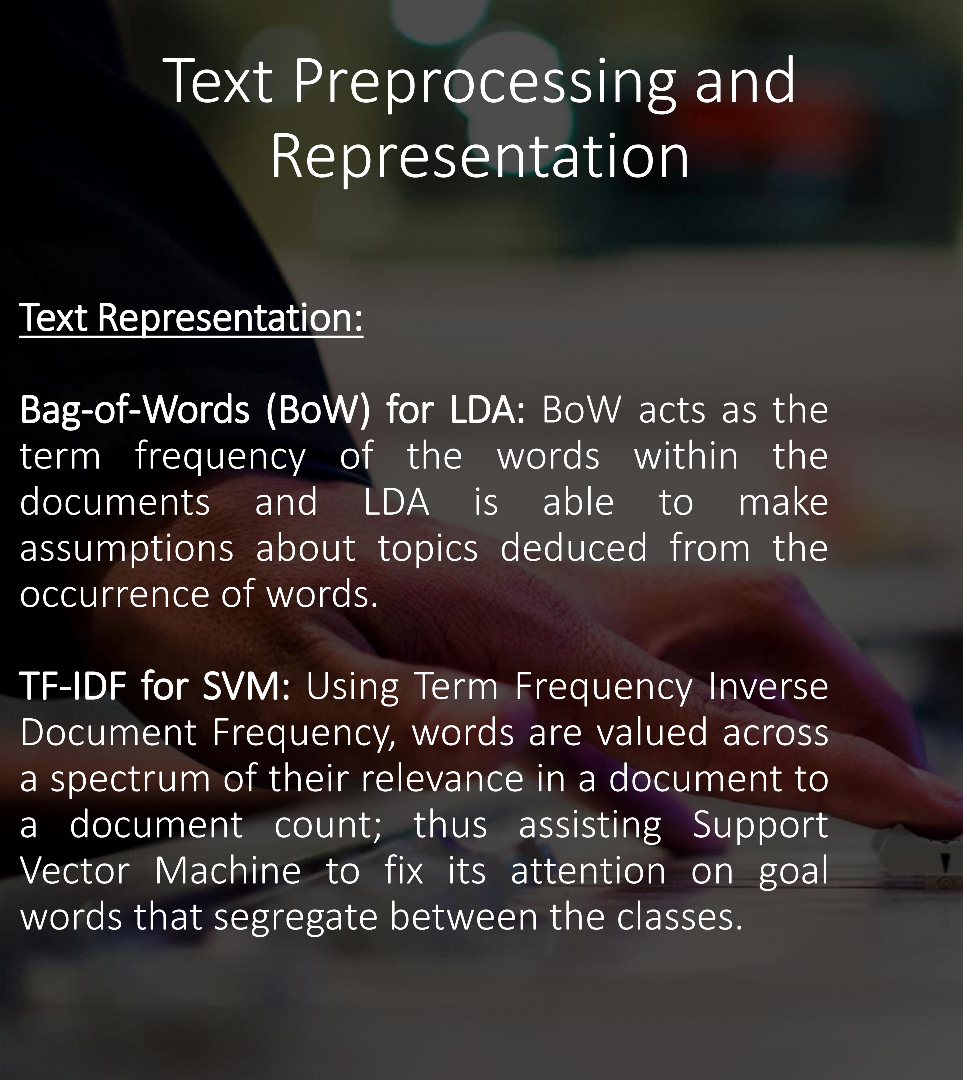
Suitability: SVM was chosen on the basis of the fact that it demonstrated good results in the data with the large number of features, for instance, the TF-IDF vectors of the text material. This makes SVM extra forgiving in the situations, the place lessons are separable linearly.

Justification of Choices

- **Support Vector Machine (SVM):**
- **Strengths:** SVM is capable for linear and non-linear classification depending on the kernel tricks. It also works well when m is much larger than n, which is quite common in the case of text data.

- **Limitations:** The shortcoming of SVMs are that it is computationally intensive particularly when working with large number of data; and not scalable. Furthermore, it is also not designed to produce probabilistic outputs of actions at the native level; however, this might be inconsequential as most packages do not do this either.

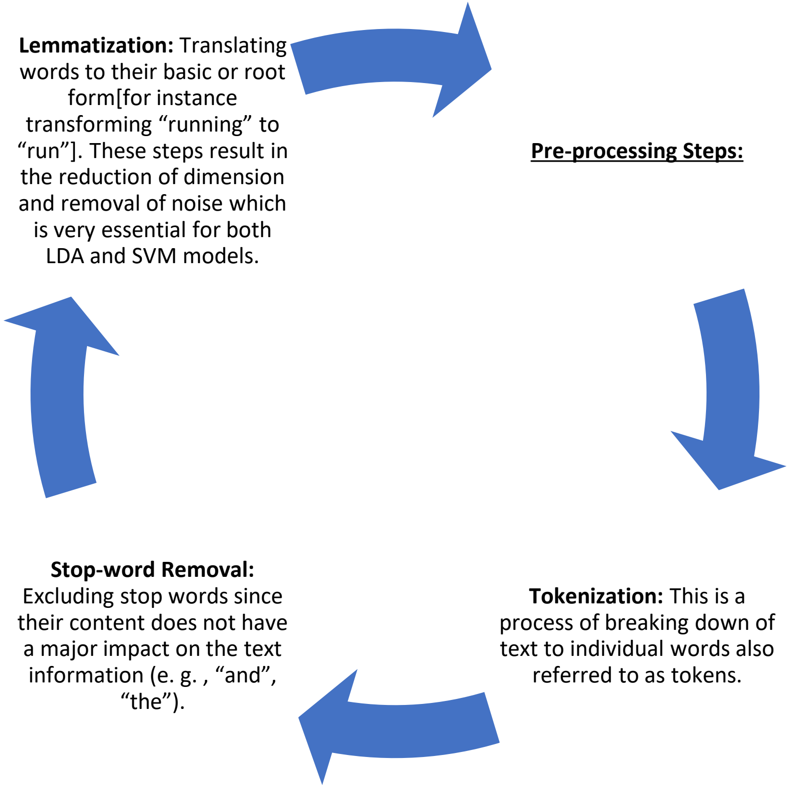
Text Preprocessing and Representation



Text Representation:

Bag-of-Words (BoW) for LDA: BoW acts as the term frequency of the words within the documents and LDA is able to make assumptions about topics deduced from the occurrence of words.

TF-IDF for SVM: Using Term Frequency Inverse Document Frequency, words are valued across a spectrum of their relevance in a document to a document count; thus assisting Support Vector Machine to fix its attention on goal words that segregate between the classes.



Pre-processing Steps:

Tokenization: This is a process of breaking down of text to individual words also referred to as tokens.

Stop-word Removal: Excluding stop words since their content does not have a major impact on the text information (e. g. , “and”, “the”).

Lemmatization: Translating words to their basic or root form[for instance transforming “running” to “run”]. These steps result in the reduction of dimension and removal of noise which is very essential for both LDA and SVM models.

Task 2: Design and Implementation of Classifiers

- **Latent Dirichlet Allocation (LDA) Model:**

- **Data Preprocessing:** Tokenized text, removed stop-words, and lemmatized data for noise reduction.

- **Feature Extraction:** Used CountVectorizer to transform text into feature vectors, focusing on the most frequent terms.

- **Model Training:** Trained LDA with 10 topics to uncover themes in the dataset.

- **Saving the Model:** Saved the LDA model and CountVectorizer for future use with joblib.

Task 2: Design and Implementation of Classifiers

- **Support Vector Machine (SVM) Model:**

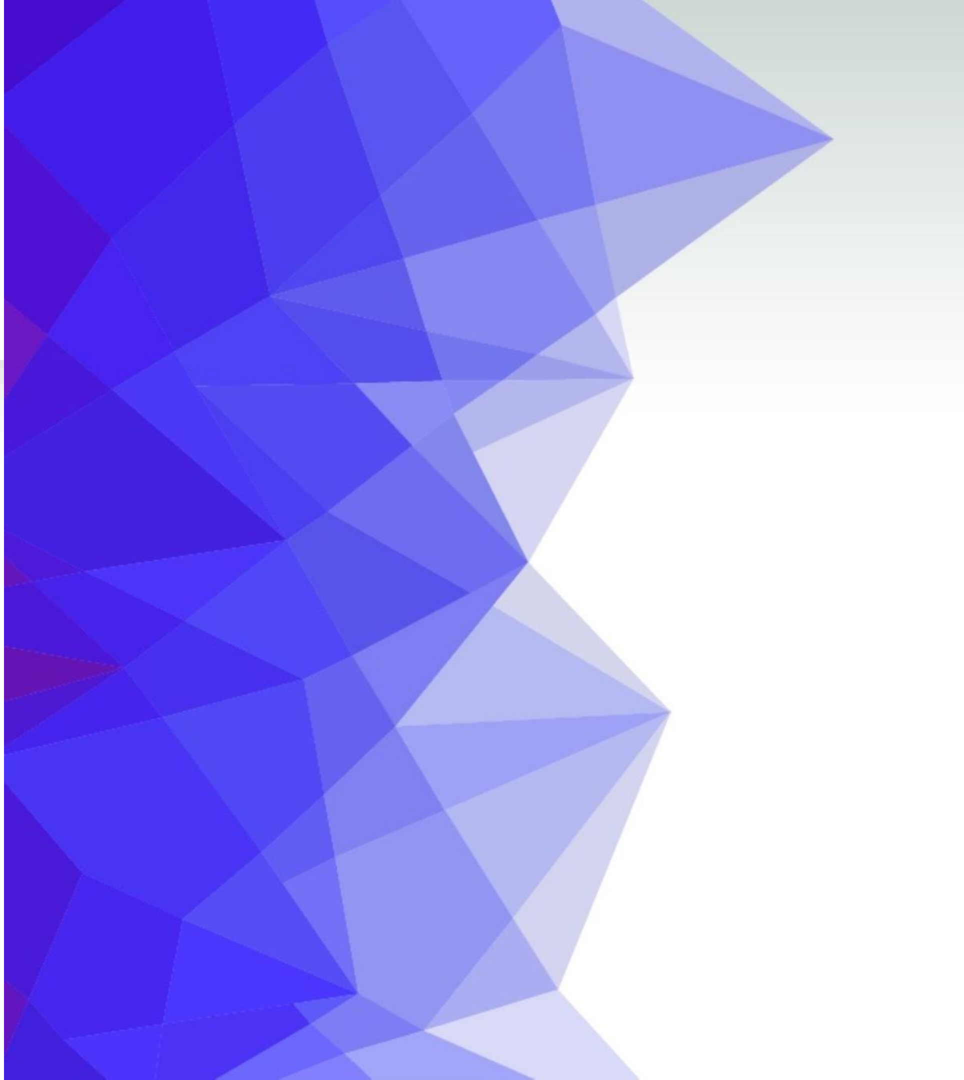
- **Data Preprocessing:** Applied similar preprocessing steps as for the LDA model.

- **Feature Extraction and Transformation:** Used TfidfVectorizer for text representation, providing better distinction for classification.

- **Model Training:** Trained SVM with a linear kernel, optimized for classification accuracy.

- **Model Evaluation:** Evaluated accuracy, precision, recall, and F1-score; achieved high accuracy for positive sentiment.

- **Model Saving:** Saved the SVM model and TF-IDF vectorizer using joblib for future analysis.



Task 2: Design and Implementation of Classifiers

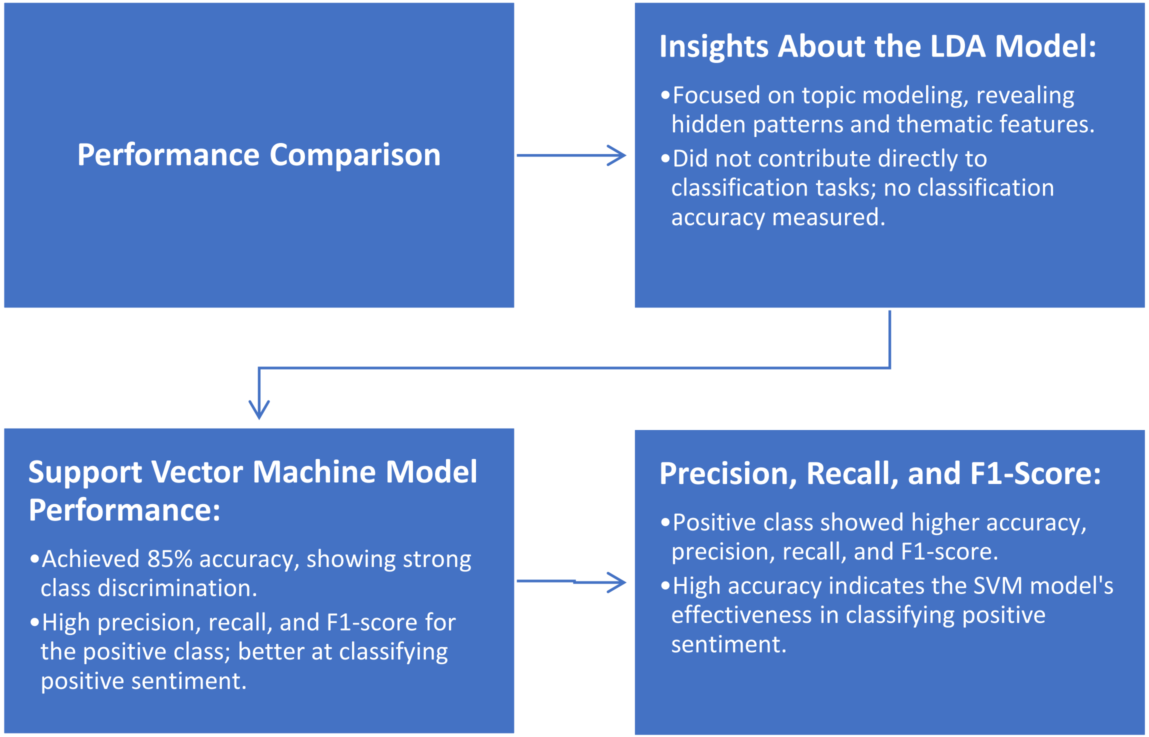
Training and Testing:

Training Process: Used preprocessed texts for LDA topic extraction and split data for SVM training and testing.

Hyperparameter Tuning: Optimized number of topics for LDA; tuned regularization parameter (C) for SVM.

Model Evaluation Metrics: Assessed models using accuracy, precision, recall, and F1-score; detailed classification report for SVM.

Task 3: Analysis and Discussion



Performance Comparison

Insights About the LDA Model:

- Focused on topic modeling, revealing hidden patterns and thematic features.
- Did not contribute directly to classification tasks; no classification accuracy measured.

Support Vector Machine Model Performance:

- Achieved 85% accuracy, showing strong class discrimination.
- High precision, recall, and F1-score for the positive class; better at classifying positive sentiment.

Precision, Recall, and F1-Score:

- Positive class showed higher accuracy, precision, recall, and F1-score.
- High accuracy indicates the SVM model's effectiveness in classifying positive sentiment.

Task 4: Summary

- **Model Selection and Justification:**
- **LDA:** Chosen for its ability to uncover hidden topics in large datasets.
- **SVM:** Selected for its effectiveness in text classification using TF-IDF vectors, with a linear kernel for sentiment classification.

Task 4: Summary

Design and Implementation:

Preprocessing: Included tokenization, stop-word removal, and lemmatization to improve input text quality.

LDA Model: Identified ten key topics within the dataset.

SVM Model: Achieved 85% accuracy on the test set using TF-IDF vectorization and linear classification.

Task 4: Summary

Analysis and Discussion:

Performance: SVM model showed higher F-scores for positive classes, indicating potential issues with class imbalance or less separable negative class features.

F1 Score: The weighted average F1-score of 0.86 demonstrates model reliability but highlights areas for improvement in handling negative sentiments.

Detailed Analysis: The model effectively detected positive sentiments but struggled with mixed or ambiguous statements, suggesting the need for better feature engineering and addressing class imbalance.

Conclusion

- The project successfully accomplished LDA and SVM algorithms for the text analysis along with results concerning content of the dataset and distribution of sentiments. The findings therefore highlight the need of stringent data preprocessing techniques, optimum choice of the model and acknowledged values for reiteration of feature construction and model assessment. Further work can be conducted in several directions: better models can be considered, for example, architectures based on transformers, and the identified shortcomings can be eliminated to improve the classification quality.