T.C.

MARMARA UNIVERSITY

FACULTY of ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

CSE4288 Introduction to Machine Learning

Term Project Final Report

Group Members

Student ID	Name Surname	Contact
150120013	İrem Aydın	irmaydin14@gmail.com
150120049	Aksanur Konuk	aksanurkonuk@gmail.com
150120054	Elife Kocabey	kocabeyelife@gmail.com
150120066	Zeynep Yılmaz	zenepyilmaz66@gmail.com
150121013	İrem Kıranmezar	iremkiranmezar@gmail.com

Abstract

The project explores sentiment analysis on tweets, a crucial task in natural language processing (NLP), aimed at classifying sentiments into positive or negative categories. Utilizing the Sentiment140 dataset, which contains 1.6 million tweets, the project focused on building a machine learning pipeline for text preprocessing, feature engineering, and model evaluation. Due to computational constraints, a subset of 10,000 balanced tweets was used. Eight machine learning models, spanning both supervised and deep learning approaches, were trained and evaluated using multiple metrics. Despite challenges such as large dataset handling and computational resource limitations, the project successfully demonstrated robust results and identified areas for improvement, paving the way for future work in sentiment analysis.

Introduction

Social media platforms, particularly Twitter, serve as a significant source for understanding public sentiment on various topics, including politics, marketing, and consumer behavior. Sentiment analysis, a subset of natural language processing (NLP), is used to classify text into categories such as positive, negative, or neutral, helping organizations make data-driven decisions. The Sentiment140 dataset, consisting of over 1.6 million labeled tweets, is widely used for developing sentiment analysis models due to its scale and diversity.

Despite its utility, sentiment analysis on social media data presents unique challenges. Tweets are often noisy, containing slang, abbreviations, and inconsistent formatting. The sheer size of datasets like Sentiment140 also demands significant computational resources. Addressing these challenges requires effective data preprocessing, feature engineering, and the application of robust machine learning techniques.

This project tackles these issues by building and evaluating machine learning models for sentiment classification. Through a systematic approach, the study aims to provide insights into effective methods for analyzing social media data and highlight best practices for handling large, noisy datasets.

Methodology

Data Preprocessing

The data preprocessing phase was critical to ensuring the quality and relevance of the dataset. Initially, the dataset was loaded using the pandas library, followed by an inspection of its structure to identify missing and duplicate values. Columns unrelated to the sentiment analysis task, such as user information and timestamps, were removed to reduce noise. Only the "Target" (sentiment) and "Text" columns were retained for analysis, as these directly contributed to the classification objective.

Text preprocessing involved converting all text to lowercase and removing unnecessary elements such as mentions, URLs, numeric characters, and punctuation. Tokenization was then performed to split the text into individual words, enabling word-level analysis. Stop-words, which do not provide significant value for sentiment prediction, were removed using predefined lists. Lemmatization was applied to convert words to their root forms, ensuring consistency and reducing redundancy. Additionally, rare words, excessive whitespace, and duplicate rows were eliminated to minimize noise and avoid overfitting. Lastly, the target variable was standardized, and the text was transformed into numerical data using the TF-IDF (Term Frequency-Inverse Document Frequency) technique, which assigns weights to words based on their frequency and importance across the dataset.

Modeling

The modeling process involved training and evaluating eight machine learning algorithms: Logistic Regression, Random Forest Classifier, Support Vector Classifier, Naive Bayes, Decision Trees (using Gini and Gain criteria), K-Nearest Neighbor, and Artificial Neural Networks (ANN). The dataset was divided into training and testing subsets, with 80% used for training and 20% reserved for evaluation. Sparse matrix formats, commonly encountered when processing text data, were converted into dense arrays to ensure compatibility with machine learning models.

Each model was trained using the training dataset and evaluated on the test dataset. Predictions were made using the predict() method, and their performance was

measured against several key metrics. Hyperparameter optimization was performed using **RandomizedSearchCV**, which searched for the optimal hyperparameters (e.g., C, solver, max_iter) by evaluating various combinations with cross-validation, ensuring the selection of the best model configuration.

Evaluation Methods

The evaluation of model performance included calculating metrics such as accuracy, precision, recall, and F1-score. Confusion matrices provided insights into the distribution of true positive, true negative, false positive, and false negative predictions, while classification reports summarized the models' overall performance. Cross-validation scores were calculated for each model to evaluate their robustness across different subsets of the training data.

This comprehensive methodology ensured the effective preprocessing, modeling, and evaluation of the dataset, facilitating the identification of the most suitable algorithm for sentiment analysis.

Results

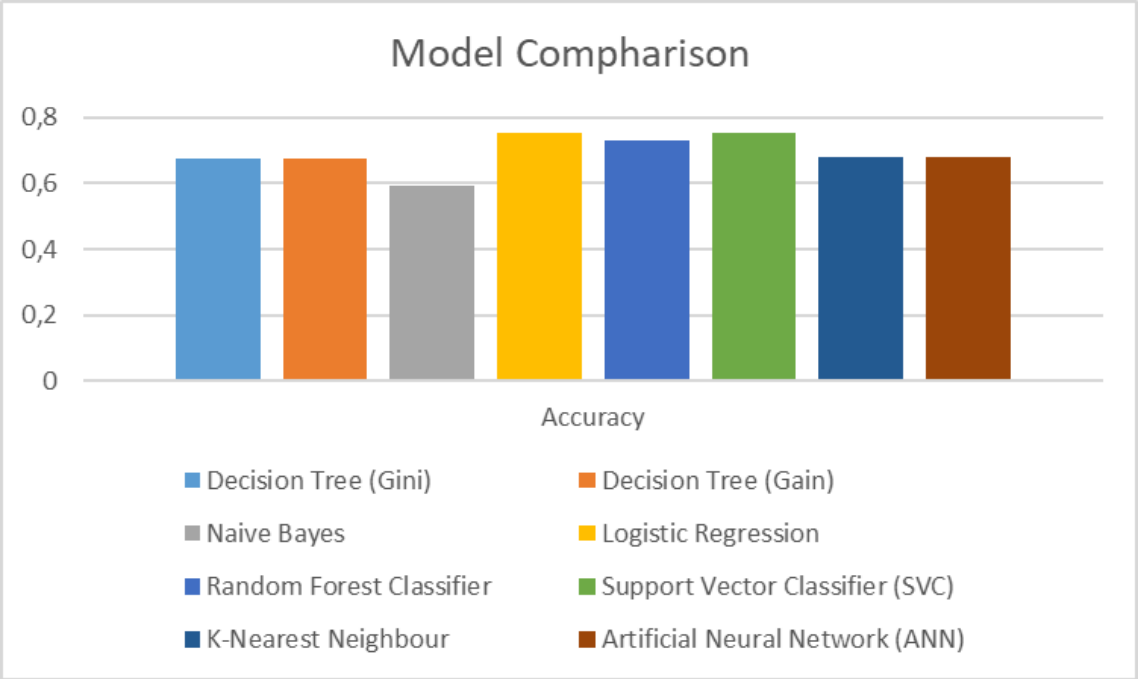

Figure 1: The Accuracy Scores of Models

```
Model: LOGISTIC REGRESSION
Accuracy Score: 0.75483
Confusion MAtrix:
[[702 268]
 [202 745]]
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.72      0.75       970
           1       0.74      0.79      0.76       947

    accuracy                           0.75      1917
   macro avg       0.76      0.76      0.75      1917
weighted avg       0.76      0.75      0.75      1917

Cross-Validation Scores:
fit_time          0.43347
score_time        0.00405
test_accuracy     0.75372
test_precision    0.73780
test_recall       0.78513
test_f1           0.76054
dtype: object
```

Figure 2: The Performance Metrics Logistic Regression

```
Fitting 5 folds for each of 18 candidates, totalling 90 fits
Best Parameters: {'solver': 'saga', 'max_iter': 300, 'C': 1}
Best Score: 0.7548843824564957
Accuracy:  0.7543
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.72      0.75       970
           1       0.74      0.78      0.76       947

    accuracy                           0.75      1917
   macro avg       0.76      0.75      0.75      1917
weighted avg       0.76      0.75      0.75      1917
```

Figure 3: The Performance Metrics of Logistic Regression After Hyperparameter Optimization

Discussion

Interpretation of Result

Logistic Regression (accuracy: 75.48%) performs well on sentiment analysis tasks, especially with text data. This model achieves high accuracy because it is sensitive to important features in the dataset, such as word frequency and term frequency. Additionally, the model provides balanced results for both classes, successfully predicting both positive and negative comments. Logistic Regression's efficiency and fast processing time make it an ideal choice for datasets like this, as it delivers solid performance while processing quickly.

Support Vector Classifier (SVC, accuracy: 75.53%) is known for its ability to handle complex, nonlinear relationships, such as those in text data. It has provided similar accuracy and balanced metrics compared to Logistic Regression. However, a major disadvantage of SVC is its longer training and prediction times, which may make it less suitable for time-sensitive applications. Nevertheless, its accuracy and balanced performance make SVC a strong option for scenarios where computation time is not a major concern.

Random Forest Classifier (accuracy: 72.82%) achieved slightly lower accuracy compared to Logistic Regression and SVC. However, it provided balanced results across classes and demonstrated very low scoring times. This is particularly advantageous when working with large datasets. Although Random Forest is effective at managing data diversity through a series of decision trees, it did not perform as well in capturing more complex relationships in text data compared to ANN or SVC.

Naive Bayes (accuracy: 59.42%) is a simple and fast model, but it showed poor performance in terms of accuracy, especially for Class 1. This is due to Naive Bayes' assumption of independence between features, which does not hold true in natural language processing tasks. Despite this, Naive Bayes successfully identified negative sentiment comments (Class 0) with a high recall rate. Its very short processing times make it suitable for scenarios that require rapid model deployment, even with lower accuracy.

Artificial Neural Networks (ANN, accuracy: 67%) are powerful for modeling complex relationships in data, making them suitable for tasks like sentiment analysis. However, ANN requires longer training times and more computational resources compared to other models. While ANN did not achieve the highest accuracy, it provided balanced results across both classes. Long training times, however, make ANN less practical for large-scale or time-sensitive tasks, though it remains effective in capturing deeper patterns in data.

K-Nearest Neighbors (KNN, accuracy: 67%) is another simple algorithm, but it struggles with high-dimensional and sparse text data. KNN achieved similar accuracy to ANN but was less efficient due to longer prediction times, making it less suitable for large datasets or real-time predictions. While KNN showed balanced results for both classes, its inefficient processing times limit its practicality compared to faster models like Logistic Regression or Naive Bayes.

The results of the models highlight the balance between accuracy and computational efficiency. Logistic Regression and SVC provided high accuracy and balanced metrics, though SVC's longer training and prediction times made it less suitable for time-critical applications. Random Forest offered fast predictions when working with large datasets, while Naive Bayes, despite lower accuracy, was ideal for scenarios requiring quick deployment. ANN and KNN produced satisfactory results but struggled with long processing times and handling large datasets. This emphasizes the importance of selecting the right model based on accuracy and performance requirements in sentiment analysis tasks.

Challenges Faced

One of the main challenges encountered during the project was the size of the dataset, which contained 1.6 million records. Such a large dataset could create problems in terms of memory usage and processing time. Therefore, instead of the entire dataset, a subset of 10,000 records was used, selected to keep the ratio of positive and negative labels in balance. This solution accelerated the data processing and model training process, providing a more efficient work and continuing to represent the general features of the large dataset. In this way, the difficulties experienced due to the size of the dataset were overcome

Conclusion

This project implemented a sentiment analysis pipeline on a large dataset of tweets, focusing on supervised learning techniques. A subset of 10,000 rows was used to address computational constraints while maintaining data representativeness. Preprocessing steps included text normalization, noise removal, tokenization, and lemmatization, ensuring data suitability for machine learning models.

Supervised models, including Logistic Regression, Random Forest, Support Vector Machines, Naive Bayes, Decision Trees, and Artificial Neural Networks, were trained and evaluated. Metrics such as accuracy, confusion matrices, and classification reports were used to assess their performance, supported by cross-validation to ensure robustness.

The project overcame challenges related to dataset size by carefully selecting a balanced subset. Future work will focus on further optimizing models and hyperparameters to improve performance, ensuring the system's reliability for real-world applications.

References

[1] https://www.kaggle.com/datasets/kazanova/sentiment140

[2] Lecture Slides