A close up of text on a white background

Description automatically generatedA picture containing black, darkness, screenshot, black and white

Description automatically generated

☐FOUNDATION ☐DIPLOMA ☒ DEGREE ☐ MASTER

*Please tick* ✓ *or click if using MS WORD*

**Marks Breakdown**

Signature: **Total**

**For office use only** – Lecturer comments (if applicable)

Signature: **Total**

*I/We declare that:*

* *This assignment is my/our own original work, except where I/we have appropriately cited the original source.*
* *This assignment or parts of it has not previously been submitted for assessment in this or any other subject.*
* *I/We allow the assessor of this assignment to test any work submitted by me/us, using text comparison software for plagiarism.****(For more information, Please read the Academic Integrity Guidelines)***

Assignment Coversheet **Please complete all details required clearly.** For softcopy submissions,please ensure this cover sheet is included at the start of your document or in the file folder.

**Assignment & Course Details:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject Code:  XBDS3034N** | | **Subject Name** *(e.g. Fundamentals of Computing*)**:**  **Natural Language Processing** | |
| **Course** *(e.g. Bachelor in Computing)* : **Bachelor of Computer Science (Hons)** | | | |
| **Lecturer Name: Ms. Sujata Navaratnam / Ms.Hemavathi Ramulu** | | | |
| **Assessment Due Date:** *(dd/mm/yy)* | **2 May 2023** | **Assessment Title:** | **Assignment: NATURAL LANGUAGE PROCESSING** |

SCCM/CSForm/Rev. Ver 2.4

**Name :**  **Student ID:**  **Email : -   
Mobile No: Signature:  
Date:**

**Name :  
Student ID:  
Email :  
Mobile No: Signature:  
Date:**

**Name :**  **Student ID:   
Email : -   
Mobile No: Signature:  
Date:**

**Name :  
Student ID:  
Email :  
Mobile No: Signature:  
Date:**

**Name :**  **Student ID:**  **Email :   
Mobile No: Signature:  
Date:**



**Name :** **Ku Ming Roong  
Student ID: 0133750  
Email : 0133750@kdu-online.com  
Mobile No: 010-4071899 Signature:  
Date: 24/04/2023**

**Table of Contents**

[1.0 Introduction 1](#_Toc133264904)

[2.0 Implementation 2](#_Toc133264905)

[3.0 Evaluation & Discussion 7](#_Toc133264906)

[4.0 Conclusion 11](#_Toc133264907)

[5.0 References 12](#_Toc133264908)

# 1.0 Introduction

This project aims to perform sentiment analysis on player reviews for a popular Steam game called Dota 2. Steam is a popular gaming platform with millions of active users worldwide. The platform allows players to purchase and play games and leave reviews to share their experiences with others.

The dataset used in this project initially contained over 6.4 million reviews for various games, but for this analysis, it was narrowed down to focus solely on Dota 2 (Larxel, 2021). Dota 2 is a multiplayer online battle arena (MOBA) game developed and published by Valve Corporation and is one of the most popular games on Steam. The Dota 2 dataset consists of around 70,000 reviews left by players, each labeled as positive or negative. The dataset used is from Kaggle – Popular Steam Game data.

Through sentiment analysis, we aim to gain insights into the overall sentiment of player reviews for Dota 2. This analysis can provide helpful information to game developers and publishers, helping them better understand the strengths and weaknesses of their games and identify areas for improvement. The following sections will discuss the methodology used to perform sentiment analysis on the Dota 2 dataset, the results obtained, and their implications. We will also discuss possible future enhancements to the project.

Research question:

* Are there any significant differences in sentiment between positive and negative reviews for Dota 2 on Steam?

Null hypothesis:

* There are no significant differences in sentiment between positive and negative reviews for Dota 2 on Steam.

Alternative hypothesis:

* There are significant differences in sentiment between positive and negative reviews for Dota 2 on Steam.

# 2.0 Implementation

Data Preprocessing - Cleaning

|  |  |
| --- | --- |
| Remove Null values | This function drops all the rows that contain null or missing values. |
| Remove Duplicate Values | This function drops all the duplicate rows from the dataset and keeps the first occurrence of the row. |
| Remove Hyperlinks and Markup | This function removes all the HTML tags, hyperlinks, and other markup from the text data. |
| Remove Numeric | This function removes all the numerical values present in the text data. |
| Remove Emoji | This function removes all the emojis from the text data. |
| Remove Symbol | This function removes all the emojis from the text data. |
| Remove Punctuation | This function removes all the punctuation marks from the text data. |
| Remove Stopwords | This function removes all the stop words (common words such as "the", "is", and "a") from the text data. |
| Unify Whitespaces | This function replaces all the multiple whitespaces with a single whitespace. |
| Normalize words using Stemming | This function normalizes the words by converting them to their base form using stemming. |

These functions help to standardize the text data and remove any unnecessary information that might affect the sentiment analysis results.

Exploratory Data Analysis (EDA)

Distribution of positive and negative reviews in the cleaned dataset.

A picture containing text, screenshot, rectangle

Description automatically generated

Based on observation, there are more positive than negative sentiment reviews.

WordCloud

|  |  |
| --- | --- |
| Distribution of Positive Sentiment    Based on positive sentiment reviews, the most common words are 'good', 'great', and 'best', as shown in the word cloud above. | Distribution of Negative Sentiment  Based on negative sentiment reviews, the most common words are 'bad' as shown in the word cloud above. |

Distribution of Number of Review Words

A picture containing text, screenshot, plot, diagram

Description automatically generated

The chart above shows that most review texts contain 1-100 words.

Distribution of Words in All Review Texts

A screenshot of a game

Description automatically generated with medium confidence

Based on the table, game & play is the highest distribution of words.

Feature Extraction – Creating Bag of Words using CountVectorizer.

A picture containing text, screenshot, font

Description automatically generated

The code splits the dataset into input (X) and target (y) and then uses the train\_test\_split function to split the data into training and testing sets. It is then transforming the text data using CountVectorizer, which converts text into a numerical format that can be used for machine learning. The stop\_words parameter removes common words that do not provide meaningful information, and max\_features is set to limit the number of features to 3000. Finally, the transformed data is stored in X\_train\_transformed and X\_test\_transformed for use in a machine-learning model.

Apply SMOTE & RUS to tackle Imbalance Class

A screen shot of a computer program

Description automatically generated with low confidence

This code resamples the data to address the class imbalance in the sentiment classification task. It uses SMOTE to oversample the minority class and RandomUnderSampler to undersample the majority class and combines them using ImbPipeline. The resampled X\_train\_transformed and y\_train are then printed to show the class proportion after resampling. The goal of this step is to balance the dataset to improve the performance of the sentiment classifier.

Model Training

The first step in the model training process is to define the pipelines for each of the three models. Each pipeline consists of a TfidfTransformer to convert the text input into numerical vectors and the respective classifier.

Next, to address the issue of imbalanced data, the training set is resampled using the SMOTE and RandomUnderSampler techniques. The SMOTE technique generates synthetic samples for the minority class (i.e., positive sentiment tweets), while the RandomUnderSampler technique randomly removes samples from the majority class (i.e., negative sentiment tweets). The resampled data is then used for training and validation using 5-fold cross-validation.

After resampling, the hyperparameters of each model are optimized using GridSearchCV. GridSearchCV is a technique that exhaustively searches over a specified hyperparameter space to find the best combination of hyperparameters for a given model. The hyperparameters and their respective values are defined in the parameter grid of each model. The performance of each model is evaluated using three scoring metrics: accuracy, f1-score, and PR\_AUC score. The PR\_AUC score is the primary metric for selecting the best hyperparameters.

Finally, the best hyperparameters for each model are selected based on the highest PR\_AUC score, and the performance of each model is printed along with the best hyperparameters, accuracy, f1-score, and PR\_AUC score. It allows us to compare the performance of the three models and select the best model for predicting tweet sentiment.

# 3.0 Evaluation & Discussion

Classification Report

|  |  |  |
| --- | --- | --- |
| Classifier | Train Set | Test Set |
| Random Forest |  |  |
| SVC |  |  |
| MultinomialNB |  |  |

Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Classifier | Train Set | Test Set |
| Random Forest |  |  |
| SVC |  |  |
| MultinomialNB |  |  |

Performance Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classifier | Accuracy (train) | Accuracy (test) | F1 Score (train) | F1 Score (test) | PR-AUC (train) | PR-AUC (test) |
| Random Forest | 0.9357 | 0.8086 | 0.9398 | 0.8101 | 0.9864 | 0.8835 |
| SVC | 0.9330 | 0.8181 | 0.9373 | 0.8260 | 0.9853 | 0.9020 |
| Multinomial  NB | 0.7858 | 0.7714 | 0.8044 | 0.7910 | 0.9064 | 0.8977 |

The Random Forest model has an accuracy of 0.9357 on the training set and an accuracy of 0.8086 on the test set. The F1 score is 0.9398 on the training set and 0.8101 on the test set. The PR-AUC score is 0.9864 on the training set and 0.8835 on the test set. Overall, this model is overfitting the training data as there is a significant drop in performance on the test set. However, the PR-AUC score suggests that the model ranks positive samples higher than negative ones.

The SVC model has an accuracy of 0.9330 on the training set and an accuracy of 0.8181 on the test set. The F1 score is 0.9373 on the training set and 0.8260 on the test set. The PR-AUC score is 0.9853 on the training set and 0.9020 on the test set. This model is also overfitting the training data, but less than the Random Forest model. The PR-AUC score is still high, suggesting that the model ranks positive samples higher than negative ones.

The Multinomial Naive Bayes (MNB) model has an accuracy of 0.7858 on the training set and an accuracy of 0.7714 on the test set. The F1 score is 0.8044 on the training set and 0.7910 on the test set. The PR-AUC score is 0.9064 on the training set and 0.8977 on the test set. This model has the lowest performance of the three models, but it is not overfitting the training data as much as the other two. The PR-AUC score is still relatively high, suggesting that the model is good at ranking positive samples higher than negative samples.

Based on the performance comparison, the SVC model shows the best performance among the three models in terms of accuracy, F1 score, and PR-AUC score on the test set. It also balances bias and variance well, indicating that the model can generalize well to new data. Therefore, the SVC model is the best candidate for further deployment to classify the Dota2 sentiment review as positive or negative. Furthermore, an ensemble model will be built from SVC and Random Forest Classifier.

Building Ensemble Model

Ensemble learning is a powerful technique that can significantly improve the performance of machine learning models by combining multiple models to produce a more accurate and robust system. The ensemble model, in this case, combines two robust classifiers - SVM and random forest - using a voting mechanism to make predictions. The "soft" voting mechanism allows for probabilistic predictions, which can be more accurate than hard voting, and helps mitigate each model's weaknesses. Moreover, the reason for building the ensemble model is that it can reduce the risk of overfitting and increase its stability. Each model in the ensemble may have its strengths and weaknesses, but by combining them, the weaknesses can be mitigated and the strengths amplified. It can result in a more accurate and robust model that can better handle complex and diverse datasets.

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy (test) | F1 Score (test) | PR-AUC (test) |
| Ensemble Model | 0.8182 | 0.8204 | 0.8902 |

This performance output suggests that the ensemble model achieved an accuracy rate of 81.82%, an F1-score of 0.8204, and a PR AUC of 0.8902. These metrics indicate that the ensemble model performs well, with a relatively high accuracy rate and F1 score. The PR AUC score is also quite good, indicating that the model can predict positive and negative cases accurately. The ensemble model is better at identifying positive and negative cases, while the SVM may better predict positive cases. Overall, the ensemble model provides a solid and robust classifier that can outperform individual models and reduce the risk of overfitting.

Simple Interactive UI

After the model building and evaluation process, a simple interactive UI has been made for predicting the Dota 2 review. Using the UI, users can easily input their Steam game reviews for Dota2 and receive a sentiment analysis prediction from one of the models. It can help users quickly understand the sentiment of their reviews and make informed decisions about their opinions on the game. The UI can also provide options for selecting which model to use and displaying the model's confidence in its prediction. With a user-friendly interface, this sentiment analysis project can become a valuable tool for anyone interested in understanding the sentiment of Dota2 game reviews. Below are some figures to show the UI.

A screenshot of a computer game

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

# 4.0 Conclusion

In conclusion, our study demonstrated the effectiveness of using machine learning techniques for sentiment analysis in the context of popular Steam game reviews. Our best-performing model, an ensemble of SVM and random forest, achieved an accuracy rate of 0.8086, an F1-score of 0.8204, and a PR-AUC score of 0.8902, which outperformed the individual models. We also showed that resampling techniques such as SMOTE and random under-sampling could improve model performance on imbalanced datasets.

As for future enhancements, one area of improvement could be incorporating deep learning techniques such as neural networks or recurrent neural networks (RNNs) to capture more complex relationships between words in the text input (Laskowski & Contributor, 2021). Another possibility is to explore transfer learning approaches, where pre-trained language models such as BERT or GPT can be fine-tuned on the Steam review dataset to boost performance further (Joshi, 2022). Additionally, incorporating user features such as game hours played, review history, or community standing could provide valuable sentiment analysis information. Finally, exploring additional text processing techniques, such as topic modeling or sentiment lexicons, could also enhance model performance.

# 5.0 References

Joshi, P. (2022, June 20). *Transfer learning for NLP: Fine-tuning bert for text classification*. Analytics Vidhya. Retrieved April 24, 2023, from https://www.analyticsvidhya.com/blog/2020/07/transfer-learning-for-nlp-fine-tuning-bert-for-text-classification/

Larxel. (2021, November 29). *Steam reviews*. Kaggle. Retrieved April 24, 2023, from https://www.kaggle.com/datasets/andrewmvd/steam-reviews

Laskowski, N., & Contributor, T. T. (2021, July 12). *What are recurrent neural networks and how do they work?* Enterprise AI. Retrieved April 24, 2023, from https://www.techtarget.com/searchenterpriseai/definition/recurrent-neural-networks