T.C.

MARMARA UNIVERSITY

FACULTY of ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

CSE4288 Introduction to Machine

Learning Term Project

Data Preprocessing 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

1. Introduction

This project focuses on sentiment analysis of tweets to classify them as positive or negative. The dataset used is the **Sentiment140** dataset, which consists of six columns:

Feature	**Explanation**
Target (Sentiment)	The polarity of the tweet (0:negative, 4:positive)
Tweet ID	A unique id for each tweet (2087)
Date	The date of the tweet. (Sat May 16 23:58:44 UTC 2009)
Flag	The query (lyx). If there is no query, then this value is NO_QUERY
User	The X (Twitter) username of the person who posted the tweet
Text	The actual content of the tweet, which represents the opinions or emotions expressed by the user

For this analysis, only the target and text features were used. The primary goals were:

1. Clean and preprocess the data to improve model performance.
2. Explore the dataset using visualizations to uncover patterns.
3. Prepare features for modeling.

2. Data Cleaning Steps

The data cleaning process is designed to improve the quality and relevance of the data. Before this step, the dataset was read using the pandas library, specifically the pd.read_csv function, which allowed for efficient loading and inspection. To better understand the structure of the data, the head(), tail(), and info() functions were used to display a sample of rows and summarize the data types and counts. Additionally, missing and duplicate values were checked using pandas' isnull().sum() and duplicated().sum() functions.

2.1 Retaining Relevant Columns

Only the target (sentiment) and text columns are kept to reduce unnecessary data and focus on the classification task. These columns were included in the data analysis because they carry critical information that directly contributes to the main task of the model. In contrast, other columns such as user information and date were considered as unnecessary data that would not contribute to the performance of the model and were removed from the analysis process.

2.2 Text Preprocessing

Text preprocessing involves converting all text to lowercase using pandas' str.lower() method. Noise removal was accomplished with the help of the re module for regular expressions. Patterns such as mentions (@username), URLs (http://, https://, www), and numeric characters were removed using re.sub(). To eliminate punctuation, Python’s string module was used with the string.punctuation attribute, combined with the translate() method. These steps cleaned and standardized the text for analysis.

2.3 Tokenization and Removing Stop-words

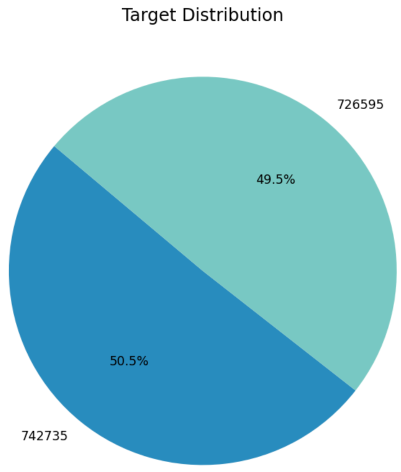
The tokenization process splits each tweet into individual words, enabling the model to process and analyze each word separately, using the word_tokenize() function from the NLTKlibrary. Stop words, such as "the" "is" and "and", these words do not carry significant meaning in predictive modeling and may skew results if not removed, were removed to focus on meaningful words and reduce noise in the dataset. The stop word list was accessed using nltk.corpus.stopwords.words('english'), ensuring only significant terms were retained for analysis.

3. EDA Findings with Visualizations

The Exploratory Data Analysis phase was used to understand the basic properties of the processed data set, develop a basic understanding of the data structure, and identify potential problems.

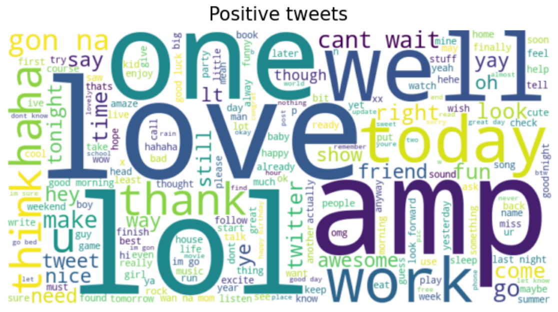
3.1 Target Distribution

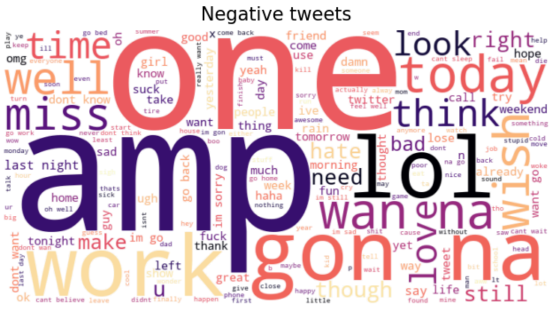
Using a balanced dataset has an impact on the model’s ability to make accurate predictions. When the dataset is unbalanced, the model will often prioritize the majority class label, which negatively impacts the model’s performance. Therefore, firstly, the positive and negative class labels in our dataset were analyzed with a pie chart. The positive rate was found to be 49.5% and the negative rate was found to be 50.5%. This situation shows that the negative and positive class labels have a balanced distribution in the dataset, and the model can learn equally both class labels.



3.2 Word Cloud for Positive and Negative Sentiments

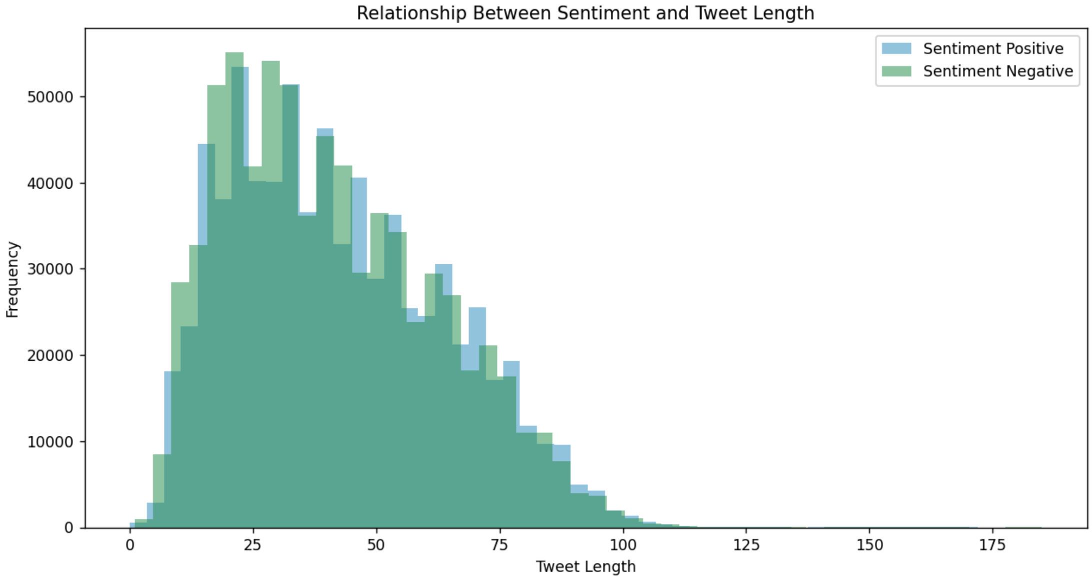
We used a WordCloud to determine the most frequently used words in the positive and negative classes and their frequency. When we examined this graph, we observed that words representing positive emotions such as **love, thank, and well** were prominent in the positive class, while negative and complaining words such as **miss, don’t, and work** were used more in negative tweets. This analysis helped us understand the frequency of repetition of words belonging to both classes and the emotional load they carried.





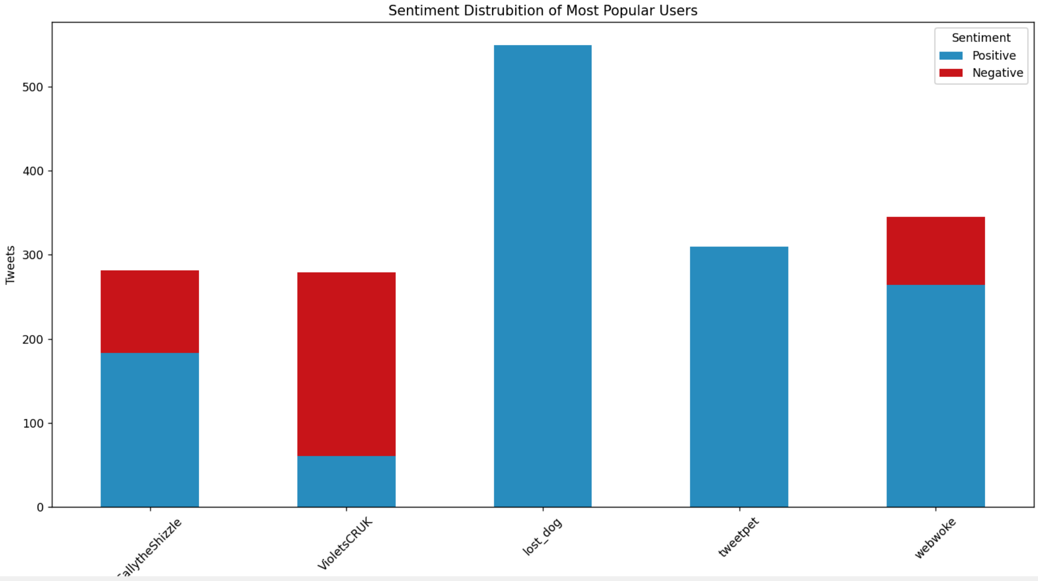
3.3 Tweet Length Distribution by Sentiment

We examined the relationship between the class labels and tweet lengths using a histogram graph. The distribution of tweet lengths reveals a similar pattern for both negative and positive class labels, indicating that the tweet length is not a strong discriminative feature for classifying tweets. Additionally, it was observed that tweets were most concentrated between 10 and 50 characters, while longer tweets were less frequent.



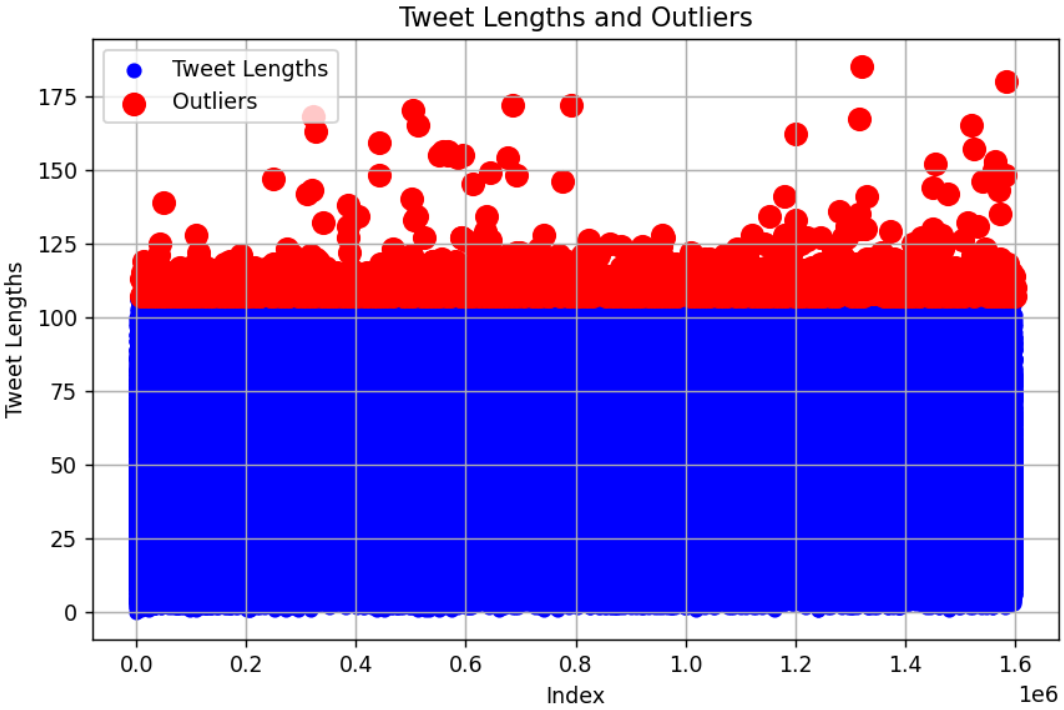
3.4 Sentiment Distribution per User

We analyzed negative and positive class labels on a per-user basis.



3.5 Outlier Detection

Outliers can negatively affect the accuracy of the dataset, so identifying unusual or extreme points is of great importance. In this project, outliers were detected based on tweet lengths and using the z-score method. With this method, tweet lengths are evaluated according to the average tweet length, and values greater than 3 or less than -3 are considered outliers. The calculation results are visualized in the scatter plot below.



4. Feature Selection and Engineering

4.1 Lemmatization

Lemmatization is a technique used in language processing processes and aims to convert words to their base or root forms. In this way, formal differences between words are eliminated. For example, the words "run", "runs" and "running" are reduced to the root "run" with the lemmatization process. This method collects words with different forms of the same meaning into a single root word and provides a more accurate analysis.

If lemmatization is not used, each form is recorded separately, which increases the memory load. Additionally, evaluating each form separately causes the model to learn unnecessary details. These situations negatively affect the efficiency and performance of the model. These problems are overcome and the performance of the model is improved using the lemmatization method.

4.2 Whitespace, Rare Word, and Duplicate Row Removal

Words that occur once in the dataset generally carry less meaningful information for sentiment analysis and may create noise. By removing these words, the model focuses on more frequently used words. Additionally, learning rare words can cause the model to memorize these words, which can lead to overfitting. As a result, removing rare words and unnecessary spaces improves the accuracy and performance of the model. It also reduces complexity and improves generalization ability by allowing the model to run with fewer words.

Duplicated rows cause the model to learn the same information more than once. This increases the model's bias on this information and cannot evaluate all data equally. This can result in the model being overfitting. By removing duplicated data, this situation is eliminated and allows the data to be learned more accurately.

4.5 Target Variable Adjustment and Feature Vectorization

Conversion of "Target" values from 4 to 1 was done to standardize the target variable, ensuring that the model can better interpret and predict outcomes. This step ensures consistency in how target values are represented.

The model to be trained can only understand numerical data, not text data directly. The TF-IDF (Term Frequency-Inverse Document Frequency) method is a widely used technique for transforming text into numerical data. The TF-IDF method assigns numerical values to words by taking into account two criteria. Term Frequency (TF) measures how often a word is used within a tweet, while Inverse Document Frequency (IDF) considers the rarity of the word in the entire tweet dataset. In this way, common and less meaningful words such as "and" are assigned lower values, while rarer and meaningful words are given higher numerical values. This transformation helps the model capture the importance of words relative to the entire dataset, allowing it to focus on more meaningful words while reducing the influence of less important terms.