**Kathmandu University**

**Department of Computer Science and Engineering**

**Dhulikhel, Kavre**



**A Project Report**

**on**

**“NepSense”**

**[Code No: COMP 308]**

**(For partial fulfillment of 3rd Year/ 2nd Semester in Computer Engineering)**

**Submitted by**

**Abrial Adhikari (02)**

**Prashant Manandhar (30)**

**Samir Wagle (60)**

**Reewaj Khanal (61)**

**Submitted to**

**Mr. Dhiraj Shrestha**

**Department of Computer Science and Engineering**

**Submission Date:06/17/2024**

**Bona fide Certificate**

**This project work on**

**“NepSense”**

**is the bona fide work of**

**“Abiral Adhikari (027923-20)**

**Prashant Manandhar (027951-20)**

**Samir Wagle (027980-20)**

**Reewaj Khanal (028006-20)”**

**who carried out the project work under my supervision.**

**Project Supervisor**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Bal Krishna Bal,**  
**Associate Professor and Head of Department,**  
**Department of Computer Science and Engineering,**

**Date:**

# Acknowledgement

We would like to express our deepest gratitude to our esteemed project supervisor, Dr. Bal Krisha Bal, for his invaluable support and guidance throughout this project. His insightful feedback and exceptional expertise were instrumental in ensuring the successful completion of this work. We are particularly grateful for his constant encouragement, which motivated us to persevere through challenges.

We are also thankful to the Department of Computer Science and Engineering at Kathmandu University for providing the necessary resources and fostering a conducive environment for research and development. This project would not have been possible without their support.

Finally, we extend our sincere appreciation to all those who, directly or indirectly, contributed to the successful culmination of this project.

Thank you.

**Team NepSense**

**Abiral Adhikari**

**Prashant Manandhar**

**Samir Wagle**

**Reewaj Khanal**

# Abstract

Nowadays almost all social media sites on the internet have automatic foul language detection and barring algorithms implemented. But such implementations are only limited to high resource languages like English, Spanish, German, etc. while there no such mechanism for the abusive content written on low resource language like Nepali. So, with this project we are proposing a profanity detection system for Nepali Language. The project will employ recurrent neural networks, a bidirectional LSTM model for machine learning to detect the profanity in Nepali language text using a dataset consisting of a bunch of profane and offensive comments. The expected outcome of this project is to create a service enabling foul language barring services to Nepali sites and content, while improving and supplementing the already conducted research on the said field. This project is meant to generalize the profanity detection models available for Nepali language as a meaningful service while aiding the precursor studies conducted on above topic.

**Keywords:** Social Media, Profanity, Nepali, NLP, Deep Learning, Bidirectional LSTM, Offensive.

Table of Contents

[Acknowledgement i](#_Toc169391922)

[Abstract ii](#_Toc169391923)

[List of Figures vi](#_Toc169391924)

[List of Tables viii](#_Toc169391925)

[Acronyms/Abbreviations ix](#_Toc169391926)

[Chapter 1 Introduction 1](#_Toc169391927)

[1.1 Background 1](#_Toc169391928)

[1.2 Objectives 2](#_Toc169391929)

[1.3 Motivation and Significance 2](#_Toc169391930)

[Chapter 2 Related Works 4](#_Toc169391931)

[2.1 Aspect Based Abusive Sentiment Detection in Nepali Social Media Texts (Oyesh Mann Singh; Sandesh Timilsina; Bal Krishna Bal; Anupam Joshi) 4](#_Toc169391932)

[2.2 Hate speech Detection in Asian Languages: A Survey (L K Dhanya; Kannan Balakrishnan) 5](#_Toc169391933)

[2.3 Offensive Language Detection in Nepali Social Media (Nobal B. Niraula, Saurab Dulal, Diwa Koirala) 5](#_Toc169391934)

[2.4 Comparison of Machine Learning Algorithms for Hate and Offensive Speech Detection 6](#_Toc169391935)

[Chapter 3 Design and Implementation 7](#_Toc169391936)

[3.1 Research And Study: 7](#_Toc169391937)

[3.1.1 Profanity Detection Methods Examination: 7](#_Toc169391938)

[3.1.2 Nepali Language Corpus Compilation: 7](#_Toc169391939)

[3.1.3 Machine Learning Model Experimentation: 7](#_Toc169391940)

[3.1.4 Social Media Profanity Analysis: 7](#_Toc169391941)

[3.2 Data Gathering and Selection: 8](#_Toc169391942)

[3.3 Annotation of dataset: 8](#_Toc169391943)

[3.4 Data Preprocessing: 9](#_Toc169391944)

[3.5 Model Design and Training: 9](#_Toc169391945)

[3.5.1 Multi Output LSTM model: 10](#_Toc169391946)

[3.5.2 Single Output LSTM model for Profanity: 11](#_Toc169391947)

[3.5.3 Single Output LSTM model for Offensiveness: 11](#_Toc169391948)

[3.5.4 Single Output LSTM model for both Offensiveness and Profanity. 12](#_Toc169391949)

[3.6 Result and Analysis: 13](#_Toc169391950)

[3.6.1 Multi Output LSTM model: 13](#_Toc169391951)

[3.6.2 Single Output LSTM model for Profanity: 16](#_Toc169391952)

[3.6.3 Single Output LSTM model for offensiveness: 18](#_Toc169391953)

[3.6.4 Single Output Multilabel LSTM for offensiveness and profanity: 20](#_Toc169391954)

[3.7 Flowchart: 23](#_Toc169391955)

[3.7.1 Annotation of dataset 23](#_Toc169391956)

[3.7.2 Preprocessing 25](#_Toc169391957)

[3.7.3 Model training 26](#_Toc169391958)

[3.8 Use case: 27](#_Toc169391959)

[3.9 Data Flow Diagram (DFD) 28](#_Toc169391960)

[3.10 Decision Tree and Diagram 31](#_Toc169391961)

[3.11 Nature of Data 32](#_Toc169391962)

[3.12 Sequence Diagram 33](#_Toc169391963)

[3.13 State Diagram 34](#_Toc169391964)

[3.14 System Requirement Specifications 35](#_Toc169391965)

[3.14.1 Software Specifications 35](#_Toc169391966)

[3.14.2 Hardware Specifications 35](#_Toc169391967)

[Chapter 4 Development Tools Specifications 36](#_Toc169391968)

[4.1 Colab 36](#_Toc169391969)

[4.2 Kaggle 36](#_Toc169391970)

[4.3 Apify 36](#_Toc169391971)

[4.4 ML tools 37](#_Toc169391972)

[Chapter 5 Discussion on the achievements 39](#_Toc169391973)

[5.1 Challenges 39](#_Toc169391974)

[5.2 Features: 40](#_Toc169391975)

[Chapter 6 Conclusion and Recommendation 41](#_Toc169391976)

[6.1 Limitations 41](#_Toc169391977)

[6.2 Future Enhancement 42](#_Toc169391978)

[References 43](#_Toc169391979)

[APPENDIX 44](#_Toc169391980)

[6.3 d 44](#_Toc169391981)

# List of Figures

[Figure 3.6.1 Confusion matrix of Multi-Output Model 14](#_Toc169391890)

[Figure 3.6.2 Graph of accuracy (profanity and offensive) of Multi Output LSTM Gender Classification Report 14](#_Toc169391891)

[Figure 3.6.3 Graph of loss (profanity and offensive) of Multi Output LSTM Gender Classification Report 15](#_Toc169391892)

[Figure 3.6.4 Graph of accuracy (gender) Multi Output LSTM Gender Classification Report 15](#_Toc169391893)

[Figure 3.6.5 Graph of accuracy of Single Output LSTM model for Profanity 16](#_Toc169391894)

[Figure 3.6.6 Figure 3.6.2 Graph of loss of Single Output LSTM model for Profanity 17](#_Toc169391895)

[Figure 3.6.7 Confusion matrix of Single output model for Profanity 17](#_Toc169391896)

[Figure 3.6.8 Graph of accuracy of Single output model for offensive 18](#_Toc169391897)

[Figure 3.6.9 Graph of loss of Single output model for offensive 19](#_Toc169391898)

[Figure 3.6.10 Confusion matrix of Single output model for offensive 19](#_Toc169391899)

[Figure 3.6.11 Graph of accuracy of Single output multilabel model for both offensiveness and profanity 20](#_Toc169391900)

[Figure 3.6.12 Graph of loss of Single output multilabel model for both offensiveness and profanity 21](#_Toc169391901)

[Figure 3.6.13 Confusion matrix of Single output multilabel model for both offensiveness and profanity 21](#_Toc169391902)

[Figure 3.7.1 Flowchart of Annotation of dataset 24](#_Toc169391903)

[Figure 3.7.2 Flowchart of preprocessing 25](#_Toc169391904)

[Figure 3.7.3 Flowchart of Model Training 26](#_Toc169391905)

[Figure 3.8.1 Use Case diagram 27](#_Toc169391906)

[Figure 3.9.1 Context Diagram (DFD) 28](#_Toc169391907)

[Figure 3.9.2 Level 0 Diagram 29](#_Toc169391908)

[Figure 3.9.3 Level 1 Diagram 30](#_Toc169391909)

[Figure 3.10.1 Decision Tree for preprocessing 31](#_Toc169391910)

[Figure 3.12.1 Sequence diagram 33](#_Toc169391911)

[Figure 3.13.1 State Diagram 34](#_Toc169391912)

[Figure 6.3.1 Gannt Chart 44](#_Toc169391913)

[Figure 6.3.2 Output-1 from all models 45](#_Toc169391914)

[Figure 6.3.3 Output-2 from all models 45](#_Toc169391915)

# List of Tables

[Table 3.6‑1 Multi Output LSTM Gender Classification Report 13](#_Toc169391916)

[Table 3.6‑2 Multi Output LSTM Profanity Classification Report 13](#_Toc169391917)

[Table 3.6‑3 Single Output LSTM model for Profanity Classification Report 16](#_Toc169391918)

[Table 3.6‑4 Single Output LSTM model for offensiveness Classification Report 18](#_Toc169391919)

[Table 3.6‑5 Single Output Multilabel LSTM for both offensiveness and profanity Classification Report 20](#_Toc169391920)

[Table 3.10‑1 Decision Table for annotation and labeling 32](#_Toc169391921)

# Acronyms/Abbreviations

NLP Natural Language Processing

ML Machine Learning

LSTM Long Short-Term Memory

RNN Recurrent Neural Network

CSS Cascading Style Sheet

API Application Programming Interface

BERT Bidirectional Encoder Representations from Transformers

RAM Random Access Memory

GB Giga Byte

GHz Giga Hertz

1. Introduction

## Background

The usage of digital content from various social media and various online platforms is increasing exponentially day by day. As the content on the internet is increasing, we need an effective way to supervise those platforms for any malicious and harmful behavior from the user. One of the sectors that must be monitored effectively is the detection of profane language. There are many profanity detection methods that are available on the internet and currently being used in various platforms concurrently. But the available models are mostly available in high resource languages like English, Hindi, Chinese language. On the language that requires attention is Nepali, as its significant speakership and growing digital presence of Nepali content. Profanity detection falls under a subset of natural language processing (NLP), which plays a crucial role in various applications, including text classification, sentiment analysis, and machine translation. This proposal aims to outline the development of a profanity detection system specifically tailored for the Nepali language.

Nepali is the official language of Nepal. Nepali is not only a recognized language in Nepal only but also used vastly in India, Bhutan, and some part of Myanmar. As the 19 million native speaker and increasing day by day, Nepali language hold the linguistics importance, both globally and regionally. Despite its larger significance, various existing language detection systems often overlook Nepali due to its relatively lower digital footprint compared to languages like English, Spanish, Hindi or Chinese.

The proposed profanity detection system, NepSense will utilize machine learning algorithms like deep learning (LSTM) trained on a dataset comprising Nepali text samples that are scraped from the various social media platform like Facebook, website using the python scrapping techniques and Apify tools. To guarantee the detection system's resilience in various scenarios, this dataset will comprise a wide variety of linguistic expressions, including formal, informal, and colloquial ones.

## Objectives

The main objectives of our application are as follows.

* To develop a comprehensive dataset of Nepali text samples for training the profanity detection system.
* To experiment the different models with different embedding layers and find the result of machine learning algorithms suitable for profanity detection.
* To decrease the usage of profanity and offensive language in the social media platform.

## Motivation and Significance

The choice of development of the profanity detection that mainly focusses on the Nepali is the lack of the effective model and underrepresentation of the linguistics importance. Nepali serves as the vital medium of the communication for not only us but for the millions of the people. However, due to the lack of effective model for the supervision of the online platform in the Nepali language poses significance challenges in the digital content processing, limiting accessibility, and hindering effective communication for Nepali speakers worldwide.

The goal of the project is to address the shortcomings of current language detection systems, specifically regarding their poor accuracy and limited coverage of Nepali text. Due to a lack of specialized algorithms that are tailored for the complexities of the Nepali language, linguistic variations, and inadequate training data, existing systems frequently have difficulty correctly identifying Nepali content. The goal of this project is to close this gap and enhance Nepali natural language processing capabilities by creating a specialized profanity detection system.

The project's main characteristics are its dedication to developing a strong and trustworthy profanity detection system, its thorough approach to addressing linguistic challenges, and its focus on Nepali language processing. This project has the potential to significantly improve digital communication, content accessibility, and linguistic diversity preservation in the Nepali-speaking community by closing the gap in language processing capabilities for Nepali.

1. Related Works

Profanity detection has garnered significant attention in both academia and industry due to its importance in maintaining online civility and safety. A multitude of studies and projects have explored various techniques and approaches for identifying and filtering inappropriate language in digital content. In this section, we review some of the key contributions in the field of profanity detection, categorizing them based on their methodologies and applications.

## Aspect Based Abusive Sentiment Detection in Nepali Social Media Texts (Oyesh Mann Singh; Sandesh Timilsina; Bal Krishna Bal; Anupam Joshi)

This excerpt outlines a research project focused on fine-grained sentiment analysis of social media comments in the Nepali language. It highlights the lack of benchmarked datasets for low-resource languages like Nepali, despite the availability of such datasets for high-resource languages such as English, French, and German, particularly in specific domains like restaurants, hotels, or electronic goods.

The proposed work aims to address this gap by creating a dataset specifically tailored for aspect-based sentiment analysis in the social media domain. This dataset includes code-mixed and code-switched comments extracted from Nepali YouTube videos. The research also involves setting up a dataset benchmark and evaluating various machine learning models using this dataset.

The initial results presented in the excerpt showcase promising baselines achieved using a multilingual BERT model for aspect term extraction and a Bi-LSTM model for sentiment classification tasks. These models achieved F1 scores of 57.978% and 81.60%, respectively.

## Hate speech Detection in Asian Languages: A Survey (L K Dhanya; Kannan Balakrishnan)

This study provides a comprehensive survey of hate speech detection in Asian languages, focusing particularly on the development of an automated detection system for Malayalam. The motivation behind this survey stems from the necessity to combat messages spreading negativity on social media platforms, encompassing topics such as sex, caste, religion, politics, and race. Detecting such hateful content poses significant challenges. The research exclusively examines language-specific studies on hate speech detection and analyzes the methodologies employed in each study. Three parameters are utilized to assess the overall landscape of hate speech detection across Asian languages. The study aims to determine the most effective classification algorithm for this task and explore the relationship between classification approaches, dataset types, sizes, and accuracy levels. By providing this survey, the research lays the groundwork for future investigations in this field and contributes to a deeper understanding of the associated challenges.

## Offensive Language Detection in Nepali Social Media (Nobal B. Niraula, Saurab Dulal, Diwa Koirala)

This paper addresses the critical issue of detecting offensive language in social media texts, including blog posts, comments, and tweets. While existing approaches primarily target resource-rich languages like English and German, this research focuses on characterizing offensive language in Nepali, a low-resource language. The study highlights the challenges associated with processing Nepali social media text and presents experiments using supervised machine learning for offensive language detection. In addition to providing the first baseline approaches for detecting offensive language in Nepali, the paper also releases human-annotated datasets to encourage future research in this crucial area of study.

## Comparison of Machine Learning Algorithms for Hate and Offensive Speech Detection

This paper highlights the prevalence of hate speech on various networking platforms, particularly exacerbated by recent events such as the COVID-19 pandemic and associated lockdowns, which have led to an increased use of social media for communication. Hate speech significantly impacts individuals' lives and can even lead to suicidal events, underscoring the urgent need for its mitigation. Detection serves as the initial step towards addressing this issue. The paper presents a comparative analysis of different machine learning algorithms for detecting hate speech, utilizing data from the Twitter platform. Through the analysis, it was concluded that the long short-term memory (LSTM) method emerges as a highly effective machine learning algorithm for this purpose.

1. Design and Implementation

## Research And Study:

The research and study phase of the NepSense project involves several key areas that aim at developing an efficient profanity detection system for the Nepali language.

### Profanity Detection Methods Examination:

In this paper, a study of the existing methods for detecting and recognizing offensive language in various languages including English, Hindi, and Chinese is done. Thus, we would like to examine the underlying computational strategies, efficiency and constraints that these techniques have so that they may be used as blueprint for their application in Nepali.

### Nepali Language Corpus Compilation:

The collection of diversified and representative sample of Nepali texts from various online platforms such as YouTube, Twitter, news sources, websites is essential. In this regard, we will need an intensive analysis of linguistic features that occur in the text under scrutiny based on which our corpus will be built upon serving as training data for our profane detection system.

### Machine Learning Model Experimentation:

One important aspect of our research is therefore experimenting with different machine learning models including deep learning architectures such as Long Short-Term Memory (LSTM). The goal is to evaluate their performance in detecting profanities from Nepali texts with respect to their optimization parameters.

### Social Media Profanity Analysis:

Understanding the prevalence and impact of profanity language on social media platforms within the Nepali-speaking community is essential. This analysis involves studying user behaviors, community guidelines, and platform policies related to profanity, enabling us to tailor our detection system to effectively address these issues.

## Data Gathering and Selection:

For our research on profanity detection, we collected data from various social media platforms, including Facebook and YouTube. This involved extensive web scraping of user comments to build a robust dataset. By targeting these platforms, we ensured a controversial, trending, profane posts, videos in diverse range of Nepali linguistic styles and contexts, enhancing the comprehensiveness and reliability of our dataset. We used Apify console to extract the YouTube and Facebook comment. Apify console extracts all the comments from the post whose link is to be provided by us. We extracted the comment in CSV format and performed the necessary preprocessing after that.

## Annotation of dataset:

Annotation of the dataset is carried out by first installing the necessary required python libraries. After that to-be-annotated dataset, which was scrapped from Facebook, is imported. The filter keyword, which was downloaded from Kaggle, NepSA dataset, and manually added keyword are imported. The imported keyword is filtered out for the removal of the duplication of keyword. From these datasets, 647 profanity words along with 1855 offensive words are used for annotation.

As project is majorly concerned with the Nepali language, to-be-annotated dataset and filter keyword dataset are converted into Nepali language. For the conversion purpose, three python package is used i.e. “ai4bharat”, “Nepali\_nlp”, “nepali\_unicode\_converter”. All three-package output was added in new column and result was observed. For most of the time, the performance of the “ai4bharat” "outperforms the other two packages. So, for this reason, result from “ai4bharat” is taken for further annotation. After the conversion of English word from both dataset is successfully converted into Nepali word, the Nepali converted dataset is filtered out and tagged whether the text is either profanity or offensive with the help of regular expression.

The gender dataset consists of name and gender, published during the election was imported from Kaggle, and other gender data are scrapped from various website are scrapped. As majority of the name was in Nepali, these names are converted into English names as name to be detected are in English language. Now the profanity and offensive tagged dataset is now again labeled for the gender. After that the dataset is saved in the file.

The saved file was again manually crossed checked for any error, wrong labelling. After the successfully manually crossed checked, all the small file was combined into one file and that file was ready to be used in preprocessing.

## Data Preprocessing:

In the preprocessing phase, necessary libraries were installed and the manually crossed checked annotated data was imported. These emojis contained in the text were removed using the regular expression. Then the text was converted into base word using the stemming with the help of the “Nepali\_nlp” package. The stop word in Nepali language is imported from the NLTK library and these stop words are removed from the text. As the punctuation does not hold more meaning to profanity and offensiveness, so this punctuation was removed by first making the list of punction and checking whether it exists in sentence or not if yes remove them. The duplicated comment data was removed from the dataset. Words longer than 10 characters were also removed because it was harder to analyze. After this preprocessed text was doing the file.

## Model Design and Training:

In this project, the model design and training process involved the development of four distinct models to perform various classification tasks based on text input. The primary model was a LSTM-based multi-output neural network that predicts both gender (a binary classification 0 for female and 1 for male) and profanity levels (a multilabel classification with three classes: None (0), offensive only (1), and profane (2)). The secondary models consist of two single-output binary classifier models and a single-output multilabel classifier. The first of these was a binary classifier that determines whether the text was profane (1) or not (0), and the second was a binary classifier that assessed whether the text was offensive (1) or not (0). The multilabel classifier determined whether text was neither profane nor offensive (0), offensive only (1) or profane (2). The LSTM architecture was chosen for its ability to capture temporal dependencies and contextual information within sequences of text, making it well-suited for natural language processing tasks. The details of model design and training are provided below:

### Multi Output LSTM model:

This model took the dense vector embedding obtained from the “multilingual-bert-cased" transformer model as input at input layer. The model contained a dropout layer with dropout of 0.4 after input layer followed by 3 Bidirectional LSTM layer third one having the max norm constraint 3. This was followed by a dense layer with “relu” activation and regularization of 0.0001. After this there were 2 dense output layer first for gender output with “sigmoid” activation returning single value between 0 to 1 with threshold being 0.5 and second layer for profanity classification with “softmax” activation giving 3 output prediction for each class. The model was compiled wiTOOth “Adam” optimizer with learning rate of 0.001, “categorical\_crossentropy” loss function for profanity output and “binary\_crossentropy” loss function for gender output. The model was then trained iteratively finally till the optimal results were obtained for training parameters: epochs of 100 and batch size of 128. The model trained with early stopping at patience level 3 taking validation split 0.2 with metrics for early stopping being validation loss. The model summary is provided in figure below:

The model was trained with accuracy of 0.65 and model training graphs, confusion matrix and classification reports were generated for analysis.

### Single Output LSTM model for Profanity:

The input layer of the model was an embedding layer to convert dense vector representation of text to continuous vector space using vocab size of dataset to determine size of embedding matrix. The second layer was a dropout layer of 0.7 to prevent overfitting Then there were two Bidirectional LSTM layer with second on having max norm constraint of 3. The next layer was a Dense layer with “relu” activation and L2 regularization of 0.0001. This is followed by another dropout layer of 0.7 for prevention of overfitting. The output layer was a dense layer with "sigmoid" activation function giving 2 outputs prediction for 2 labels (not profane and profane). The model was compiled with "Adam" optimizer at learning rate of 0.00006 and "binary\_crossentropy" loss function. The model was trained iteratively till optimal results were obtained at following parameters: epoch of 30 and batch size of 128. Early stopping was implemented to prevent overfitting with validation split of 0.2, patience level 5 with metric for early stopping being validation loss. The model summary is provided in figure below:

The model was trained with accuracy of 0.878 and model training graphs, confusion matrix and classification reports were generated for analysis.

### Single Output LSTM model for Offensiveness:

The input layer of the model was an embedding layer to convert dense vector representation of text to continuous vector space using vocab size of dataset to determine size of embedding matrix. The second layer was dropout layer of 0.7 to prevent overfitting. Then there were two Bidirectional LSTM layers with the second one having max norm constraint of 3. The next layer was a Dense layer with “relu” activation and L2 regularization of 0.0001. This was followed by another dropout layer of 0.7 for prevention of overfitting. The output layer was a dense layer with "sigmoid" activation function giving 2 outputs prediction for 2 labels (not offensive and offensive). The model was compiled with "Adam" optimizer at learning rate of 0.00006 and "binary\_crossentropy" loss function. The model was trained iteratively till optimal results were obtained at following parameters: epoch of 30 and batch size of 128. Early stopping was implemented to prevent overfitting with validation split of 0.2, patience level 5 with metric for early stopping being validation loss. The model summary is provided in figure below:

The model was trained with accuracy of 0.782 and model training graphs, confusion matrix and classification reports were generated for analysis.

### Single Output LSTM model for both Offensiveness and Profanity.

The input layer of the model was an embedding layer to convert dense vector representation of text to continuous vector space using the vocab size of the dataset to determine the size of the embedding matrix. The second layer was a dropout layer of 0.7 to prevent overfitting. Then there were two Bidirectional LSTM layers, with the second one having a max norm constraint of 3. The next layer was a Dense layer with “relu” activation and L2 regularization of 0.0001. This was followed by another dropout layer of 0.7 to prevent overfitting. The output layer was a dense layer with a "softmax" activation function giving 3 output predictions for 3 labels (not offensive nor profane, offensive, and profane). The model was compiled with the "Adam" optimizer at a learning rate of 0.00006 and a "categorical\_crossentropy" loss function. The model was trained iteratively until optimal results were obtained with the following parameters: epoch of 30 and batch size of 128. Early stopping was implemented to prevent overfitting with a validation split of 0.2, a patience level of 5, with the metric for early stopping being validation loss. The model summary is provided in the figure below:

The model was trained with an accuracy of 0.663 and model training graphs, confusion matrix, and classification reports were generated for analysis.

## Result and Analysis:

The graphs, classification reports and confusion matrix obtained after training and testing the model on test and validation dataset for all the models are provided below from which we obtained insights about out dataset and model results.

### Multi Output LSTM model:

Table ‑ Multi Output LSTM Gender Classification Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Multi Output LSTM Gender Classification Report** | | | | |
|
|  | **Precision** | **Recall** | **F1-Score** | **Support** |
| **Class Female** | 0 | 0 | 0 | 70 |
| **Class Male** | 0.87 | 1 | 0.93 | 454 |
| **Accuracy** |  |  | 0.87 | 524 |

Table ‑ Multi Output LSTM Profanity Classification Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Multi Output LSTM Profanity Classification Report** | | | | |
|  |
|  | **Precision** | **Recall** | **F1-Score** | **Support** |  |
| **Class None** | 0.66 | 0.58 | 0.62 | 179 |  |
| **Class Offensive** | 0.47 | 0.58 | 0.52 | 169 |  |
| **Class Profane** | 0.72 | 0.65 | 0.68 | 176 |  |
| **Accuracy** |  |  | 0.6 | 524 |  |

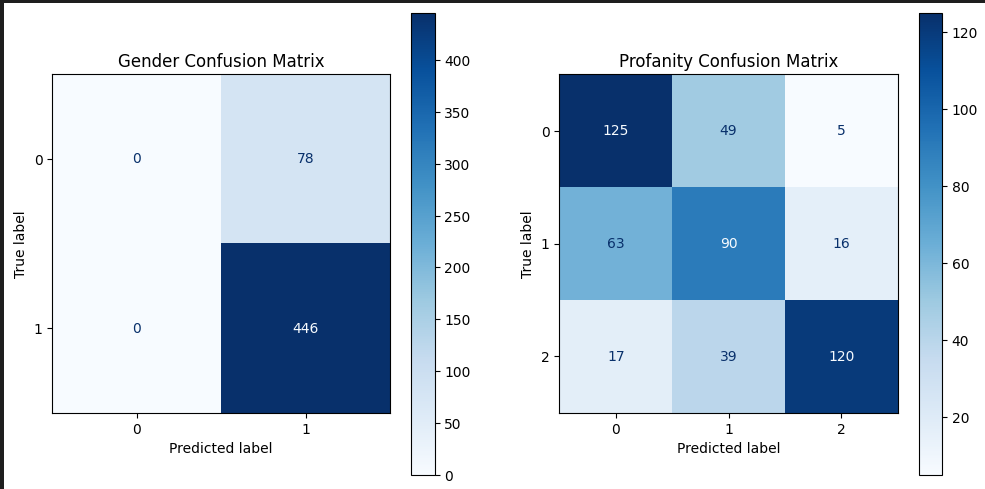


Figure . Confusion matrix of Multi-Output Model

A graph with blue and orange lines

Description automatically generated

Figure . Graph of accuracy (profanity and offensive) of Multi Output LSTM Gender Classification Report

A graph of a graph with blue and orange lines

Description automatically generated

Figure . Graph of loss (profanity and offensive) of Multi Output LSTM Gender Classification Report

A graph with numbers and lines

Description automatically generatedw

Figure . Graph of accuracy (gender) Multi Output LSTM Gender Classification Report

### Single Output LSTM model for Profanity:

Table ‑ Single Output LSTM model for Profanity Classification Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Single Output LSTM model for Profanity Classification Report:** | | | | |
|  |
| **Profanity Model** | **Precision** | **Recall** | **F1-Score** | **Support** |  |
| **Class None** | 0.74 | 0.73 | 0.74 | 175 |  |
| **Class Profane** | 0.73 | 0.74 | 0.74 | 174 |  |
| **Accuracy** |  |  | 0.74 | 349 |  |

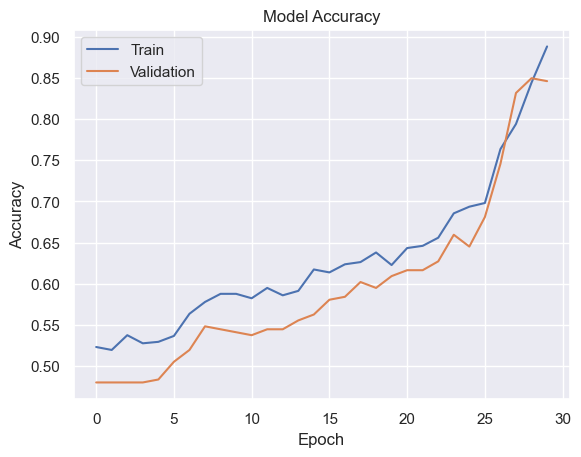


Figure . Graph of accuracy of Single Output LSTM model for Profanity

A graph with a line and a curve

Description automatically generated

Figure . Figure 3.6.2 Graph of loss of Single Output LSTM model for Profanity

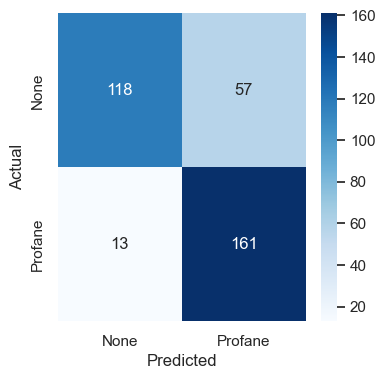


Figure . Confusion matrix of Single output model for Profanity

### Single Output LSTM model for offensiveness:

Table ‑ Single Output LSTM model for offensiveness Classification Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Single Output LSTM model for offensiveness Classification Report** | | | | |
|
| **Offensive Model** | **Precision** | **Recall** | **F1-Score** | **Support** |
| **Class None** | 0.68 | 0.92 | 0.78 | 175 |
| **Class Offensive** | 0.87 | 0.56 | 0.68 | 174 |
| **Accuracy** |  |  | 0.74 | 349 |
|  |  |  |  |  |

A graph showing the value of a train

Description automatically generated

Figure . Graph of accuracy of Single output model for offensive

A graph with a line and a curve

Description automatically generated

Figure . Graph of loss of Single output model for offensive

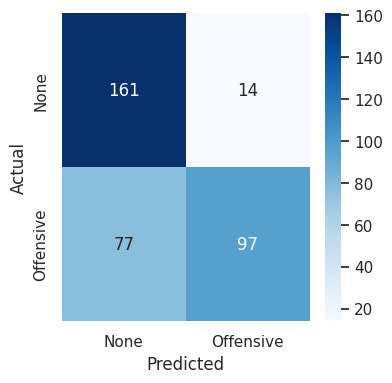


Figure . Confusion matrix of Single output model for offensive

### Single Output Multilabel LSTM for offensiveness and profanity:

Table ‑ Single Output Multilabel LSTM for both offensiveness and profanity Classification Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Single Output Multilabel LSTM for both offensiveness and profanity Classification Report** | | | | |
|
| **Multilabel Model** | **Precision** | **Recall** | **F1-Score** | **Support** |
| **Class None** | 0.54 | 0.64 | 0.58 | 174 |
| **Class Offensive** | 0.57 | 0.51 | 0.54 | 175 |
| **Class Profane** | 0.81 | 0.75 | 0.78 | 175 |
| **Accuracy** |  |  | 0.63 | 524 |

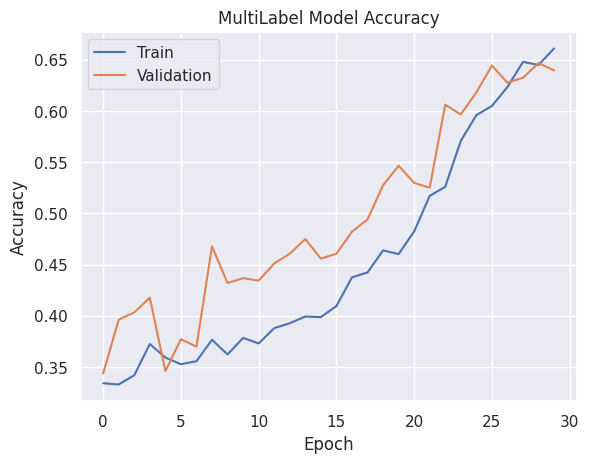


Figure . Graph of accuracy of Single output multilabel model for both offensiveness and profanity

A graph with a line graph and numbers

Description automatically generated

Figure . Graph of loss of Single output multilabel model for both offensiveness and profanity

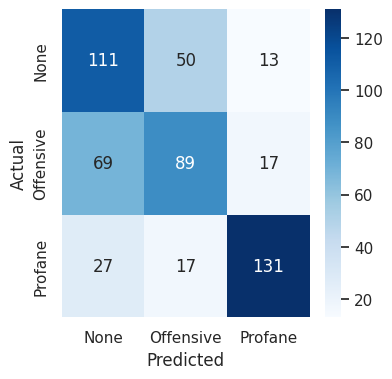


Figure . Confusion matrix of Single output multilabel model for both offensiveness and profanity

From study of the classification report for precision, recall and f1-score of all the models along with the confusion matrix for the test set we observed that the classification of the profane comments is working well. But the model both multilabel classification and binary classification for the offensive comments is not producing desired result, showing low accuracy, precision, recall and f1 score when compared with the profane comment classification conducted with same hyperparameters for the models.

From the confusion matrix of both multilabel classification and binary classification for the offensive comment we found the models were making mispredictions in case of offensiveness. The confusion matrix indicated that the models were predicting the comment to be non-offensive while the comments are offensive in most cases. On further inspection the results showed that the precision for the offensive classification was higher than the recall. This meant that for the offensive classifications, comments predicted to be offensive by the models were correct most of the time, but the models failed to correctly predict most of the offensive comments as offensive instead predicting them as not offensive.

Another result that was observed was that the multi output multilabel model with BERT embedding showed slightly better result than the single output multilabel model that used custom vocab for embedding. We also observed that both multilabel models showed significantly higher loss and lower accuracy than the binary classifier model.

The multioutput model showed high accuracy in predicting the gender of the author, with almost all predictions indicating male authors. This result may had been due to the class imbalance in the dataset, where the number of male authors were significantly larger than the number of female authors, leading to biased predictions. But this observation also indicated that of the person posting profane and offensive comments in Facebook are usually male.

The study of the results of training and testing of all the above models indicated that the dataset must be updated for proper selection, cleaning and labelling of the offensive comments and redesigning of the offensive comment classification model while the profanity model needed no change.

## Flowchart:

### Annotation of dataset

The flow chart of the process of annotation of the dataset used in our project is given below:

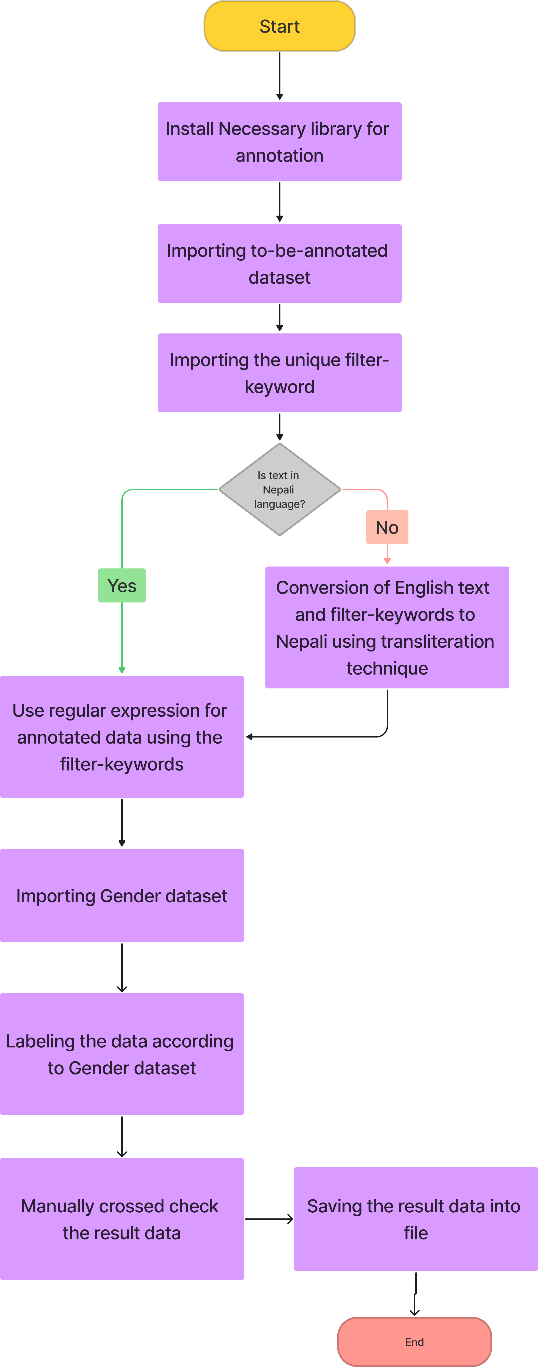


Figure . Flowchart of Annotation of dataset

### Preprocessing

The flow chart of the process of preprocessing of the dataset used in our project is given below:

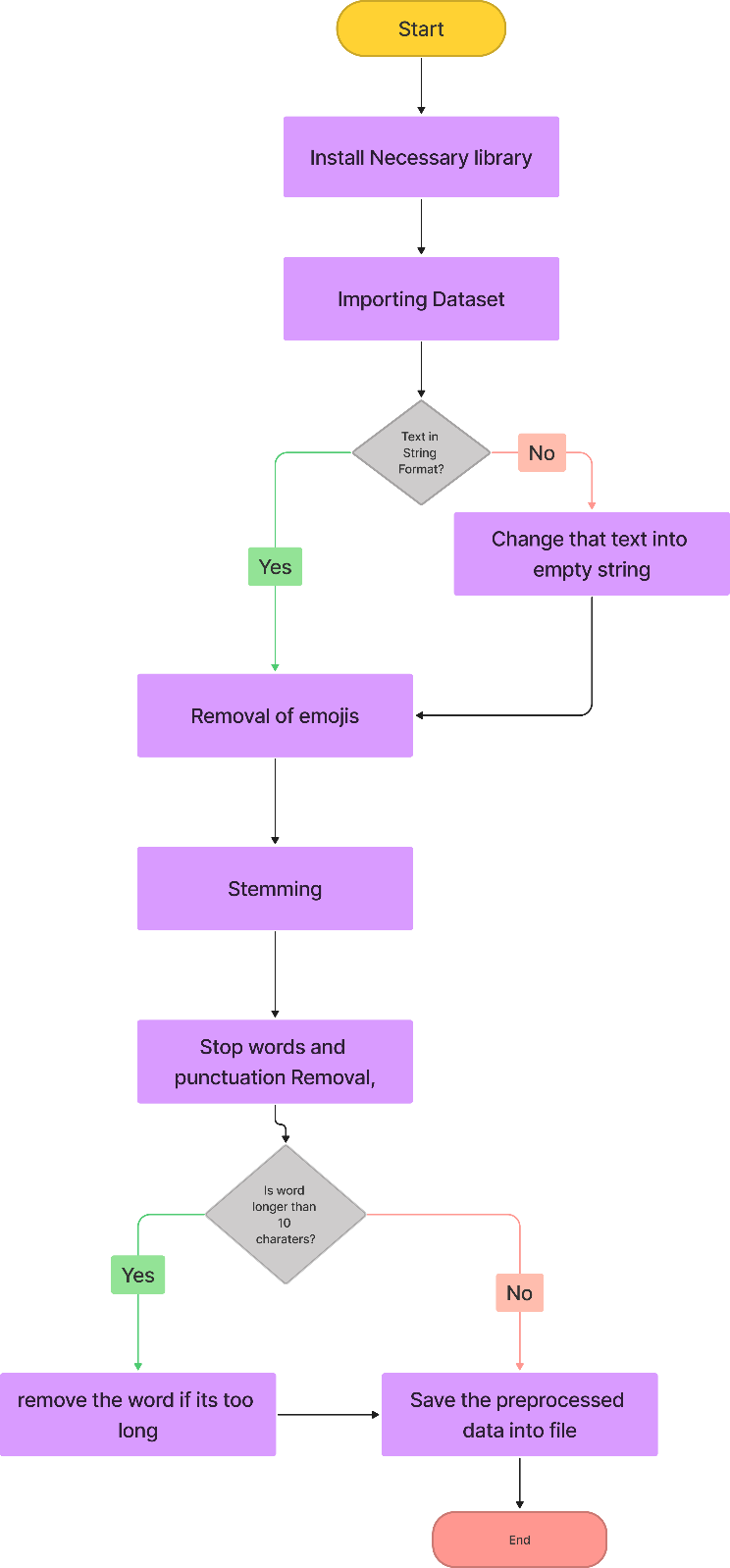


Figure . Flowchart of preprocessing

### Model training

The flow chart of the process of model training in our project is given below:

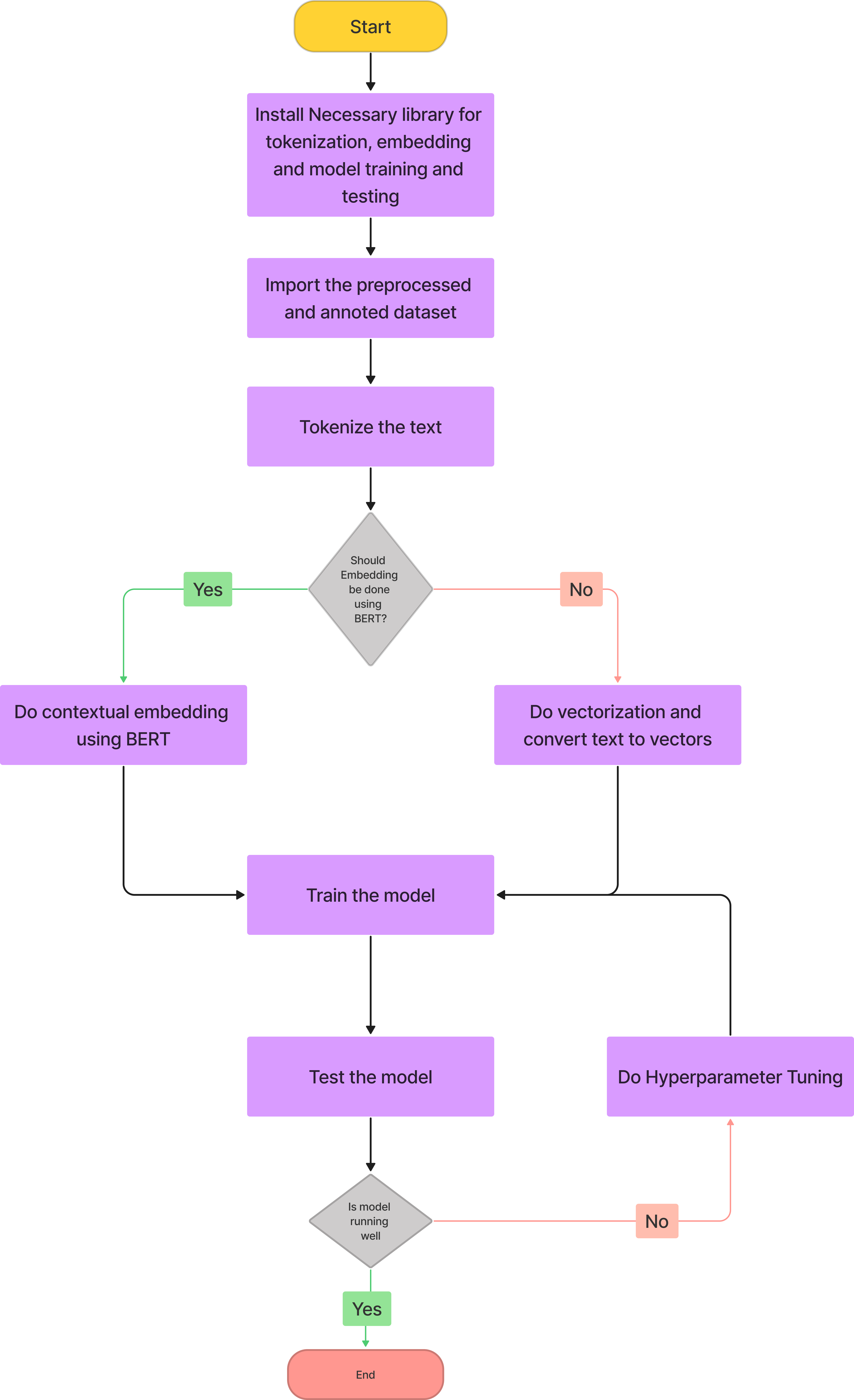


Figure . Flowchart of Model Training

## Use case:

The use case of the overall project of detecting the profanity and offensive in the text is shown below:

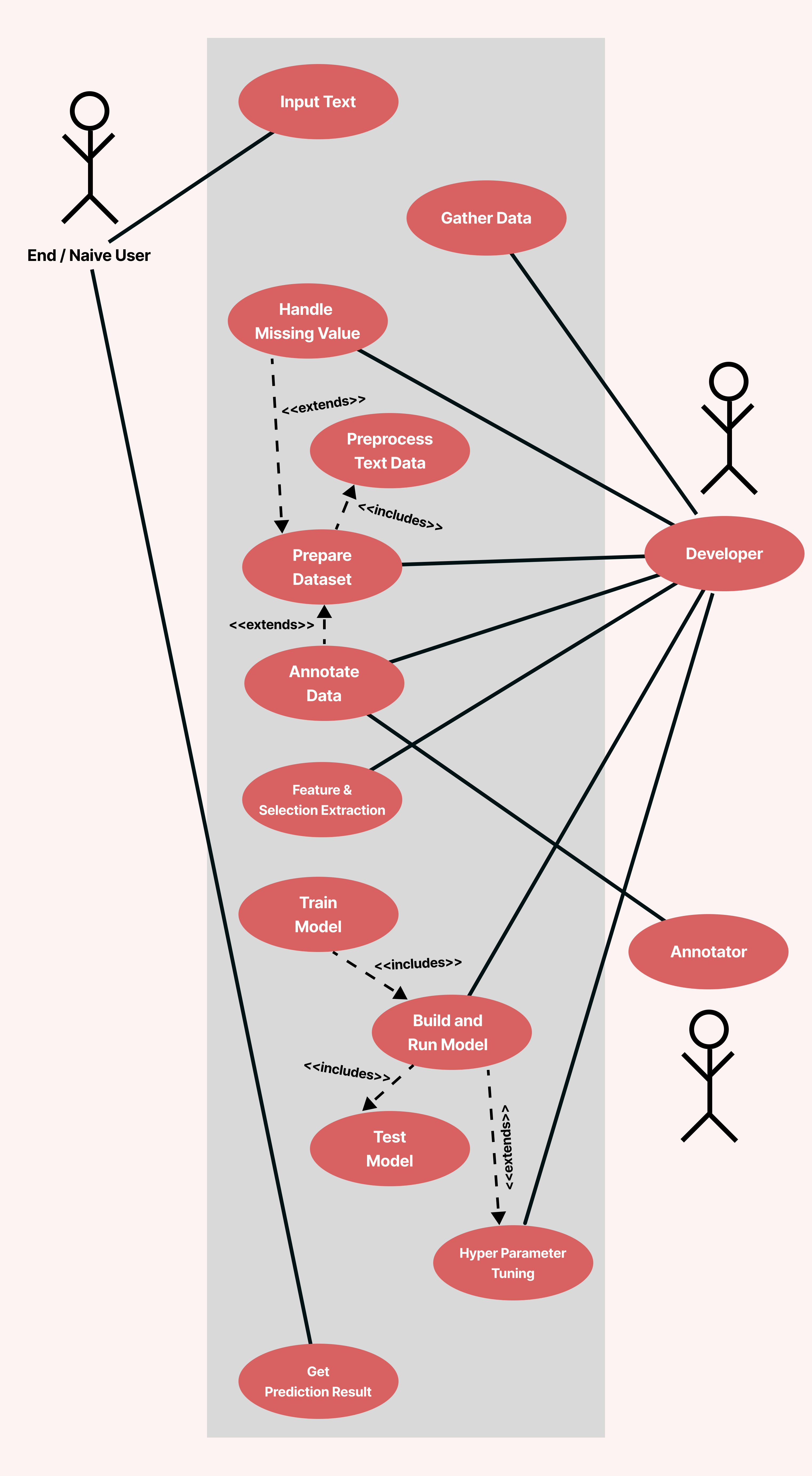


Figure . Use Case diagram

## Data Flow Diagram (DFD)

The context diagram of the profanity detection system implemented in the project is given below:

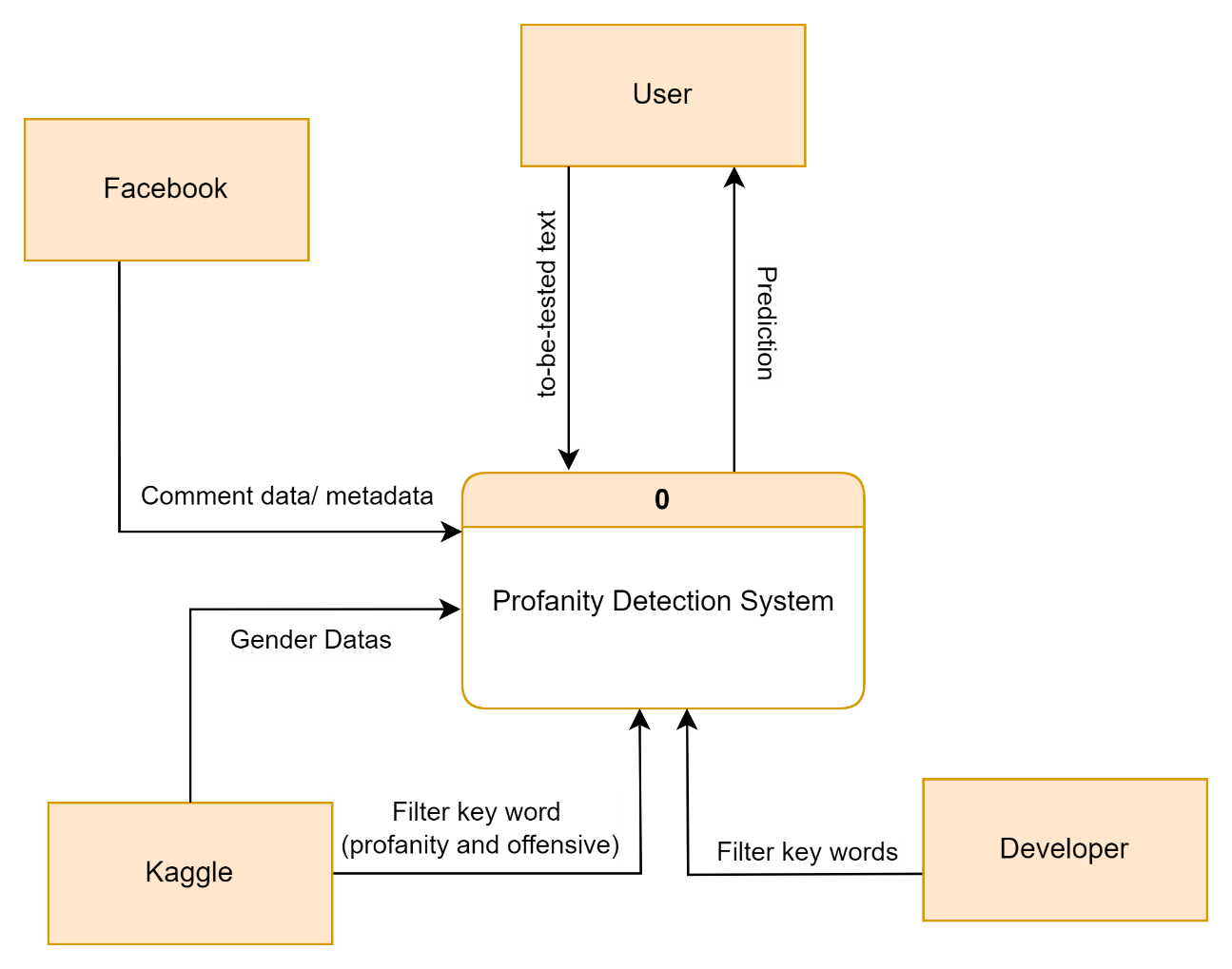


Figure . Context Diagram (DFD)

The above shown context diagram is further broken into Level 0 diagram and Level 0 is further broken down into Level 1 diagram. Both Level 0 and Level 1 diagram are shown below:

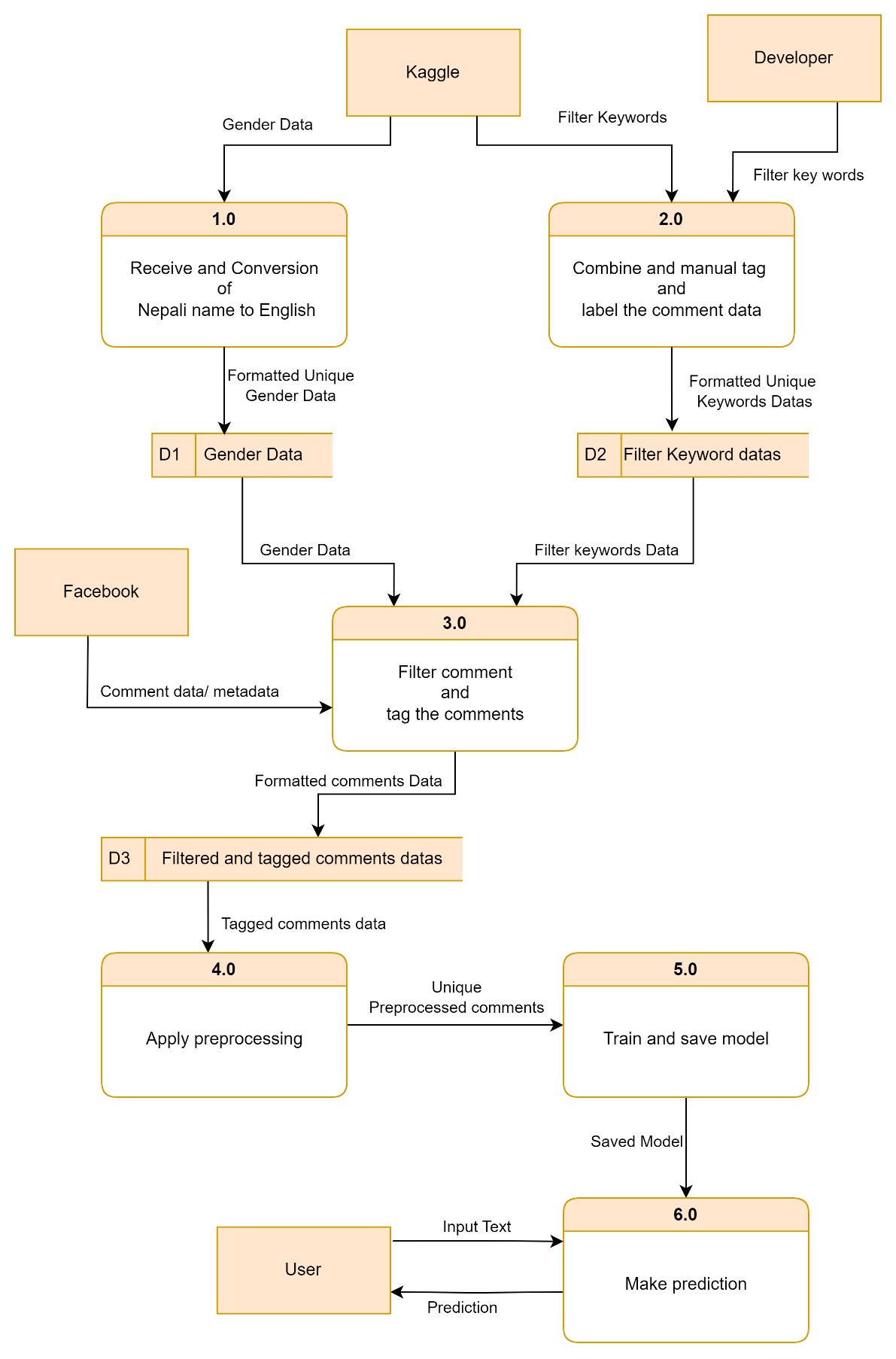


Figure . Level 0 Diagram

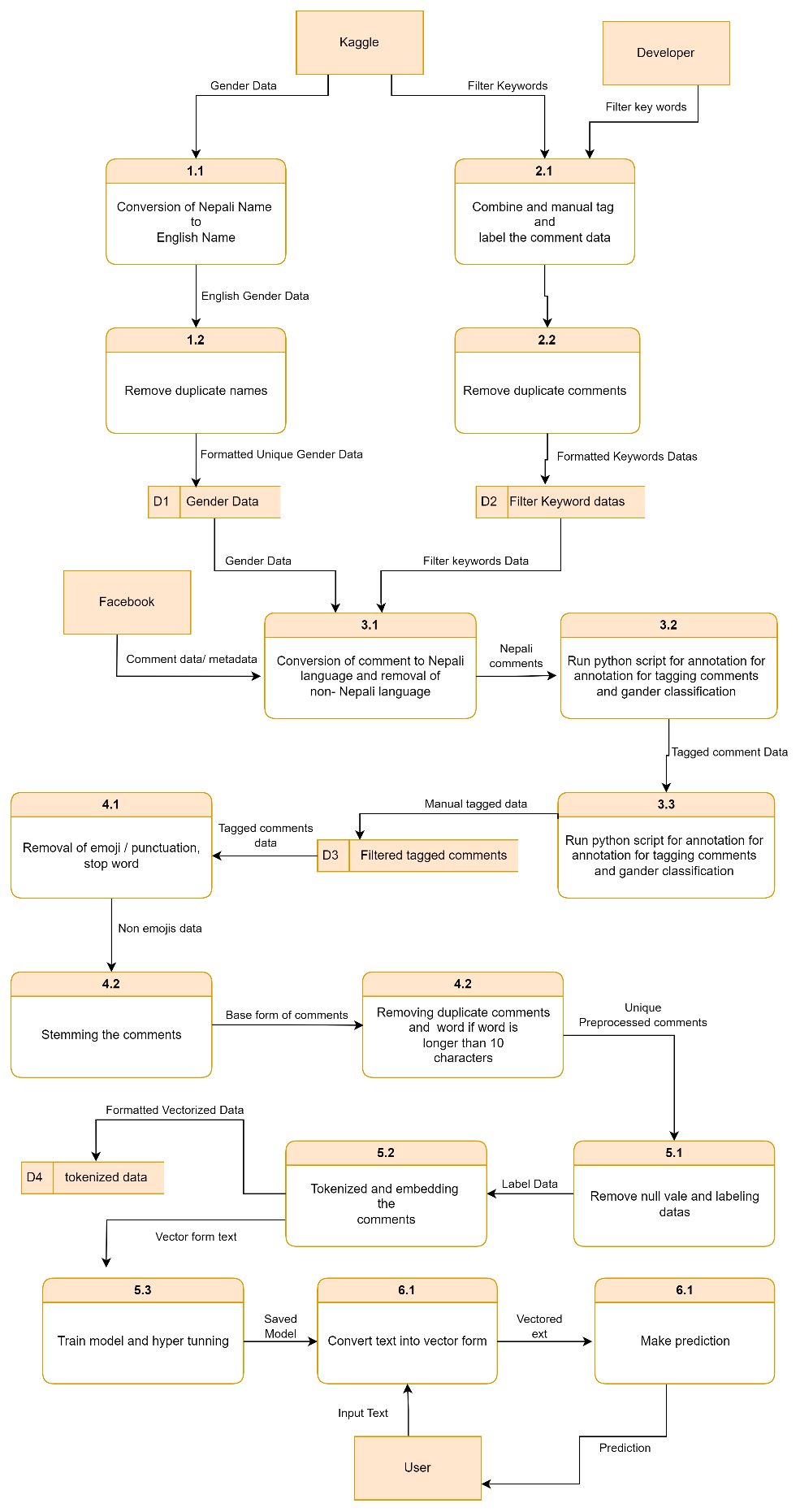


Figure . Level 1 Diagram

## Decision Tree and Diagram

The decision tree showing the decision made during the preprocessing phase are shown below:

A diagram of a flowchart

Description automatically generated

Figure . Decision Tree for preprocessing

The decision regarding the annotation of the text using the regular expressions and gender detection are described in table below:

Table ‑ Decision Table for annotation and labeling

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Conditions** | **Rule 1** | **Rule 2** | **Rule 3** | **Rule 4** | **Rule 5** | **Rule 6** |
| Comment has profane word | No | No | Yes | Yes | No | No |
| Comment has only offensive word | No | No | Yes | Yes | Yes | Yes |
| Is gender of author male? | No | Yes | No | Yes | No | Yes |
| **Actions** |  |  |  |  |  |  |
| Set Comment Label 0 | **X** | **X** |  |  |  |  |
| Set Comment Label 1 |  |  | **X** | **X** |  |  |
| Set Comment Label 2 |  |  |  |  | **X** | **X** |
| Set Gender Label 0 | **X** |  | **X** |  | **X** |  |
| Set Gender Label 1 |  | **X** |  | **X** |  | **X** |

## Nature of Data

In our study, we conducted an extensive analysis of 16,147 comments collected from various social media platforms including Facebook, Twitter, and YouTube. The primary objective was to understand the prevalence and nature of offensive and profane language within these online interactions.

Before proceeding with the analysis, we identified and eliminated 65 comments that were completely devoid of any text. This ensured the integrity of our dataset.

Our analysis revealed the presence of 3,564 instances of profanity and 5,059 instances of offensive language within the dataset. Among these instances, we identified 1,577 unique offensive words and 870 unique profane words. Interestingly, the data indicates a higher frequency of offensive language (5,059 instances) compared to profanity (3,564 instances) in online discourse.

Additionally, our dataset consists of 9,714 male users, 4469 unknown gender (which we failed to detect) and 1,962 female users. This gender distribution provides valuable demographic insights that can further inform our understanding of online behavior and language usage patterns. From this dataset, we can clear view that number of male users commenting offensive and profane comment is large in number.

## Sequence Diagram

The sequence diagram of the overall working of the project is given below:

A screenshot of a computer

Description automatically generated

Figure . Sequence diagram

## State Diagram

The change in the state in the project is shown by the state diagram below:

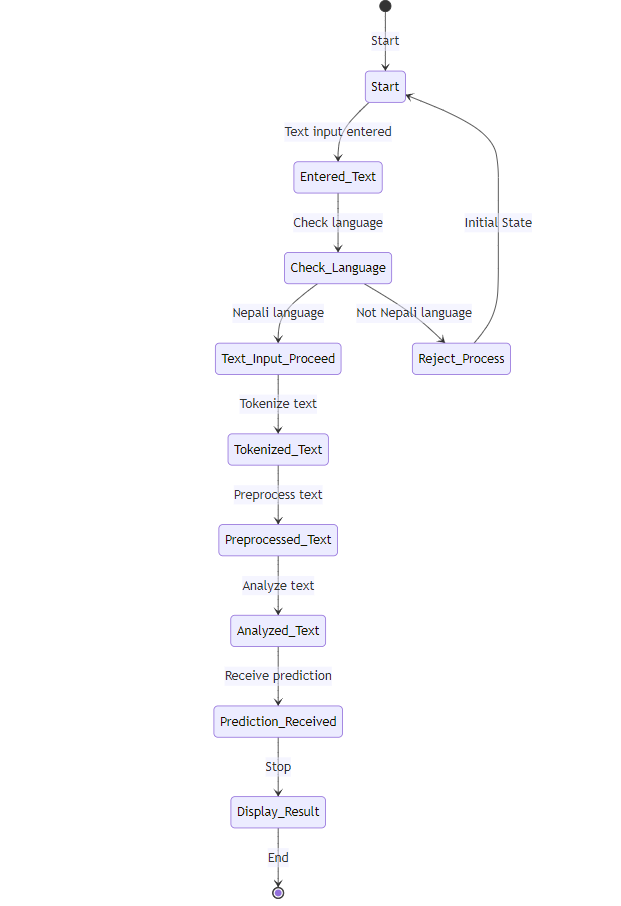


Figure . State Diagram

## System Requirement Specifications

This section specifies the requirements of the developed system. It may include software specifications and hardware specifications.

### Software Specifications

* 64-bit Windows 10 and higher,
* MacOS 10.14 or higher.
* Any 64-bit Linux distribution.

### Hardware Specifications

* Simple Laptop
* Computer with 2.4 GHz processor
* minimum 4 GB RAM
* 100 GB storage.

1. Development Tools Specifications

## Colab

Google Colab (Collaboratory) is a web platform that lets users write and execute Python code in a Jupyter notebook surroundings. It is especially popular for statistical evaluation, and educational purposes because of its ease of use and the reality that it affords unfastened access to computing assets, consisting of GPUs and TPUs. Collab supports collaboration by permitting more than one user to paint at the identical pocketbook concurrently, making it ideal for group initiatives and getting to know environments.

## Kaggle

Kaggle is a platform for data science competitions, datasets, and collaboration. It offers a rich environment for records scientists and gadget getting to know fans to practice, compete, and share their work. Kaggle hosts competitions wherein customers can solve actual world issues using machine getting to know and information analysis, regularly with vast prizes. Additionally, it offers a giant repository of public datasets and network-pushed surroundings for sharing code, insights, and tutorials. Kaggle Kernels (now referred to as Notebooks) allow users to write down and run code within the cloud without having their very own hardware.

## Apify

Apify is a web scraping and automation platform that permits customers to extract statistics from web sites, automate internet workflows, and construct web scrapers the use of JavaScript or a visible interface. The platform offers a marketplace in which users can discover and install pre-constructed scrapers and automation gear, as well as a set of APIs for integrating scraping talents into programs.

## ML tools

* **Python:** Python is a high-level, interpreted programming language known for its readability and simplicity. It supports object-oriented, imperative, and functional programming paradigms, making it versatile. Python's extensive standard library and large community contribute to its popularity for diverse applications.
* **Pandas:** Pandas are a data manipulation and analysis library for Python. It provides data structures like Series (1D labeled array) and Data Frame (2D table with labeled axes) that are built on top of NumPy arrays. Pandas simplify data manipulation tasks, such as cleaning, filtering, grouping, and merging datasets, making it a crucial tool for data wrangling in data science projects.
* **NumPy:** NumPy is a powerful numerical computing library for Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy is fundamental for scientific computing in Python and serves as the foundation for many other libraries in the data science and machine learning ecosystems.
* **NLTK (Natural Language Toolkit):** NLTK is a powerful library in Python designed for working with human language data, primarily in the field of natural language processing (NLP). It provides tools and resources for tasks such as tokenization, stemming, lemmatization, part-of-speech tagging, and named entity recognition, among others. NLTK is widely used for text processing and analysis, making it valuable in applications related to linguistics, information retrieval, and machine learning.
* **Scikit-Learn:** Scikit-learn is a Python package designed to facilitate the use of machine learning and AI algorithms.
* **Matplotlib:** Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.
* **TensorFlow:** TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.
* **Keras:** Keras is an open-source library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.

1. Discussion on the achievements

In our study, we've gained valuable insights into the prevalence of offensive language and profanity in social media interactions. By analyzing a significant dataset of 16,147 comments from platforms like Facebook, Twitter, and YouTube, we've quantified the extent of objectionable language in online discussions. Our findings highlight a concerning trend: offensive language is pervasive, with 5,059 instances detected, compared to 3,564 instances of profanity. We contributed this data to Nepali dataset of profanity and offensive words. Also, we have a system that detects whether a given word is profanity or not.

## Challenges

* **Data Scraping and Quality Control**: Challenges arose during the scraping of YouTube comments from the YouTube API, with many comments being meaningless, spam, or unsuitable for sentiment analysis. This affected the quality and relevance of the data collected, posing initial hurdles in the project.
* **Data Preprocessing Complexity**: Preprocessing the collected data presented various challenges, particularly in handling emojis, mixed-language comments, and non-standard text formats. These complexities impacted the accuracy and reliability of the data used for training and analysis.
* **Model Training and Generalization**: Training models proved difficult due to low accuracy and overfitting issues. Models initially struggled to capture semantic relations between words and generalize to new data, affecting their performance and applicability.
* **Class Imbalance and Data Integrity**: Class imbalance within the datasets affected model performance and robustness, necessitating strategies to address class imbalance while maintaining data integrity and size. Balancing the data reduced its size, further impacting model performance.

## Features:

* Gender Detection on basis of comment
* Profanity and offensive Detection on Nepali language
* Presence of profanity detection on social media (Facebook)
* Contribution to Nepali Profanity dataset

1. Conclusion and Recommendation

In conclusion, this project has successfully proposed and developed a profanity detection system tailored for the Nepali language, addressing a significant gap in the existing language resource landscape. By leveraging recurrent neural networks, specifically a bidirectional LSTM model, the project demonstrates the feasibility and effectiveness of machine learning algorithms in identifying and filtering profane and offensive content in Nepali text.

Through the creation of a comprehensive dataset of Nepali text samples, we have laid the groundwork for robust training and evaluation of our models. Experimentation with different models and embedding layers has allowed us to identify the most suitable configurations for achieving high accuracy in profanity detection. This approach not only contributes to the practical application of reducing offensive language on social media platforms but also enhances the body of research in the field of low-resource language processing.

Overall, this project paves the way for future research and development in low-resource language processing, encouraging further innovation and refinement of profanity detection systems.

## Limitations

Some of the limitations of our project are:

* Multiple models cannot be evaluated concurrently.
* The enormous number of comments cannot be extracted from YouTube.
* The models lack adequate generalization to understand nuances better.
* Lack of personalization for user.

## Future Enhancement

* Improvement of Offensiveness Detection Model using advanced language models like transformers and expanded dataset for generalization.
* Expansion of the Dataset to incorporate the reginal slangs and increase the capabilities of the model.
* Integration of the model with web apps for practical usage of the model.

# References

*Dhanya, L. K. (2021, June 16). Hate speech Detection in Asian Languages:A Survey. IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9484922Oyashi. (2023, November 30). oya163/nepali-sentiment-analysis. GitHub.* [*https://github.com/oya163/nepali-sentiment-analysis*](https://github.com/oya163/nepali-sentiment-analysis%20)

*Mehant Kammakomati, et al. “Comparison of Machine Learning Algorithms for Hate and Offensive Speech Detection.” Lecture Notes on Data Engineering and Communications Technologies, 1 Jan. 2022, pp. 873–881,* [*https://doi.org/10.1007/978-981-16-9605-3\_61*](https://doi.org/10.1007/978-981-16-9605-3_61)*. Accessed 15 Mar. 2024.*

*Singh, Oyesh Mann; Timilsina, Sandesh; Bal, Bal Krishna; Joshi, Anupam; Aspect Based Abusive Sentiment Detection in Nepali Social Media Texts; UMBC HPCF;* [*http://hpcf-files.umbc.edu/research/papers/NepSA\_ASONAM.pdf*](http://hpcf-files.umbc.edu/research/papers/NepSA_ASONAM.pdf%20)

*Sushil79g. (n.d.). GitHub - sushil79g/Nepali\_nlp: A python based library for NLP in Nepali language. GitHub.* [*https://github.com/sushil79g/Nepali\_nlp*](https://github.com/sushil79g/Nepali_nlp)

Tamrakar, S., Bal, B. K., & Thapa, R. B. (2020). Aspect based sentiment analysis of Nepali text using support Vector machine and naive bayes. *Technical Journal*, *2*(1), 22–29. <https://doi.org/10.3126/tj.v2i1.32824>

# APPENDIX

## d

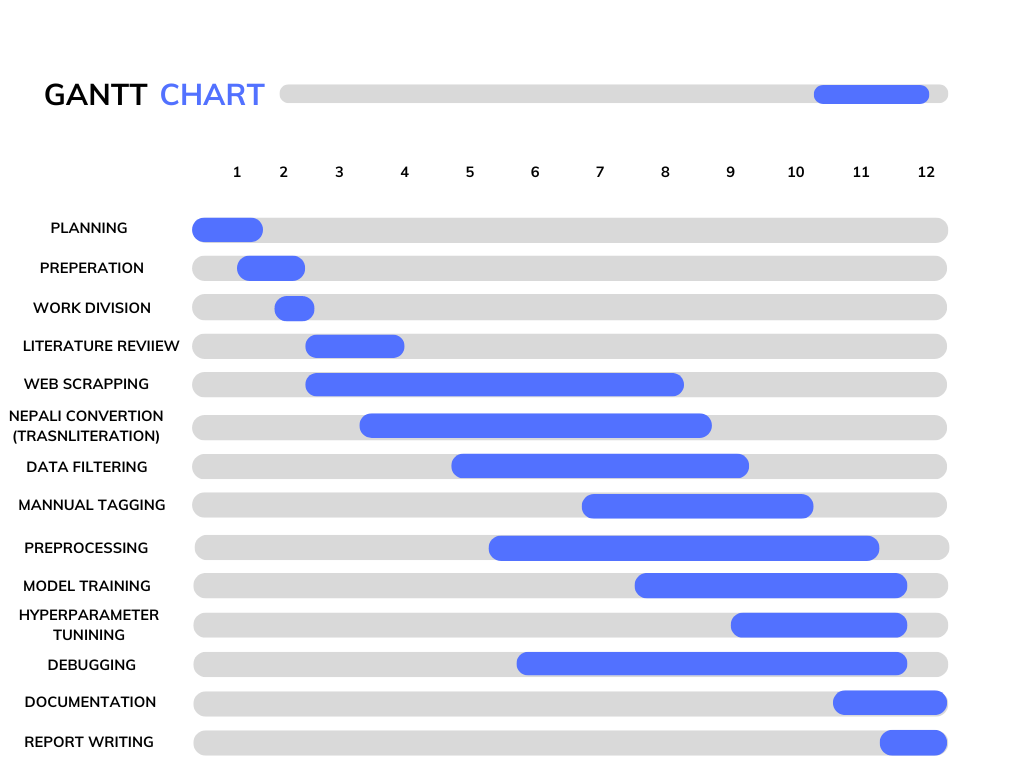


Figure . Gannt Chart

A screenshot of a computer

Description automatically generated

Figure . Output-1 from all models

A screen shot of a computer

Description automatically generated

Figure . Output-2 from all models