KANTIPUR ENGINEERING COLLEGE

(Affiliated to Tribhuvan University)

Dhapakhel, Lalitpur

[Subject Code: CT654]

A MINOR PROJECT FINAL REPORT ON

THYROID CLASSIFICATION WITH L1 REGULARIZATION FOR EFFECTIVE FEATURE SELECTION

Submitted by:

Aadarsha Regmi[KAN077BCT001]

Angel Tamang [KAN077BCT012]

Anil Bhatta [KAN077BCT013]

Manish Karki [KAN077BCT047]

A MINOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

Department of Computer and Electronics Engineering

February, 2023

THYROID CLASSIFICATION WITH L1 REGULARIZATION FOR EFFECTIVE FEATURE SELECTION

Submitted by:

Aadarsha Regmi[KAN077BCT001]

Angel Tamang [KAN077BCT012]

Anil Bhatta [KAN077BCT013]

Manish Karki [KAN077BCT047]

A MINOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

Department of Computer and Electronics Engineering

Kantipur Engineering College

Dhapakhel, Lalitpur

February, 2023

ABSTRACT

In today's age, education is the most important way of achieving success. There are many to many doubts and queries so solving these doubts lead to proper implementation of education. This project completely deals with the system that is to be built to fulfill the importance of the project work. This project would be developed with the complete approach of the software engineering from start to end. The project is all about making use of the internet to connect groups of students together for exchange of knowledge. In other words, it reduces the extra effort of students to find help in academic as well as non-academic problems.

The project is on a Web Application system developed using VS code studio. The major focus of the project is on providing a space for students to place their queries and get answers from others, answer other queries, rate the questions and answers they find useful. The users can also search the questions they are curious about.

Keywords – Web Application

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

DTC	Decision Tree Classification
EDA	Exploratory Data Analysis
ML	Machine Learning
ROC	Receiver Operating Characteristics
PR	Precision-Recall

CHAPTER 1
INTRODUCTION

1.1 Background

Thyroid disease is a significant global health concern, affecting millions of people worldwide. The thyroid gland, a vital organ in our body, plays a crucial role in metabolism, growth, and development. It produces two main hormones, thyroxine (T4) and triiodothyronine (T3), which are essential for the body's metabolic processes. The production of these hormones is regulated by thyroid-stimulating hormone (TSH), which is released by the pituitary gland. An imbalance in these hormones can lead to thyroid diseases [**?**]. Thyroid diseases can be broadly classified into conditions that affect the structure of the gland, such as goiter and thyroid nodules, and those that affect the function of the gland, such as hypothyroidism, hyperthyroidism, and thyroiditis. Hypothyroidism is a condition where the thyroid gland does not produce enough thyroid hormones, leading to symptoms like fatigue, weight gain, and depression [**?**]. On the other hand, hyperthyroidism is a condition where the thyroid gland produces too much thyroid hormones, leading to symptoms like rapid heart rate, weight loss, and anxiety. Thyroiditis is an inflammation of the thyroid gland, which can cause either hyperthyroidism or hypothyroidism. More serious conditions include thyroid cancer and autoimmune thyroid diseases, such as Graves' disease and Hashimoto's thyroiditis [**?**]. In the context of Nepal, thyroid disorders are prevalent, with a study showing that the prevalence of thyroid dysfunction was 17.42% among the population of the western region of Nepal [4]. However, this project will not be using a dataset specific to Nepal.

Machine learning, a subset of artificial intelligence, has shown great promise in the field of healthcare, particularly in disease diagnosis. It has the potential to improve the accuracy and speed of diagnosis, thereby enhancing patient outcomes. In recent years, there has been a growing interest in applying machine learning algorithms for thyroid disease classification. Among various machine learning algorithms, decision trees have gained popularity due to their interpretability and ease of use. They predict the target variable by learning simple decision rules inferred from the data features.

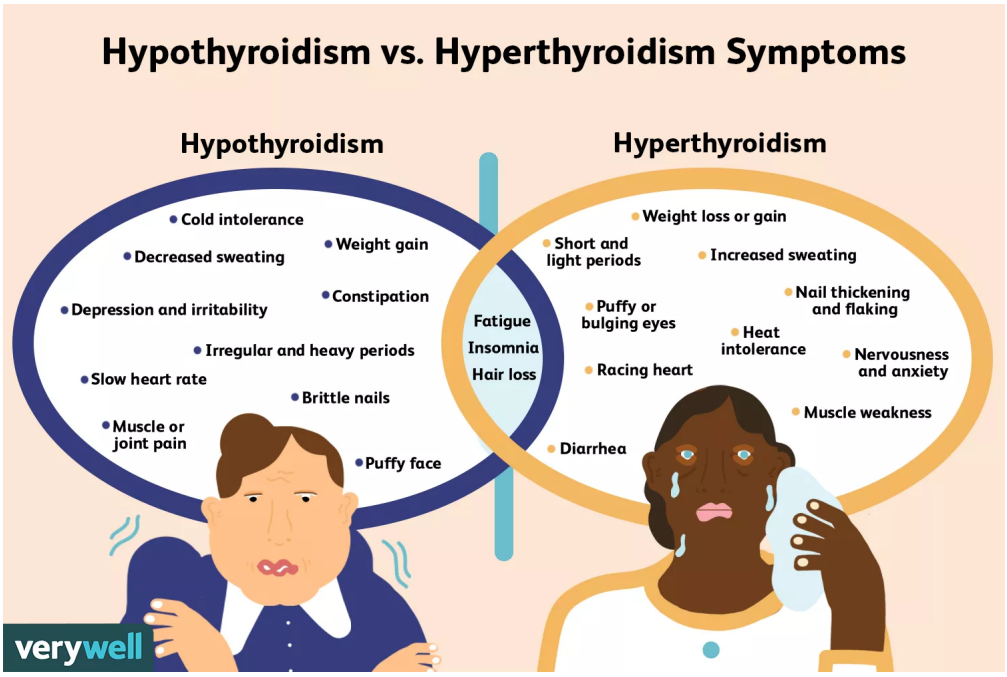

Figure 1.1: Symptoms

1.2 Problem Statement

Thyroid diseases are a significant health concern due to their prevalence and potential impact on quality of life. Accurate and timely diagnosis is crucial for effective treatment and management. However, traditional diagnostic methods can be time-consuming, costly, and may not always provide accurate results. The interpretation of thyroid function tests can be complex due to various influencing factors such as age, gender, and comorbid conditions. Machine learning algorithms, such as decision trees, offer a promising solution to these challenges. They have the potential to improve the accuracy and speed of diagnosis by learning from patterns in the data. However, their application in thyroid disease classification is still an emerging field and needs further research. This project aims to address these challenges by developing a decision tree-based model for thyroid disease classification using a dataset that is not specific to any particular region.

1.3 Objectives

The primary objective of this project is to develop a decision tree-based machine learning model for the classification of thyroid diseases. The specific objectives are as fol-

lows:

I To develop a decision tree-based model for thyroid disease classification.

II To compare the performance of the decision tree model with another machine learning model.

1.4 Application Scope

The application of machine learning, specifically decision tree algorithms, in the classification of thyroid diseases has the potential to revolutionize the way these diseases are diagnosed and managed. The model developed in this project could serve as a valuable tool for healthcare professionals, aiding in the accurate and timely diagnosis of thyroid diseases. The scope of this project is broad, as the dataset used for model development is not specific to any particular region. This enhances the applicability and relevance of the project findings, potentially benefiting a wider population. The project also aims to contribute to the growing body of research on the application of machine learning in healthcare, particularly in the context of thyroid disease classification.

1.5 Features

The key features of this project include:

- Use of decision tree algorithms for thyroid disease classification.
- Comprehensive data analysis and preprocessing.
- Development and rigorous evaluation of the decision tree model.
- Performance evaluation of the developed model using appropriate metrics.
- Comparison of the decision tree model with other machine learning models, if applicable.
- Interpretation of results to identify key features for thyroid disease classification.
- Broad applicability due to the use of a non-region-specific dataset.

1.6 System Requirements

This project needs certain hardware and software requirements in order to be developed and run. These requirements are discussed below:

1.6.1 Development Requirements

Hardware Requirement(Minimum)

- PC with minimum 4 GB RAM and fifth generation i5 processor.

Software Requirement

- HTML, CSS, JavaScript, Express ,MongoDB
- Browser : Chromium based browser

1.6.2 Deployment Requirements

Hardware Requirement(Minimum)

- PC or Mobile with internet connection

Software Requirement

- Operating System : Windows 8 / Ubuntu/ macOS Big Sur
- Browser : Chromium based browser

1.7 Feasibility Study

1.7.1 Economic Feasibility

This system is economically feasible which consists of a laptop or a personal computer without any expense of other items. Nowadays, almost every house has access to the internet, so our system is economically feasible.

1.7.2 Operational Feasibility

For the operation of the system, the person doesn't need to be an expert in using a computer. Someone with minimum knowledge about computers and technology can also benefit from the system. There is no requirement for huge and expensive hardware.

1.7.3 Schedule Feasibility

Since the development team has the capacity and basic understanding of the project the project was completed in 9 month time period.

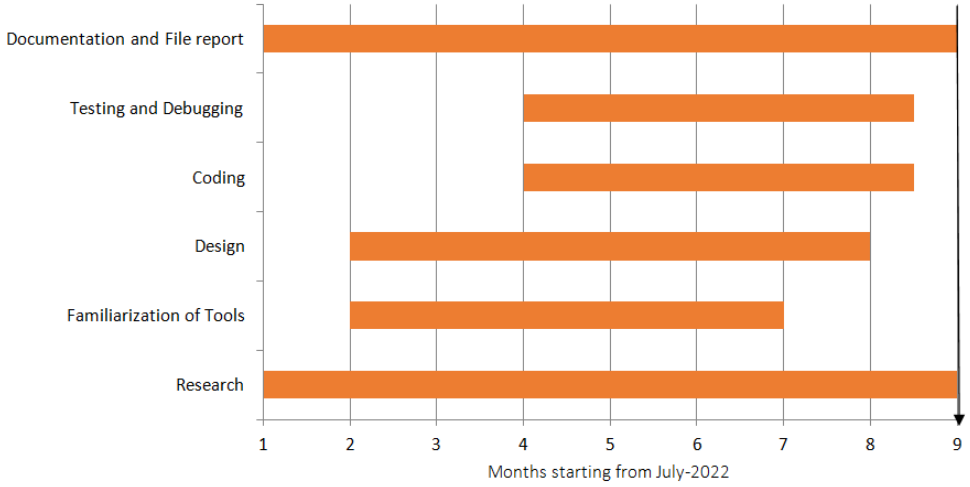

Figure 1.2: Gantt Chart

CHAPTER 2
LITREATURE REVIEW

2.1 Related Research

According to the paper "Thyroid function", Iodine is most important as a component of the hormones, thyroxine and 3,3,5-triiodothyronine (T3). The recommended daily requirement is 150-200μg. Since iodine is a crucial constituent of thyroid hormones, it is not surprising that thyroid dysfunction is very common in geographical areas of iodine deficiency. However, even when this trace element is present in adequate supply, thyroid disease is present in 3-5% of the population. Furthermore, the regulated supply of thyroid hormone to specific tissues is crucial during fetal development [**?**].

Paper "Thyroid Disease Classification Using Machine Learning Algorithms" aims to classify thyroid conditions because medical reports show serious imbalances in thyroid diseases. The data was applied to a range of machine learning algorithms (Decision Tree, SVM, Random Forest, Naive Bayes, Logistic Regression, Linear Discriminant Analysis, k-Nearest neighbors, Multi-Layer Perceptron). On using all the attributes the results were as: Decision Tree 98.4 accuracy, SVM 92.27 accuracy, Random Forest 98.93 accuracy, Naive Bayes 81.33 accuracy, Logistic Regression 91.47 accuracy, Linear Discriminant Analysis 83.2 accuracy, KNeighborsClassifier 90.93 accuracy and MLP (NN) 97.6 accuracy. In the second step, 3 traits were removed, the deleted attributes were query_thyroxine, query_hypothyorid & query_hyperthyroid. The algorithms' performance after this were noted as: Naive Bayes algorithm has a high accuracy of 90.67 after the three traits have been omitted, the SVM algorithm, the logistic regression algorithm and the KNeighbours Classifier algorithm have increased slightly and reduced the accuracy of the other algorithms [**?**].

Crucially, in evaluating these classifiers, the choice of metrics, particularly in the context of imbalanced datasets like those in thyroid disease classification, becomes paramount. The article "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets" emphasizes this. It points out

that PRC plots are more informative than ROC plots in such situations, as they focus on the minority class, providing a clearer picture of the classifier's performance in identifying less prevalent but clinically significant cases. This insight is vital for our project's methodology and aligns with our objective to enhance diagnostic accuracy in thyroid disease [**?**].

CHAPTER 3
METHODOLOGY

3.1 Working Mechanism

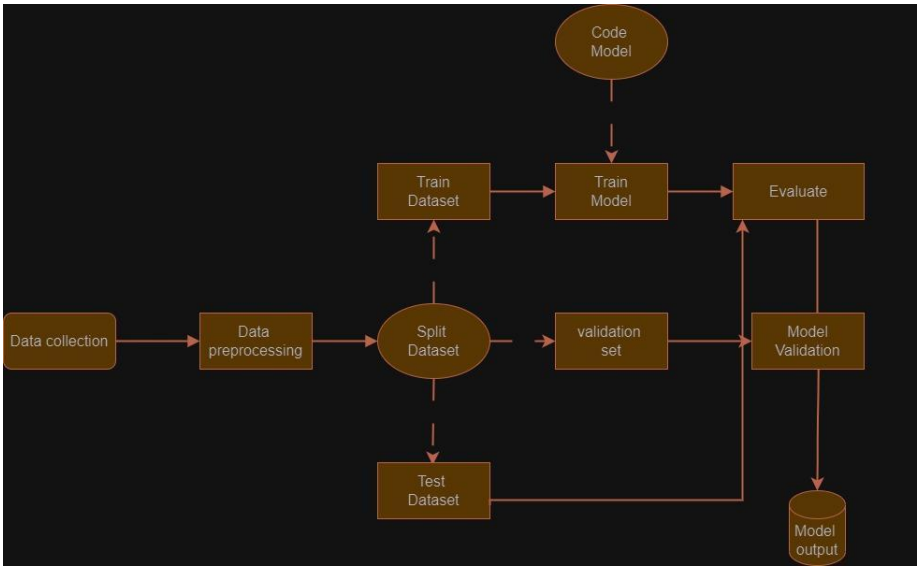

Figure 3.1: Block diagram of Thyroid Classification

3.1.1 Data Collection

Machine learning algorithms are used in the rapid and early diagnosis of thyroid diseases and other diseases, as they are now in a significant position in the health field and help us in diagnosing and classifying diseases and for this reason we have collected our dataset that was found on the Kaggle Website [**?**].The data that we have used in our study is a set of data taken from external hospitals and laboratories specialized in analyzing and diagnosing the thyroid diseases. In this dataset, we have found 9172 observations along with 31 attributes.

3.1.2 Data Preprocessing

The process of pre-processing the data is very important and it is a major step in data mining, as it has a good effect on the data, as the pre-processing process is used to reveal the data through analyzing the data and discovering the lost data, as it examines the data with great care. The pre-processing process includes cleaning the data, preparing the

data, etc. for data processing we use the technique called EDA which is used to analyze the data and discover trends, patterns, or check assumptions in data with the help of statistical summaries and graphical representations.

3.1.3 Data Machine Learning Technique

The key aim of using machine learning algorithms is to differentiate between three forms of thyroid disease. The first is hyperthyroidism, the second is hypothyroidism, and the third is stable patients who do not have any thyroid issues. In order to facilitate this, DTC will be implemented.

Decision Tree

A decision tree is one of the most powerful tools of supervised learning algorithms used for both classification and regression tasks. It builds a flowchart-like tree structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. It is constructed by recursively splitting the training data into subsets based on the values of the attributes until a stopping criterion is met, such as the maximum depth of the tree or the minimum number of samples required to split a node[**?**].

- Impurity: A measurement of the target variable's homogeneity in a subset of data. It refers to the degree of randomness or uncertainty in a set of examples. The Gini index and entropy are two commonly used impurity measurements in decision trees for classification tasks.

- Information Gain: Information gain is a measure of the reduction in impurity achieved by splitting a dataset on a particular feature in a decision tree. The splitting criterion is determined by the feature that offers the greatest information gain, It is used to determine the most informative feature to split on at each node of the tree, with the goal of creating pure subsets
- Entropy:
  ○ Entropy is the measure of the degree of randomness or uncertainty in the dataset.

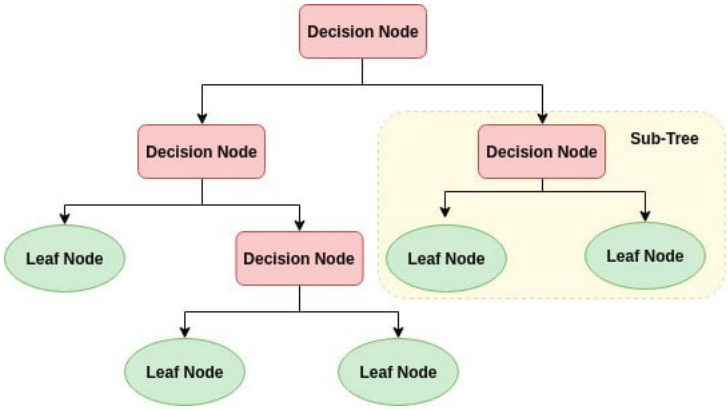

Figure 3.2: Decision Tree

In the case of classifications, it measures the randomness based on the distribution of class labels in the dataset.

◦ Entropy = Σ - (P_i * $\log_2 P_i$)

Where, P_i = Probability of Class

◦ Information Gain= E(Parent node) - Σ(W_i * E(child node))

Where, E = Entropy

W_i = Size of Child / Size of Parent

- Gini Impurity or index:

  ◦ Gini Impurity is a score that evaluates how accurate a split is among the classified groups. The Gini Impurity evaluates a score in the range between 0 and 1, where 0 is when all observations belong to one class, and 1 is a random distribution of the elements within classes. In this case, we want to have a Gini index score as low as possible. Gini Index is the evaluation metric we shall use to evaluate our Decision Tree Model.

  ◦ Gini index = 1 - Σ P_i^2

  ◦ Information Gain= G(Parent node)- Σ(W_i * G(child node))

  Where, G = Gini index

  W_i = Size of Child / Size of Parent

3.1.4 Split Dataset

An important phase in the model development process is splitting the facts for system mastery models. It entails breaking up the available dataset into several subgroups for

training, validation, and version testing [**?**]. Here are a few typical methods for dividing data that we used in our project are listed below:

Train Dataset Split

The dataset is split into a training set and an experimentation set. The checking out set is used to evaluate the model's overall performance while the education set is used to teach the model. The typical cut up is 20–30% for checking out and 70–80% for training, but this might change based on the size of the dataset and the specific use case [**?**].

Validation Set Split

Three subsets of the dataset are created: a learning set, a validation set, and a testing set [**?**]. The training set is used to train the version, the validation set is used to fine-tune hyperparameters and confirm the version's overall performance throughout training, and the testing set is used to assess the performance of the most recent version.

Test Set Split

The testing set provides an objective assessment of the generalizability of our model. We assess our model's performance on the testing set once it has been fully trained and tuned using the training and validation sets. This phase allows you to accurately predict how well our model will perform on new thyroid patient samples that it hasn't seen before, and this helps in finding the thyroid patients.

Train Model

Training a machine learning model involves the process of feeding data to the model, adjusting its internal parameters, and optimizing its performance. In the context of thyroid patient classification, the model will learn from the input features and their corresponding labels during this process. The specific training algorithm and duration depend on the chosen model and complexity of the data.

Model validation

Model validation is the task of evaluating whether a chosen statistical model is appropriate or not. The significance of a correct medical diagnosis necessitates specific considerations when validating a machine learning model for thyroid illness categorization [**?**]. Models can be validated by comparing output to independent field or experimental data sets that align with the simulated scenario. Validating a machine learning model for thyroid disease classification requires specific considerations due to the importance of accurate medical diagnosis [**?**].

Model Output

The model output refers to the predictions generated by a trained machine learning model when given input data (features). In the context of thyroid disease classification, the model output will indicate the predicted probability or class label for each patient, indicating whether the patient is likely to have thyroid disease or not. In our project we use streamlit which allows us to display descriptive text and model outputs, visualize data and model performance and modify model inputs through the UI using sidebars [**?**]. For saving the model that we have trained in our project we use the module called pickle, which can be defined as a module in Python used for serializing and de-serializing Python objects. This converts Python objects like lists, dictionaries, etc. into byte streams (zeros & ones) [**?**].

3.2 Development Model

Since our project demands few requirements and is fairly simple, the Incremental model is opted, which is a mixture of waterfall and iterative development approach. As the name suggests, the major objective is focused as the first increment of the project. The activities in this model are completed iteratively and each outcome acts as an input for the next phase, thus increments are developed in such a way.

3.2.1 Requirement Specification

The final product can be divided into two parts: the frontend and the backend. The frontend shall be a simple page where users can input a few specific hormones (assuming the user has measured these hormones prior to using the product), and obtain their thyroid condition. Whereas, the backend consists of training our DTC model; checking its accuracy and comparing its result with a fine tuned trained ML model like XGBoost Multi-class Classification.

3.2.2 Development and Implementation

The project shall have an architecture close to any other ML projects, i.e. EDA –¿ preprocessing -¿ model training -¿ save model. The product will be written using python programming language and jupyter notebook. The interface for the use of the product will be rather simple as the only requirement is classification of thyroid, which can be achieved through the use of the streamlit framework.

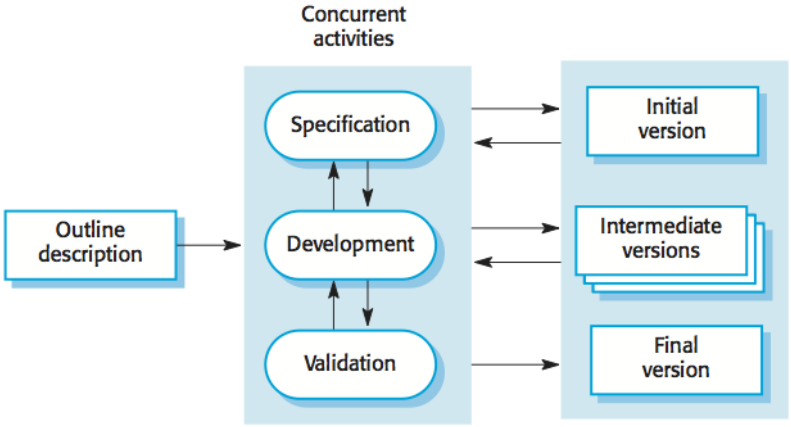

Figure 3.3: Incremental Development Model

3.2.3 Verification and Validation

In order to approve that the product meets its defined objectives, verification and validation will be exercised. This involves Meetings and Code reviews. The DTC model will be verified through the comparison with XGBoost Multi-class Classification. However, considering the imbalance of data in our thyroid dataset there are chances our model

overfitting. Thus, we will implement balancing methods to handle the imbalance data. The code itself can be verified with inspections and walkthroughs among the group and even by external parties. Tools like selenium can also be implemented to test the code. Finally, the product can be verified by testing its classification prediction on some known data.

3.2.4 Increments

As discussed earlier the project will go through several increments. The first increment solely focuses on the major requirement, developing a decision tree model for the required classification. The further increments are focused on comparison with another well developed, finely tuned model, tuning of hyperparameters, developing a front-end for the product.

CHAPTER 4
RESULT AND DISCUSSION

4.1 Result

We have completed the design and development of the system along with the required output of the project. The project currently allows user to register and log-in into the system. User can post queries, answer, vote, follow other users and search queries. Trending section shows most relevant post using half life decay algorithm.

4.2 Discussion

Kantipur Online Portal is a web application developed with an objective to provide an environment for students to ask and share queries with each other.This application was built using Express framework. Likewise,EJS bootstrap was used for user interface layout along with HTML and css.User input was passed to the system, in the form of queries, answers and votes.

Finally, we implemented Half-Life Decay formula to generate trending section. At first, we used Half-Life Decay to generate weight of individual votes and sum of the votes was the total weight of a query. Then queries were sorted on descending order of total weight.

After the completion of the project, we analyzed the result of our system to check if our system performs the way we expected or not. Firstly, the UI of the system was interactive for better user experience. Then we analyzed the if the correct result was returned and verified it. After that we analyzed the half-life decay algorithm by increasing the h(half life parameter) from 7 to 30, and verified that the weight of a vote was taking longer to decrease.

At the beginning of the project, we only had theoretical idea about database, framework and Half-Life decay algorithms. As the project started to take place, we learned about requirement analysis, and found the requirements of the project. We researched on many papers and found that there are many ways to develop a recommendation system and many different algorithms could be implemented. We learned to implement

this algorithm using JS and also learned the real world application of the algorithm of which we had only theoretical knowledge of.

CHAPTER 5
CONCLUSION AND FUTURE ENHANCEMENTS

5.1 Conclusion

In this project, we were able to build a web application system using Exponential half-life decay and merge sort algorithm. We implemented Half-life decay a to generate a trending section to display the most relevant queries to the user.

5.2 Future Enhancements

- This project can be further extended to include Real-time chat system between users.
- This system could be made more user friendly by adding chat-bot to help user with different problems.
- This project can be further upgraded to support contents like images and videos.

REFERENCES

[1] J. A. Obar and S. S. Wildman, "Social media definition and the governance challenge-an introduction to the special issue," *Obar, JA and Wildman, S.(2015). Social media definition and the governance challenge: An introduction to the special issue. Telecommunications policy*, vol. 39, no. 9, pp. 745–750, 2015.

[2] "What refactoring topics do developers discuss? a large scale empirical study using stack overflow," *IEEE Access*, vol. 10, pp. 56 362–56 374, 2021.

[3] L. Yang and X. Amatriain, "Recommending the world's knowledge: Application of recommender systems at quora," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 389–389.

[4] S. Paira, S. Chandra, and S. S. Alam, "Enhanced merge sort-a new approach to the merging process," *Procedia Computer Science*, vol. 93, pp. 982–987, 2016.

[5] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 287–296.

[6] A. Arampatzis, J. Beney, C. H. Koster, and T. P. van der Weide, "Incrementality, half-life, and threshold optimization for adaptive document filtering." in *TREC*, 2000.

[7] P. Ardagelou and A. Arampatzis, "A half-life decaying model for recommender systems with matrix factorization." in *TDDL/MDQual/Futurity@ TPDL*, 2017.