



ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

Classifying Stack Overflow Questions Quality using SVM

Authors

Muhammad Bello Atiku – 170041076

Naayif Ismail – 170041069

Mohamadou Alhadji – 170041071

Supervisor

Md Jubair Ibna Mostafa

Lecturer

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), OIC

A thesis submitted in partial fulfillment of the requirements for the degree of B. Sc. Engineering
in Computer Science and Engineering
Academic Year: 2020-2021

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
A Subsidiary Organ of the Organization of Islamic Cooperation (OIC)
Dhaka, Bangladesh

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by **Muhammad Bello Atiku**, **Naayif Ismail**, and **Mohamadou Alhadji** under the supervision of **Md Jubair Ibna Mostafa**, Lecturer, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:



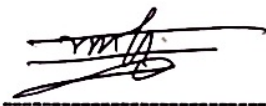
Name: Muhammad Bello Atiku

Student ID – 170041076



Name: Naayif Ismail

Student ID – 170041069



Name: Mohamadou Alhadji

Student ID – 170041071

Approved By

Supervisor:



Md Jubair Ibna Mostafa

Lecturer

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), OIC

Dedication

We dedicate our thesis work to our family. A special feeling of gratitude to our parents. In addition, we express our deep gratitude towards our respected thesis supervisor **Md Jubair Ibna Mostafa**. We also dedicate this thesis to our many friends who have supported us throughout the process. We will always appreciate what they have done.

Acknowledgment

We would like to express our grateful appreciation to **Md Jubair Ibna Mostafa**, Lecturer, Department of Computer Science & Engineering, for being our adviser and mentor. His motivation, suggestions, and insights for this research have been invaluable. Without his support and proper guidance, this research would never have been possible. We are grateful to him.

Abstract

This thesis aims to discover indicators of quality, and to use this knowledge to correctly classify the quality of questions and answers from Stack Overflow. The proliferation of technical questions and answers on Q&A websites such as Stack Overflow means there is more information available than ever. However, the ease of publishing such information also tends to mean the quality varies significantly. The job of moderating Stack Overflow is left to the community. Stack Overflow performs some basic quality analysis, but this is an area where improvement would have many benefits to not only Stack Overflow, but many other domains where the quality of text is important.

List of Tables

Table 2.1.3: 7zip extract command inflates the compressed archive

Table 4.2.2: Schema of the post's dataset

List of Figures

Figure 3.0: categorization using a hyperplane

Figure 3.1: The supervised learning process. [NLTK]

Figure 3.1.1: Perception updating its linear boundary as training examples are added Goodspeed [2015]

Figure 3.2: Methodology

Figure 3.2.4: Decision tree

Figure 3.2.5: Trials

List of Acronyms

SVM	Support Vector Machine
NLP	Natural Language Processing
TF	Term Frequency
IDF	Inverse Document Frequency
SO	Stack-Overflow
RBF	Radial Basis Kernel
HTML	Hypertext Markup Language
XML	Extensible Markup Language
Q&A	Question and Answer

Contents	1
1. Introduction	2
1.1. Overview	2
1.2. Problem Statement	2
1.3. Contribution	2
2. Background Study and Literature Review	3
2.1. Stack overflow Datasets	4
2.1.1 Location of the Dataset	4
2.1.2 The importance of this Dataset	4
2.1.3 Managing the Data	5
3. Proposed Approach	6
3.1. Supervised Learning + Support Vector machine	7
3.2. Methodology	9
3.2.1 Data Representation	10
3.2.2 Preprocessing	12
3.2.3 Training and Testing Strategies	12
3.2.4 Decision Trees	13
3.2.5 TF-IDF + SVM parameters tuning with Optuna	14
4. Experimental Evaluation and Result Analysis	16
4.1. Overview	16
4.2. Experimental Setup	16
4.2.1 Dataset Preparation	17
4.2.2 Dataset Description	17
4.2.3 The Software and Hardware materials	19
4.3 Result Analysis	19
4.3.1 Experimental Results	19
4.3.2 Analysis	19
5. Conclusion and Future Work	20

Chapter 1

Introduction

1.1. Overview

This thesis aims to discover indicators of quality and to use this knowledge to correctly classify the quality of questions and answers from Stack Overflow. Stack Overflow is a popular questions and answers (Q&A) website (a Stack-Exchange Q&A Community) where users exchange knowledge about various topics related to programming. The study shows that about 73% of the questions on Stack Overflow are answered and almost 27 % of questions are unanswered on Stack Overflow due to the quality of the Questions.

1.2 Problem Statement

The quality of the content provided by Q&A websites varies and ranges from high quality to low-quality. This low-quality content can also be abusive or dangerously ignorant/misleading in addition to merely being a poor question. This thesis delves into the problem of analyzing and classifying the quality of questions. Through this, it is possible to discover what affects a question's quality, and how the process of quality monitoring can be at least partially automated. Additionally, the techniques should generalize across multiple domains including other Stack Exchange Q&A communities.

1.3 Contribution

A positive quality classification performance indicates that several of the observed variables substantially connect with question quality. In all situations, the findings are statistically significant, indicating a notable difference or trend. Our analysis reveals some intriguing conclusions. The strategies pros and drawbacks are examined, as well as alternatives. This dissertation provides a solid basis for others to build upon. Quality will always be a subjective concept that differs from person to person. The quality classifications should be as accurate as a subject matter expert. This document shows that while quality classification is not perfect, it is a significant step in the right direction. Thus, this thesis has achieved and evaluated its objectives.

Chapter 2

2. Background Study and Literature Review

Stack Overflow

Programming and information technology questions and answers are exchanged on Stack Overflow (a Stack Exchange Q&A forum). There were 5.5 million users in May 2016. These users have posted 12 million questions, 19 million answers, and 48 million comments. Stack Overflow answers around 73% of inquiries. [Stack-Exchange] [Stack Exchange] While this is remarkable, why are 27% of Stack Overflow inquiries unanswered? (Asad Uzzaman et al.) What makes questions appealing to users? Why do certain questions get more consideration than others? Numerous questions may be illuminated by burrowing into the information. This proposition analyzes the quality and how it changes in questions and forms, picking up understanding into subjective estimations of quality in a community like Stack Flood. Can the quality of an address be expected at the time of conception utilizing fair its text?

In recent years, academics have shown an interest in automating quality-based question categorization on Q&A portals (Correa and Sureka, 2014; Ponzanelli et al., 2014; Arora et al., 2015; Yao et al., 2013). The researchers' diverse techniques yielded relatively excellent findings with 73 percent accuracy (Correa and Sureka, 2013) or precision from 62.1 percent to 76.2 percent (Ponzanelli et al., 2014). Previous research shows that reliably classifying a question based on quality is difficult. The first article focuses on the linguistic categorization of Stack Overflow queries. Their work serves two purposes. Our elegant and precise solution to the quality classification challenge may help Stack Overflow moderators discover low-quality queries uploaded to the platform. However, their technique also helps questioner's pre-test their questions before submitting them. With this in mind, they employed natural language processing (NLP) and deep learning techniques to categorize queries by quality, which is thought to be the key factor in shutting or deleting them. Using this method, they attained 74% accuracy and 75% precision. Ponzanelli et al. [2014a, b] examine Stack Overflow queries to forecast quality at the moment of creation. The second article examines how measures affect question quality. The authors choose decision trees because their output is simply comprehended, giving them insight regarding question quality. For categorization, a genetic algorithm replaces the decision tree. The third article employs a genetic algorithm to identify low-quality postings. A print line statement, for example, has a strong negative Pearson correlation with a score, suggesting that it devalues the code fragment. The current work focuses on the quality-based categorization of Stack Overflow questions based on linguistic features. We provide an intuitive and accurate solution to the quality classification issue that may help Stack Overflow moderators identify high-quality and low-quality questions submitted on the site. Therefore, we utilized Time Frequency-Inverse Document Frequency (TFIDF) + Support Vector Machines (SVM) to solve the Natural Language Processing (NLP) binary classification of stack-overflow questions quality rating. The methods in this thesis have various uses. These strategies might be used in Q&A forums to detect and prevent low-quality (potentially aggressive or spam) posts. If integrated into a continuous integration environment for software engineering, these approaches might reject changes unless the documentation fulfills a quality requirement. Journals might examine submitted articles for style and quality before editors personally reviewed them.

2.1 Stack overflow Datasets

The Stack Overflow dataset of posts is central to this thesis. Therefore, this section is solely dedicated to the following:

- Where to Find the Dataset?
- What is in the Dataset?
- Why Use this Dataset?
- Managing the Data

2.1.1 Location of the Dataset

The Internet Archive has the Stack Overflow (SO) dataset. [Stack Exchange] The Internet Archive is a non-profit Internet library. Its goals include providing academics and others with permanent access to digitized historical resources. They also keep a copy of all user-generated material on the Stack Exchange network. Each site is a distinct package of XML files compressed using 7-zip using bzip2 compression. All posts are archived along with their history and links. Only the postings. There is an update every 2 months or such. This data has particular attribution requirements, requiring the author and Stack Overflow to be digitally and visually attributed. In the Stack Exchange Data Explorer and on meta.stackexchange.com, the dataset's schema is described. Users of Stack Exchange Q&A domains may discuss software bugs, features, and support concerns on Meta Stack Exchange. [Stack Exchange].

2.1.2 The importance of this Dataset

The Stack Overflow (SO) dataset contains a wealth of technical information from user-generated questions and answers. However, real-world data will always contain noise, therefore the dataset quality is excellent (essential properties are not missing). Real-world data analysis is significantly more engaging and informative. Real data, on the other hand, approaches cope with factors like noise and missing information. The Stack Overflow data set is widely utilized in machine learning research to compare approach applicability to data. To compare measures and create intuitions about the approaches, other researchers may use this data set to compare metrics and develop intuitions about the methodologies.

2.1.3 Managing the Data

The files connected with Stack Overflow are kept separate rather than being merged into a single package to keep the file size down. The file is called “stackoverflow.com-Posts.7z. ”. This command displays compression/encoding information, archive size, and file size after inflation. displays the 7zip extract command uncompressing the “Posts.xml” file. On a Laptop (Fourth Gen i7, SSD), the eight(8GB) compressed archive. Ballooned to a Forty(40GB) XML file named POSTS.XML.

```
\$ 7z l stackoverflow.com-Posts.7z
Listing archive: stackoverflow.com-Posts.7z

--
Path = stackoverflow.com-Posts.7z
Type = 7z
Method = BZip2
Solid = -
Blocks = 1
Physical Size = 8512952500
Headers Size = 122

  Date       Time       Attr         Size   Compressed  Name
  -----
2016-01-04 16:27:37 ....A 42327180776 8512952378 Posts.xml
  -----
                                42327180776 8512952378 1 files, 0 folders

...

\$ 7z e stackoverflow.com-Posts.7z

Processing archive: stackoverflow.com-Posts.7z

Extracting Posts.xml

Everything is Ok

Size:          42327180776
Compressed:    8512952500
```

Table 2.1.3: 7zip extract command inflates the compressed archive

We extracted the data, analyzed it, and then reduced it to roughly 60,000 lines of text with six columns: ID, Title, Body, Tags, Creation date, and data quality.

Chapter 3

3. Proposed Approach

This thesis researches Machine Learning and Support Vector Machines, which are common supervised learning algorithms used for classification and regression issues. But it is mostly utilized in Machine Learning for categorization. The SVM algorithm's purpose is to find the optimal line or decision boundary that divides n-dimensional space into classes so that subsequent data points may be readily classified. A hyperplane is the optimal choice boundary. SVM selects the hyperplane's extreme points/vectors. These extreme situations are called support vectors, and the method is called SVM. Observe the picture below, where two groups are categorized using a decision boundary or hyperplane.

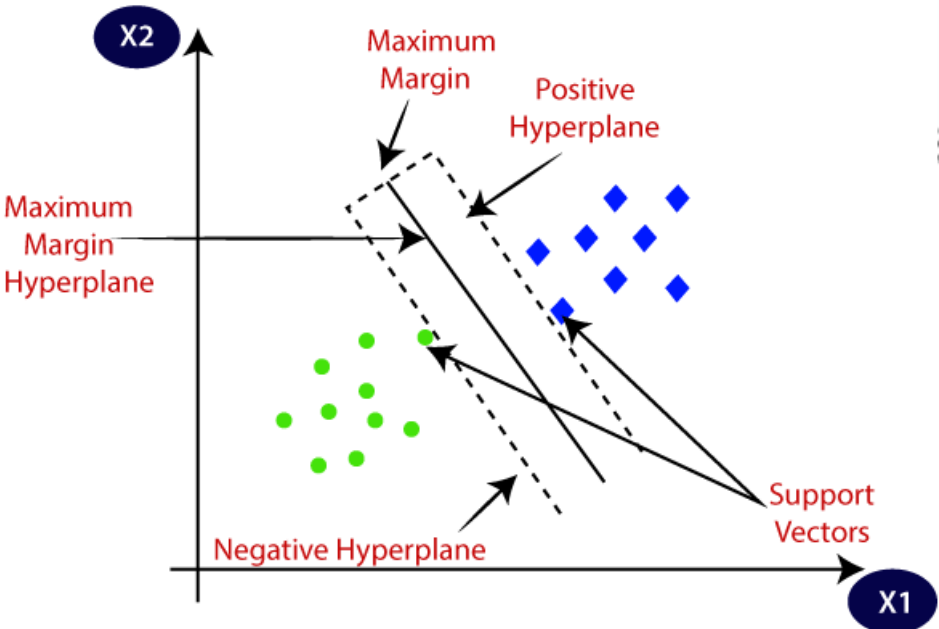


Figure 3.0: categorization using a hyperplane

In addition to the preceding ways, machine learning may be used to classify question quality on Stack Overflow. To categorize questions by quality, which is supposed to be the key perspective in the ultimate choice about shutting or removing them. This study looked at prior research on Stack Overflow question quality.

3.1 Supervised Learning

These approaches use tagged data as input. For example, an input dataset may comprise people's age, gender, and a Boolean value indicating whether they purchased anything. This Boolean value may be used as a target label, allowing an algorithm to learn to anticipate its value for incoming data by looking for patterns in other properties. It may discover that guys under 35 but over 21 are extremely inclined to purchase the goods. So, when unlabeled data says a person is 28 and male, the program predicts they will purchase the product. Sometimes the basic characteristics provide poor forecasts. It may be used to develop new and valuable characteristics to learn from, typically enhancing prediction accuracy. The date of birth in the format “DD/MM/YYYY” as an unstructured string would make this characteristic challenging for an algorithm to learn from. Instead of a mixed ordering, each day, month, and year would be perceived as a single random string. This data may be used to create beneficial features. The most apparent benefit is that age can be determined and utilized in the training process. Once all necessary attributes are present, the data must be translated into a machine-readable format, since learning algorithms cannot comprehend the meaning of words. Once the data is properly represented, a supervised learning algorithm may use it to learn how to predict a target variable. After training and fitting the model, it may be used to make prediction unlabeled data.

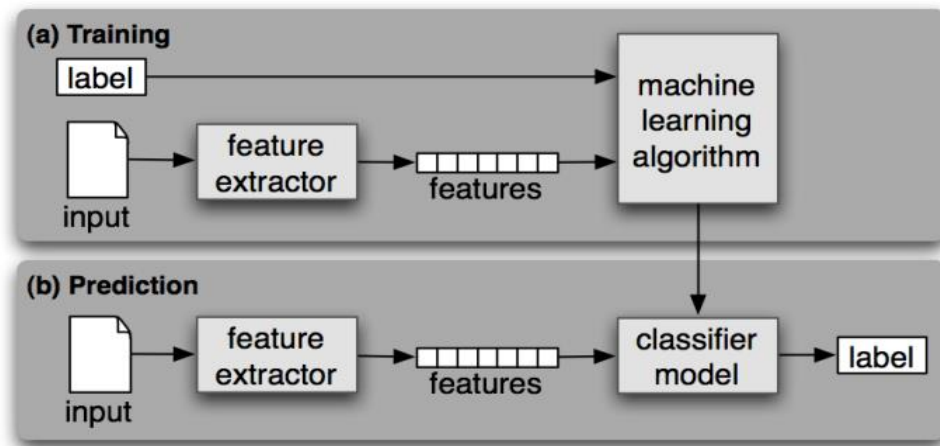


Figure 3.1: The supervised learning process. [NLTK, 2015]

Classification is a sort of prediction that involves forecasting one or more classes/categories rather than a continuous number. This thesis focuses on the quality categorization of text documents into two labels or classes. Supervised document classification involves training on examples of accurate document categorization.

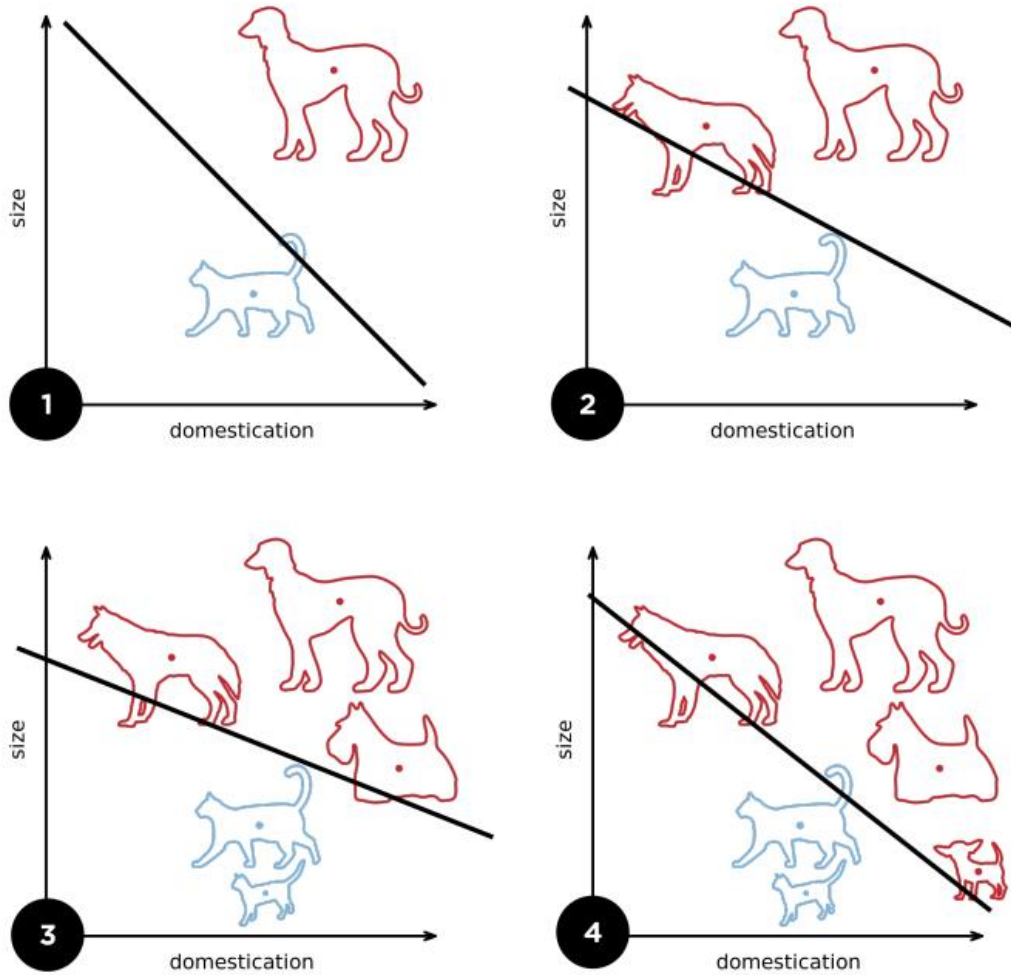


Figure 3.1.1: Perceptron updating its linear boundary as training examples are added
 Goodspeed [2015]

In the graphic above, a perceptron updates its linear border when new training examples of dogs or cats are added. To categorize a fresh unnamed animal as a dog or cat, it learns to balance the variables “size” and “domestication”.

3.2 Methodology

First of all, we split read the data from the drive and label transformation for multi class classification with 0 low quality close, high-quality questions as 1 and Low-quality Edit as 2. Then we unify the title and body in the question part and from there we remove all non- text and original features so that our model can give a better prediction.

We later on proceed to text cleaning where we removed all HTML tags and other links. Then we split the datasets into training, validation and testing sets, for the training we gave it the highest portion for data which is 60%, validation 20% and testing sets 20%.

Then we used TF-IDF algorithm to generate sparse data with weighted word frequencies for SVM classifier. From there we evaluate our tuned TF-IDF + SVM on testing subset and we got the accuracy of 86%.

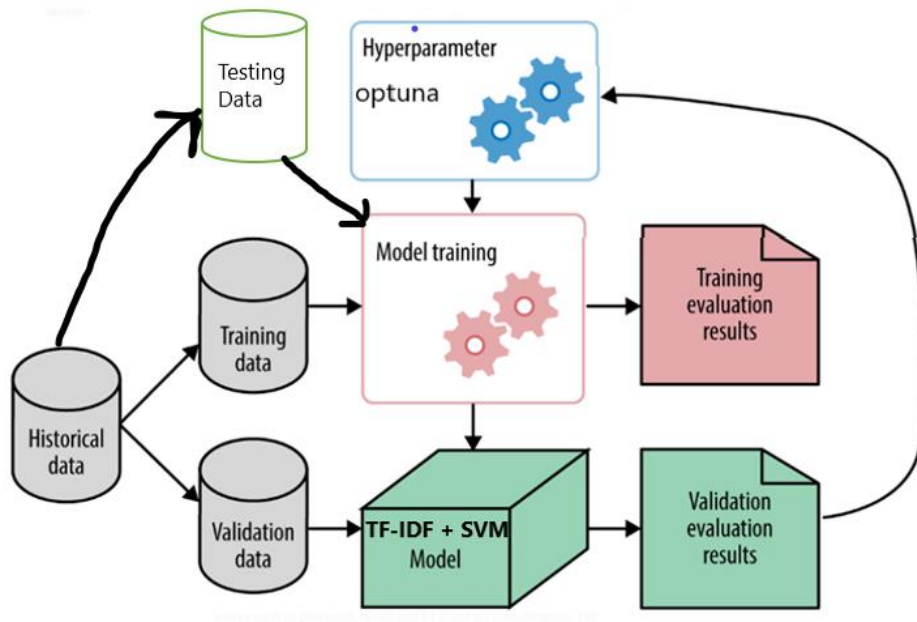


Figure 3.2: Methodology

3.2.1 Data Representation

Data must be represented in a machine-readable format for algorithms to learn from it and predict it. Commonly used feature vectors for input to machine learning algorithms. Feature-Vectors are N dimensional Vector of numerical features. By default, Machines do not get a handle on human ideas. Machines don't translate languages the way common speakers do. Machines cannot see a picture and recognize things as people can. So, this data must be encoded for a computer in a format that it knows, which is numbers, namely binary. We don't see paintings as numbers, but that's how a computer sees them, with each pixel represented by a number. Feature names (presumably represented as strings) are mapped to indices. Take the example from the previous section, the feature names (dog, cat) will be represented as the numbers (1, 2). For inspection purposes when performing tasks such as visualization, these indices can be mapped back to their string representation. For the feature values themselves, there are two main types of data that need to be represented; numerical and categorical. This section also covers how text data is in turn represented in the form of numerical and categorical features.

The quality categories (High Quality and Low Quality) utilized in this thesis are also categorical variables, or more especially ordinal categorical variables, although this distinction should not matter for the purposes of this dissertation. Categorical data is a set of distinct labels that individually represent something. This item might be a number, a color, a clothes size label, an animal species, a flower kind, etc. For example, three animal labels may be [dog, cat, horse]. To utilize machine learning algorithms on text data, it must first be converted. The bag-of-words model is a common simplifying representation that ignores syntax and word order while keeping multiplicity. Each item represents the number of occurrences of a certain word in a document. It incorporates essentialism and reductionism in text data by reducing a text document to its constituent items and doing so without regard for relationships between them. To train a classifier, the bag-of-words approach is widely employed. This event may be expressed in several ways, such as existence, absence, or frequency. Other representations strive to model similarity-based information and other types of connections from the data. The bag-of-words model often represents text documents, however not all words are discriminative or distinctive, therefore they produce noise. Term frequencies, named entities, and term pairings may be used to represent documents. A document corpus's vocabulary after stop-word elimination and word stemming. Named entities include persons, organizations, and places. To keep the feature set small, term pairings only include statistically significant term relationships for the document corpus. Word co-occurrence matrices may also explain word co-occurrence. This records word connections, which is useful for identifying word relationships between texts. Many ways exist to extract and infer information from text. As an example, you may create a weighted bag-of-words feature representation by translating term frequency-inverse document frequency (TF-IDF). The TF-IDF weights the data depending on the term frequency of each text and the inverse of the term frequency across texts (inverse document frequency). TF-IDF determines the words that matter most in determining the text's category and eliminates the problem of document length and category frequency in the training sets. TF-IDF measures term frequency (TF) in a document. [Luhn] The raw term frequency $tf(t, d)$ represents the number of times the term t appears in document d . The basic term frequency scheme is $tf(t, d) = f(t, d)$, where tf is term frequency. Other frequency measurements include Boolean, log scaled, and enhanced. $tf(t, d) = 1$ if t is in d , else = 0. It's either $1 + \log(t, d)$ or 0, depending on what's true. $tf(t, d) = 0.5 + 0.5 \frac{f(t, d)}{\max_{t'} f(t', d)}$ The inverse document frequency component measures the term's frequency across all documents. [Jones, 1988] You may get it by dividing the total number of documents by the total number of

documents containing the phrase, and then multiplying the resulting quotient by the logarithm of that quotient. Here, N is the total number of documents in the corpus, and $|dD:t|$ is the number of documents that include the word t . This indicates term frequency. $TF-IDF = tf(t, d) idf(t, D)$. A high term frequency and low document frequency of the term in the collection of documents results in a high TF-IDF. The IDF's log function filters out frequent terms in TF-IDF. As more documents include a phrase, the ratio within the log approaches 1, lowering the IDF. TF-IDF is a dot product of TF and IDF. Character N-Grams are similar to the bag-of-words representation, but instead of characters. [Cavnar et al. Instead of splitting on word boundaries, it employs n-character splits as features. This method is best suited for the text from noisy sources, such as user-generated material containing code and markup, as in Stack Overflow postings. Character N-Grams are extensively used in this dissertation and are addressed in the section on Feature Engineering and Extraction.

3.2.2 Preprocessing

Preprocessing is a crucial stage in data mining. In many machine learning methods, “garbage in, garbage out” remains true. Data frequently have difficulties including out-of-range numbers, missing values, and too much volatility. Because of these flaws, the algorithm may provide erroneous and misleading results. It is more difficult to find knowledge during the training phase when there is a lot of irrelevant or duplicate material. Preparing and filtering data might take a long time. These steps involve cleaning, normalizing, and transforming raw data. Feature scaling, or data normalization, is a technique for standardizing the range of independent variables or features. Scaling features to a minimum and the maximum value is termed min-max scaling. $X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$. Using max absolute scaling, features are scaled such that their absolute values are smaller than a maximum value.

3.2.3 Training and Testing Strategies

After preprocessing, the data may be used to train and evaluate supervised machine learning models. The test data must be entirely fresh and unseen, as the learning model's prediction accuracy is checked. The performance measurements will be artificially increased if test data leaks into the learning process. This may be achieved through a variety of training and assessment methods. The easiest strategy is to divide the data into train/test ratios of 70/30. So, the training phase only learns from the learning data and not the testing data. The test data is thus fully unknown, providing a realistic test of how the model would perform on real-world unknown data.

With cross-validation, you may test how well your statistical conclusions generalize to a new data set. K-fold cross-validation divides data into k equal folds. One-fold is kept as validation data for testing the model, while the other k folds are utilized as training data. The cross-validation is then done k times across all of the distinct fold permutations. One may then average the k fold findings to get a single guess. Unlike the typical train-test split technique, all observations are utilized for both training and validation.

3.2.4 Decision Trees

Decision Trees (sometimes known as Classification Trees when used for classification) are attractive because, among other data mining methods, decision trees have various advantages. They are simple to understand as they can quite easily be visualized and graphed, to the extent that they are considered a white-box model meaning that the results are easily explained due to the inherent Boolean logic. They require little data preprocessing such as normalization and are efficient and robust in use. In fact, tree-based learning algorithms are very unique in the trait of being scale-invariant.

As a result of using this method of split selection, decision trees have the ability to estimate how important features are in the classification process. The relative rank (depth) and frequency of a feature in a tree can be used to estimate feature importance with respect to the predictability of the target variable.

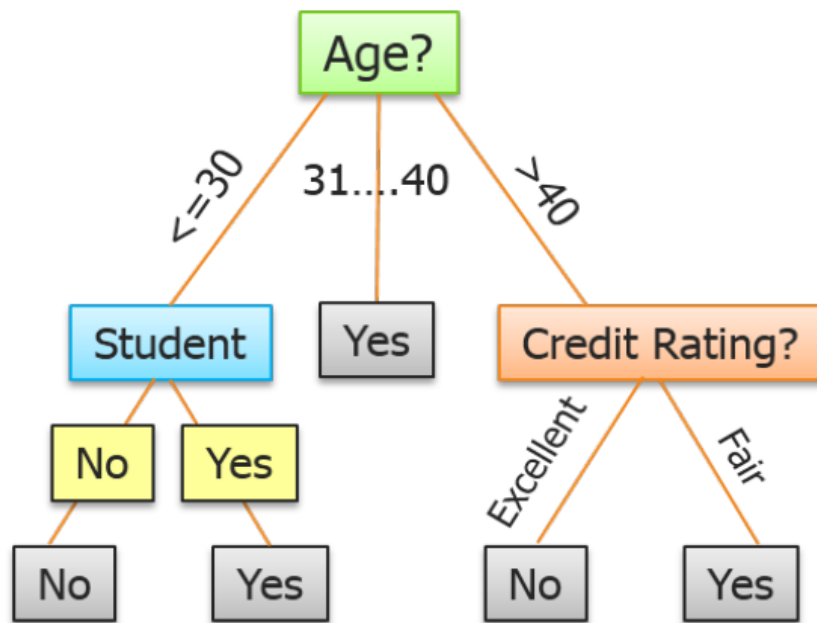


Figure 3.2.4: Decision tree

3.2.5 TF-IDF+SVM Parameter Tuning using Optuna

What is Optuna?

Optuna is a machine learning software framework for autonomous hyper-parameter tuning. It's small, flexible, and cross-platform. It employs Python syntax for conditions and loops. It uses cutting-edge algorithms to sample hyper-parameters and prune unproductive trials. It facilitates parallelization and gives good visualization for future research.

Optuna uses the phrases Study and Trial. Iteration is the optimization of a single objective function. The purpose of the research is to fine-tune our model using a pre-selected set of parameters using repeated trials and the Optuna framework. Selecting Parameters:

The SVM algorithm uses several parameters. We can select the following parameters for the algorithm training but our algorithm used the "RBF Kernel".

- **Kernel:** Indicates the type to be utilized within the algorithm. Conceivable values are Poly, Linear, Sigmoid, and RBF.
- **C:** Regularization parameter. The quality of the regularization is contrarily relative to C. It must be entirely positive. The punishment could be a "Squared-l2-Penalty".
- Degree
- Gamma

We train this model by calling its fit strategy and return the Model-Accuracy to the caller. Optuna runtime will call this function numerous times during study, it record the precision at each iteration and will return the finest accuracy at the very end. You choose on the number of iterations - trials. Isn't it basic?

```
metrics = ['accuracy', 'balanced_accuracy', 'logistic_loss', 'recall', 'precision']
tfidf_svm_metrics = {metric: [] for metric in metrics}
tfidf_svm_study = optuna.create_study(direction="minimize")
tfidf_svm_study.optimize(tfidf_svm_objective, n_trials=20, timeout=60 * 60 * 2)
tfidf_svm_study.best_params.
```

Optuna provides a method to create a study to start the process. The n trials decide the number of iterations alongside the best trial. Each and every iteration may take a considerable sum of time to complete. Depending on the needs, setting this to a proper value yields the finest result. The bigger the value, the superior would be the Outcomes-Results. After getting the values, then you can pick up the best parameter value and use it for model fitting. Train the model on the training dataset and check the model accuracy. From there you can see an enormous improvement in the model accuracy because we used the hyperparameter tuning (Magic of Optuna?).



Figure 3.2.5: Trials

Chapter 4

4. Experimental Evaluation and Result Analysis

4.1 Overview

This proposed dissertation would identify quality indicators and utilize them to categorize Stack Overflow questions accurately. With the explosion of technical queries on sites like Stack Overflow, more knowledge is accessible than ever. This thesis' implementation has two key functional components: data management and machine learning. Obtaining, storing, analyzing, and interacting with data is part of data management. Specific elements of these utilise quality data sets established by the thesis' data management component, and output findings of classification success and insights such as which attributes were most essential.

4.2 Experimental Setup

The Internet Archive has the Stack Overflow data set. [Stack Exchange] After downloading, inflate the archive to get a file like "Posts.xml". The chosen data set has numerous labeled columns and is trained using Support Vector Machines (SVM) for classification tasks in Machine Learning.

4.2.1 Dataset Preparation

The Internet Archive has the Stack Overflow dataset. 15c Stack-Exchange the Internet Archive is a non-profit Internet library. Its goals include providing academics and others with permanent access to digitized historical resources. It also keeps a copy of all user-generated material on the Stack Exchange network.

4.2.2 Dataset Description

Large volumes of real-world data from the Stack Overflow Stack Exchange Q&A area. As stated earlier, over 5.5 million people have asked 13 million questions, answered 19 million, and left 48 million comments as of May 2016. [StackExchange,2015a]. In the Stack Exchange Data Explorer and on meta.stackexchange.com, the dataset's schema is described. Users of Stack Exchange Q&A domains may discuss software bugs, features, and support concerns on Meta Stack Exchange.

Table 4.2.2: Schema of the post's dataset

Key	Description of column
Id	Unique identifier for each question
PostTypeId	1 = Question
AcceptedAnswerId	Id of the accepted answer for a question
ParentId	Id of the question an answer is associated with
CreationDate	Datetime of the post creation
Score	Number of Up votes – Down votes for a post
ViewCount	Times the post was viewed
Body	Text of the question or answer (HTML)
OwnerUserId	User Id of the post
LastEditorUserId	User Id of the last editor of the post
LastEditorDisplayName	User display name of the last editor of the post
LastEditDate	Datetime of the most recent edit to the post
LastActivityDate	Datetime of the last action on the post
Title	Title of a question (null if answer)
Tags	Associated tags of the question, e.g. Java, Android, Machine Learning...
AnswerCount	Number of answers for the question (null if no answers)
CommentCount	Number of comments on post
FavoriteCount	Number of times the post has been favorited
ClosedDate	Datetime when the post was closed (null if the post is open)
CommunityOwnedDate	when the post was community wikied

The posts dataset found on the previous page shows the schema of the Posts dataset, and briefly describes each attribute. Unfortunately, deleted posts are not included in the POSTS.XML file from the archive. Certain Information from deleted posts can be obtained from the POSTS with Deleted tables, which can be accessed at <http://data.stackexchange.com/stackoverflow/> [Stack-Exchange, 2015b]. However, when a post is deleted only a subset of its attributes are stored Even though the important information is missing from these posts to be part of the quality analysis themselves.

4.2.3 The software and hardware materials

The experiments have been performed on an Intel Core I7 6500 2.98 GHz processor with 4 Cores. The used chipset was the Intel Skylake Series and a RAM of 8 GB @ 2600 Mhz. All the experiments have been performed in Python language on the platform of Google Collab.

4.3 Result Analysis

These findings were obtained from the Implementation section. These findings include classification reports, feature importance histograms, and tables illustrating feature distribution across question and response quality categories. These findings enable the evaluation and comparison of strategies such as learning algorithms.

4.3.1 Experimental Results

The classification of both qualities proves to be statistically significant results, with an accuracy of 87% across all runs. Increasing the amount of data and number of trial iterations in the Optuna framework significance test would improve the classification results and significance level even further, however computational resources and time are finite resources. The logistic loss of 4.441725344901136 was achieved. The following results were all generated using 20000 samples. This sample size was chosen in order to keep the categories and amount of data consistent because it is roughly the smallest category size across all the quality distributions.

4.3.2 Analysis

Machine Learning approaches were selected for their impact on quality classification performance and openness in acquiring insight into what impacts performance. The quality attributes and how they change across the quality classes might be identified by examining the performance of the quality classification. First, this thesis was written in Python. Python is a dynamic high-level library with the standard performance against development ease trade-off. This turned out to be a terrific option, thanks to Python's excellent resources and community. Some of the key machine learning algorithms in this thesis involve storing the whole dataset in memory. This is not a scalable approach, but there are numerous alternatives. Some algorithms in Scikit-Learn use an approach termed “partial fit” to simulate online learning. Online learning allows an algorithm to train on data segments rather than everything at once. To save the data in memory, just a portion of it at a time. This is a problem since we need to learn from additional data. This problem did not significantly affect the conclusion of this study, but it should be considered for future iterations. The supervised learning algorithm used was crucial. SVMs were selected for their efficiency, robustness, intuitiveness, and ease of tuning. Tuning parameters include maximum decision tree depth, number of decision trees, minimum split or leaf node depth samples, and number of characteristics to employ. Performance, precision, and behavior may all be changed using these factors.

Chapter 5

5. Conclusion and Future Work

This thesis sought to identify quality markers and utilize them to appropriately categorize Stack Overflow queries. The aim was attained, as shown by factual and anecdotal data. The findings show how effectively each characteristic reflects the quality and how accurate the categorization method is. This thesis will ideally serve as a basis for future research in this field, allowing others to learn from it and improve classification accuracy and quality insight. This thesis illustrates that these strategies work well enough to be employed in real-world situations, but that much more can be done to enhance them. Volunteers from the community moderate Stack Overflow. Automating the quality analysis of Stack Overflow postings would be very beneficial.

To improve classification performance, new techniques, notably for content analysis and validation, will need to be developed and deployed. The strategies utilized were detailed and reviewed, as well as alternative approaches and future work to overcome some of their drawbacks. This includes both modest adjustments to the process and major changes to the project's strategy and design. Quality will always be a subjective concept that differs from person to person. This implies flawless performance is unlikely. In an ideal system, quality classifications are as accurate as subject matter experts. This dissertation is a step towards making this a reality.

Reference

Asaduzzaman, M. Mashiyat, A. S., Roy, C. K., and Schneider, K. A. (2013). Answering questions about unanswered questions of stack overflow. In Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13, pages 97–100, Piscataway, NJ, USA. IEEE Press.

[www.researchgate.net/publication/333574762/](http://www.researchgate.net/publication/333574762) Towards an Accurate Prediction of the Question Quality on SO using a Deep Learning-Based (NPL Approach).

Goodspeed, E. (2015). A diagram showing a perceptron updating its linear boundary as more training examples are added. <https://en.wikipedia.org/wiki/Perceptron/>.

Duijn, M., Kućera, A., and Bacchelli, A. (2015). Quality questions need quality code: Classifying code fragments on stack overflow. In Proceedings of the 12th Working Conference on Mining Software Repositories, MSR '15, pages 410–413, Piscataway, NJ, USA. IEEE Press.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317.

L. Ponzanelli, A. Mocchi, A. Bacchelli, M. Lanza, and D. Fullerton, "Improving Low Quality Stack Overflow Post Detection," in University of Lugano, 2014. [Online]. Available: <http://www.inf.usi.ch/phd/ponzanelli/profile/publications/2014e/Ponz2014e.pdf>.

Baltadzhieva, A. and Chrupała, G. (2015). Predicting the Quality of Questions on Stack Overflow. In Proc. of the International Conference Recent Advances in Natural Language Processing.

Barua, A., Thomas, S. W., and Hassan, A. E. (2014). What are developers talking about? An analysis of topics and trends in Stack Overflow. *Empirical Software Engineering*.

Correa, D. and Sureka, A. (2013). Fit or unfit: Analysis and prediction of 'closed questions' on stack overflow. In Proc. of the First ACM Conference on Online Social Networks.

Saini, T. and Tripathi, S. (2018). Predicting tags for stack overflow questions using different classifiers. In 2018 4th International Conference on Recent Advances in Information Technology, pages 1–5.

Schuster, S., Zhu, W., and Cheng, Y. (2017). Predicting Tags for Stack Overflow Questions. In Proc. of the LWDA 2017 Workshops: KDML, FGWM, IR, and FGDB.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In Dasgupta, S. and McAllester, D., editors, Proceedings of the 30th International Conference on Machine Learning, volume 28 of Proceedings of Machine Learning Research, pages 1139–1147, Atlanta, Georgia, USA.