Predicting the response time of StackOverflow questions

Abhishek Vyas (MT19086) **Chirag Chawla (MT19089)** **Mansi Sharma (MT19092)**

Department of Computer Science and Engineering

Indraprastha Institute of Information Technology, Delhi

Introduction

- Stack Overflow is an open community for anyone that codes.
- Now days most of coders use stack overflow when stuck in some part of coding, to ask the question or to check the answers of questions that have already been asked.
- Now if someone asks a question on StackOverflow, he/she has to wait till the question get answered, and waiting time may vary from question to question depends on the type and difficulty level of the question.
- So here we are trying to predict the time taken to get the answer of the question on the basis of certain feature, syntax and semantics of the question using various NLP, IR and Machine Learning Techniques.

Dataset

- The Data set is consist of 10% of question and answers from Stack Overflow programming QA website, which are taken from kaggle. It has three tables :
- Question : It consist of question id, title, body, creation date, closed date (if answered), score, and owner ID for all non-deleted Stack Overflow questions. There are approx 1.24 Million enteries (Questions) in this table.
- Answer : It contains the body, creation date, score, and owner ID for each of the answers to above questions. The ParentId column links back to the Questions table.
- Tags : It contains the tags used for each question.

Feature Engineering

- We make features based on tags and body (text) of the questions. They can be categorized as :-
- **Body or text (Non-tag) based features :-** These features made using text content of questions data.
  - ❖ Start word of Question Title : Store starting word of Title.
  - ❖ Title end with question mark : True if title end with question mark otherwise False.
  - ❖ Question Created on Week days : The day of week on which question is asked.
  - ❖ Question Creation Hours of day : It store the hour information at which question asked.
  - ❖ Body Length : It store length of body of question.
  - ❖ Time duration to answer first time.: Time taken to answer a question first time from question creation date and time.
- **Tag based features :-** These features are made by doing certain analysis of tags.
  - ❖ Tag popularity : Average frequency of tags.
  - ❖ Tag specificity : Average co-occurrence rate of tags.
  - ❖ Number of active subscribers : User who respond actively within a month.
  - ❖ % of active subscribers: Ratio of Active subscriber to the subscriber of tag
  - ❖ Number of responsive subscribers:User who respond actively within an hour.
  - ❖ % of responsive subscribers : Ratio of responsive subscriber to the subscriber of tag.

INDRAPRASTHA INSTITUTE *of* INFORMATION TECHNOLOGY **DELHI**

Similarity based Models

- **Jaccard Coefficient :** we are using jaccard coefficient method to find the similarity between questions. we consider questions as set of tokens. Now for each test question we converted it to the set and for each train question we calculate jaccard score using this formula : $J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$
- **Linear Regression :** Using various tagged and non-tagged based features, we predict time required to answer questions using Linear regression.For performance measure we calculate mean absolute error and mean squared error.We calculate it for both cases with and without tagged based features.
- **TF-IDF with champion list :** we are using tfidf vectorizer to find the similarity between questions. We have splitted the question data into train and test data, now we are finding the tfidf similarity score of test questions with respect to the training questions, and find the top 5 most similar questions with high tfidf score, and average their response times to get the predicted response time of test questions. And in order to reduce the time we are using champion list method which evaluates only top 100 questions with high tfidf values with respect to the test questions.
- **TFIDF with quality score and fast cosine method on classification model :** Here we are implementing tfidf model based on classification model which means that for the particular test question we predict the class label using classification model, and find the similarity of the test question with all the train questions which are having same label predicted as of test question. Now in order to apply tfidf we are using fast cosine method. And after getting the tfidf score for the train questions with respect to test questions, we have added the score of the question to get the net score values. Then we have extracted square root n (“n” is the number train questions which are having some score with respect to test question) questions having high net score value, average their response times to get the predicted response time of the test question. **Net Score(q, d) = score(d) + cosine(q, d).**

Classification

- With the help of tagged and non-tagged based features we do classification of the questions.
- We have created three classes based on response time of questions which are named as :-
  a) “0” => which have response time less than or equal to 16 minutes.
  b) “1” => which have response time greater than 16 minutes but less than or equal to 1 hour.
  c) “2” => which have response time greater 1 hour.
- Using these features and with the help of various classification methods we do predictions of class in which questions belong to.For performance measure we calculate accuracy and f1_score of of these methods.The used methods are :- 1) **KNN** 2) **Gaussian Naive Baye** 3) **Support Vector Machine**

Results

- **Jaccard Coefficient method** : Mean Absolute Error is: **593.14 hrs** and Mean squared Error is **2623.67 hrs**
- **Linear Regression :-** Mean Absolute Error is: **518.36 hrs** and Mean squared Error is **2291.41 hrs**
- **TF-IDF with champion list :** Mean Absolue Error is : **392.99 hrs**
- **TFIDF with quality score and fast cosine method on classification model :** Mean Absolue Error is : **237.85 hrs**
- **KNN :-** F1 Score is **0.5804**
- **Gaussian Naive Bayes:-** F1 Score is **0.5826**
- **Support Vector Machine :-** F1 Score is **0.5833 .**