Mining CQA Data

Response Time Prediction

Phani Santhosh Vamsi Deepak Darbha	**Akarsh Cholaveti**	**Venkata Guru Sai Vemulakonda**
pdarbha@asu.edu	acholave@asu.edu	vvemulak@asu.edu
1208681717	1208568487	1208580954
Venkata Vineel Vutukuri	**Prerna Satija**	**Surbhi Aggarwal**
vvutukur@asu.edu	psatija1@asu.edu	saggarw9@asu.edu
1207634125	1206312714	1208920371

***Abstract*—**Community based Question & Answering **(**CQA) forums are gaining reputation day by day. They are proving to be an important source for technical knowledge. Unlike official product documentations they provide solutions to the problems that occur in real time environments. User engagement is the driving factor for the success of any CQA forum. There can be many factors that can influence user engagement. Response time is one such factor that plays a vital role in influencing user engagement of CQA forums which in-turn has an indirect affect on site popularity. So the next question would be what are the factors affecting response time? In this project we estimated the response time for questions posted in StackOverflow website. We analyzed many factors from the data and have identified a few factors which has an impact on the response time. We divided the factors into three categories Tag based, Non-tag based, Questionnaire based. We analyzed how each of them has an effect on response time. Finally, we trained models using the identified evidential features for predicting the response time of newly posted questions.

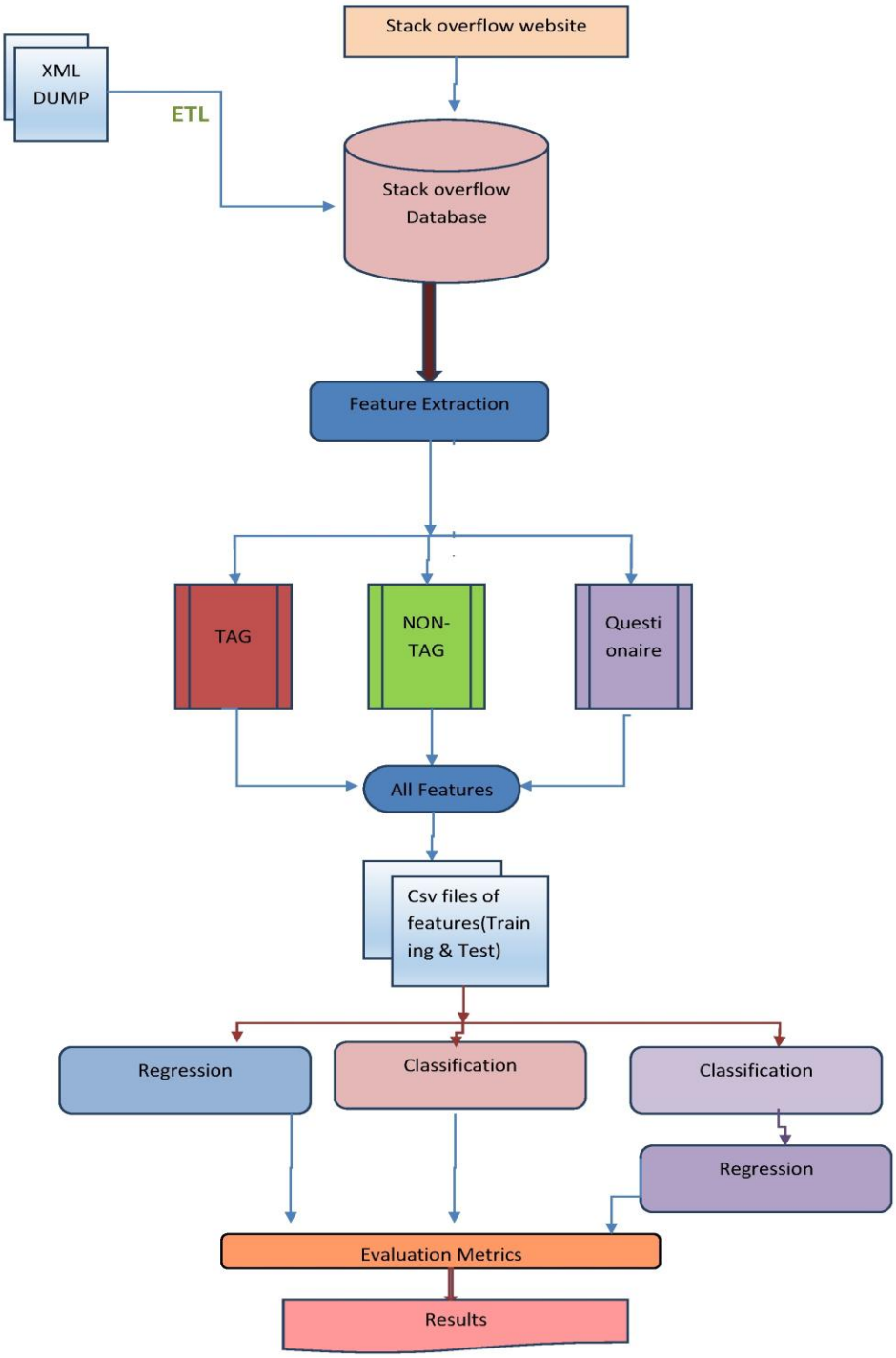
I. INTRODUCTION

With the enormous increase in usage of web, CQA sites like Yahoo answers, Stack overflow, Wiki answers are becoming very popular. These are community based platforms aimed at community based knowledge sharing, whose end product is a knowledge repository. Their main goal is to be a continually evolving source of good information. They allow users to post and answer questions. Stack overflow[1] is one such website which features queries on broader range topics related to computer science and programming. It is based on a system rewarding badges and reputation points to users who post useful questions. A question receiving a huge number of up- votes suggests its usefulness of providing knowledge to users. In such system where reputation/user engagement plays a crucial role response time has a huge impact in determining the site popularity. People who post questions would want to know the time by which they can expect a response to their question so that it prevents repeated polling to the website. From our analysis, 60% of the questions are answered in an hour, and nearly 20% of the questions are answered after a day. So it is very critical task to predict the response time to keep the user engaged with the site. Predicting the response time in Q&A sites is very challenging as there are many factors involved which affects it. In our analysis we identified many factors that affect response time. So we divided them into three categories, Tag based, Non-tag based,

[1] http://stackoverflow.com/

Questionnaire based. We performed analysis on these features and predicted the response time of the question posted on the site.

PROJECT FLOW:



II. SYSTEM ANALYSIS

Stack Overflow is a community based question answering forum for professional and enthusiast programmers around the world. It allows its users to post questions and/or answer the posted questions. A Post (Question) is considered good or bad based on the number of Upvotes or Downvote it receives. Upvotes[2] or Downvote is an option given to users of the stack overflow, if the user feels the question is valid and informative he/she can Upvote it or else Downvote. This option is given only after a user gets certain threshold reputation points to check their integrity. Reputation[3][3] can be earned by user by being an active member of the forum by posting valid questions or answering to good questions and getting upvotes to their questions/answers. While posting a question user is given an option to tag 5 areas the posted question relates to(for example if a user posts a question on HTML he will include the tags html4, html5, css etc). Users are further given rewards like badges, bounty and special privileges by stack overflow for the increased user engagement.

III. DATA ANALYSIS

We are using the dataset of stackoverflow.com. Stack Exchange made this dataset available in the form of XML data dump through Internet Archive under the cc-by-sa 3.0 license, intended to be shared and remixed. https://archive.org/details/stackexchange. The total data comprises of historical data from 2009-2014. Each XML file corresponds to unique feature modules such as Posts,Users,Tags,Badges.We have taken third quarter of 2014 data as training set and October 2014 data as test set for our analysis.
Some statistics of subset data considered for analysis are shown in below table

QUESTIONS POSTED	502784
QUESTIONS ANSWERED	354318
QUESTIONS UNANSWERED	148466
AVG TAG PER QUESTION	2.998

Table 1

For response time prediction we have only taken Questions Answered into consideration as unanswered questions will not have any impact. We used unanswered questions for finding out gray areas. Some statistics of answered posts data

[2] http://stackoverflow.com/tour
[3] http://meta.stackoverflow.com/help/whats-reputation

% Questions answered in <30min	53.15%
% Questions Answered in >1day	10.62%
Avg No of Answers for posts	1.53

Table 2

As 10.62 % of questions are answered after a day they impact the prediction of response time of other questions of the tag they belong to while performing regression. As we compute response time in minutes taking direct values of response time for questions answered after a day can produce huge variance to the response time. So for obtaining better results for regression models and to check variance we have made the response time of all questions that are answered after a day as 1440 minutes which is exactly equal to 1 day. Therefore our class column Response time has values ranging from 0 to 1440.

Interesting results from analyzing Tags of 2014 third quarter data number.

TAG FREQUENCY STATISTICS

Total # of tags in the posts	27148
#Tagswith frequency>=1000	190(0.7% of total tags)
# Tags with frequency>=100	1630(6% of total tags)
# Tags with frequency>=50	2907(10.7% of total tags)
# Tags with frequency>=25	4839(17.8% of total tags)
#Tags with frequency<5	13638(50.23% of total tags)
Most Popular Tag	JAVASCRIPT (Tag-Frequency=53820)

Table 3

From the above stats it is clearly evident that Stack Overflow is famous for only small percent of tag areas with JavaScript as most popular area followed by Java, Android, and PHP and C #. So chances of a question getting answered and the response time greatly depends on how popular the tag corresponding to that particular post is. Even to define popularity we planned of taking top 5, 10 and 20% popular tags and so we have taken threshold frequencies of 25, 50 and 100 which approximates our desired percentages.

We can see from the below statistics (in Table 4) that only 227 areas have got subscribers (people who answered to posts related to that tag in third quarter 2014) more than 1000. The

higher the number of subscribers more likely the post corresponding to that particular tag gets answered. Therefore Tag based features might have good impact on response time.

Also there are 12252 tag areas where there are no subscribers. These areas can be analyzed and concentrated by Stack Overflow to increase their website popularity by increasing user engagement in those areas as well. We analyzed such areas where there is a greater chance of user engagement in future and named them as gray areas. We plan to do more analysis in those areas and is discussed in future work section of this project report.

TAG SUBSCRIBERS STATISTICS

TOTAL # OF SUBSCRIBERS	1868744
#TAGS WITH SUBSCRIBERS>=1000	227(0.83% OF TOTAL TAGS)
# TAGS WITH SUBSCRIBERS>=100	1838(6.7% OF TOTAL TAGS)
# TAGS WITH SUBSCRIBERS>=50	3200(11.8% OF TOTAL TAGS)
# TAGS WITH SUBSCRIBERS>=25	5225(19.25% OF TOTAL TAGS)
#TAGS WITH SUBSCRIBERS=0	12252(45.13% OF TOTAL TAGS)
#TAG WITH HIGHEST # OF SUBSCRIBERS	JAVASCRIPT (#SUBSCRIBERS=73481)

Table 4

IV. FEATURE EXTRACTION

We categorized features into 3 categories. Tag-based, Non-tag based, Questionnaire based.

Tag based features	
Tag frequency	Frequency of particular tag in the data
Tag popularity	Tags that are frequent in the data
Num_pop_tags	Number of tags that are popular
Active Subscribers	Number of active subscribers
% Active_sub	Percentage of active subscribers
Resp_subs	Number of responsive subscribers
% resp_sub	Percentage of responsive subscribers

Table 5

Non-Tag based features	
Num_of_images	Number of images in body of a post
Body_length	Total Length of the body
Title_length	Length of the title of the post
Is_weekend	Whether a Question is posted in Weekend
End_Question_mark	Whether a post ends with a question mark
Begin_Question_word	Whether a post begins with WH word e.g.What,Why,Where etc
Num_code_Snippets	Total number of code segments in the body of a post

Table 6

Questioner based features	
No. of questions	Number of questions posted previously by the user
Avg. Response Time	Avg response time for previously posted questions of the user
User Reputation	The current reputation achieved by the user

Table 7

Tag Based Features:

Tag frequency: Number of times a particular tag is repeated in all the questions posted.

Tag Popularity: A tag is defined as popular if its frequency is greater than certain threshold. We defined frequency thresholds of 25, 50 and 100 and classified each tag as popular or not using those thresholds.

Num_pop_tags: For each post we then found the number of popular tags.

Subscribers: Subscribers of a tag are the users who have answered to at-least one post during the time period of the considered dataset for that particular tag.

Active Subscribers: A subscriber is considered as active subscriber of a tag (t) if he has posted sufficient answers for more than threshold (th) of the questions containing t in recent past. We have taken thresholds of 10, 20 and 30 i.e. when the user answers more than 10 questions of a particular tag, he/she is considered as an active subscriber for that tag for 10 threshold.

% Active Subscribers: Percentage Active Subscribers is obtained by dividing active subscribers for a particular tag divided by the number of subscribers for that tag. Percentage Active Subscribers is computed for all the thresholds of active subscribers.

Resp_subs: We define responsive subscriber as the subscriber who have answered posts related to a particular tag with in 1 hour from the time the question is posted. We then computed number of responsive subscribers for each tag.

% Resp_Sub: Percentage of responsive subscribers is the ratio of number of responsive subscribers to a particular tag to the number of subscribers to that tag.

NON-TAG BASED FEATURES:-

Num_of_images: Number of images in body of the post.
Body_length: Length of the body [in characters]
Title_length: Length of title of the post [in characters]
Is_weekend: If the posted question is on weekend or not. This is a binary attribute. If posted on weekend it is considered as 1, else 0.
End_Question_mark: If the posted Question ends with a question mark or not. This is a binary attribute. If posted question has a question mark in the end it is considered as 1, else 0.
Begin_Question_word: If the posted questions begin with '**WH**' or not. This is a binary attribute. If posted question begins with WH it is considered as 1, else 0.
Num_code_Snippets: Total number of code snippets posted in the question.

V. FEATURE ANALYSIS

Features based on tag: From the below figure[Fig 1] we observe that as the number of subscribers for the tags increases the time required for receiving an answer to that post decreases. so if we include these tags there is high chance of receiving answers very quickly. Y axis is the Average response time for the period of data in X-axis

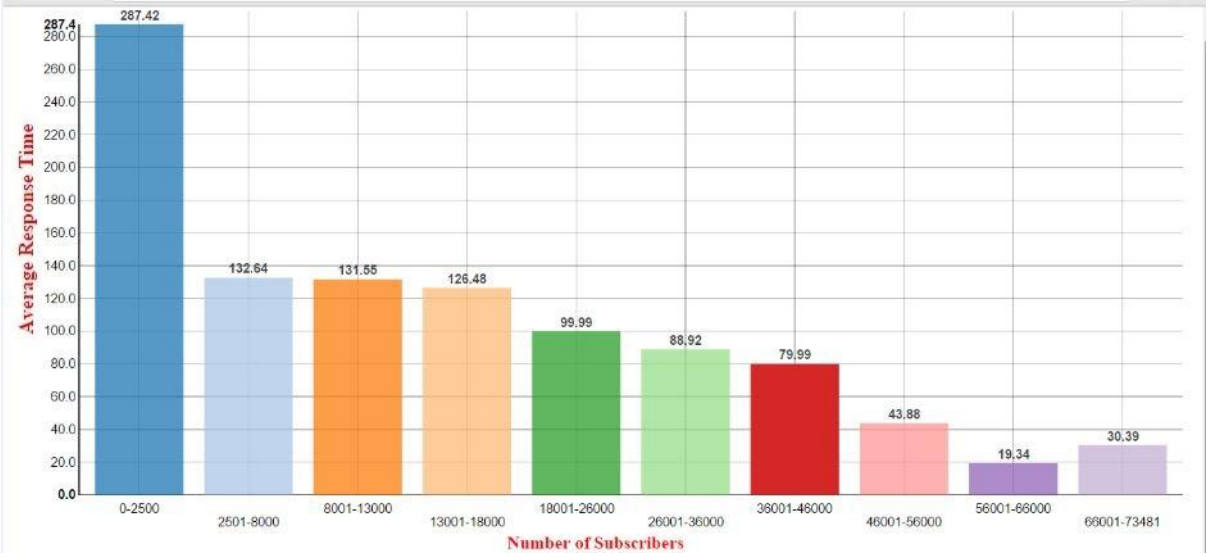


FIGURE: 1

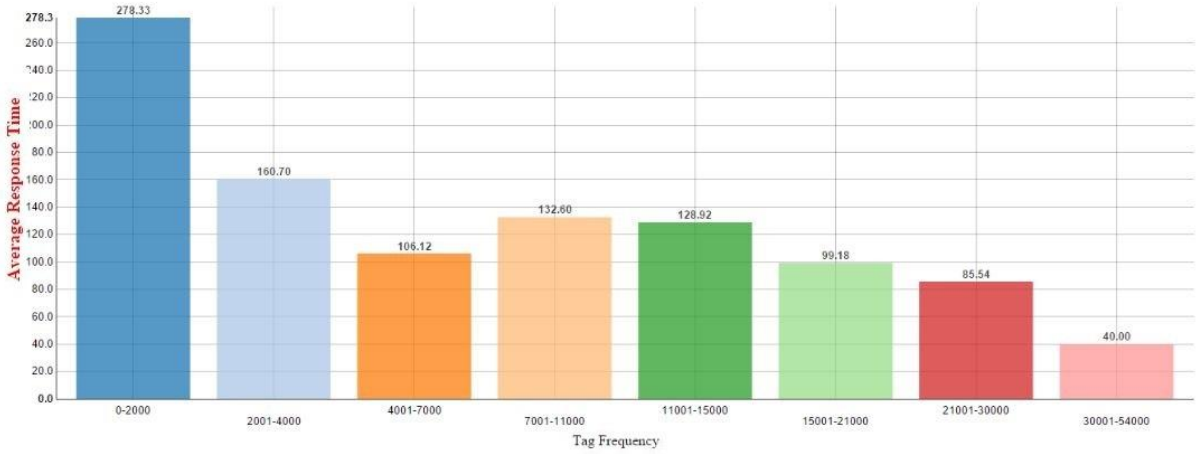


FIGURE: 2

From figure 2 we can imply that using more popular tags in our post we can receive the response to our question sooner. We can observe that tag frequency and response time are inversely related. Y axis is the Average response time for the period of data in X-axis

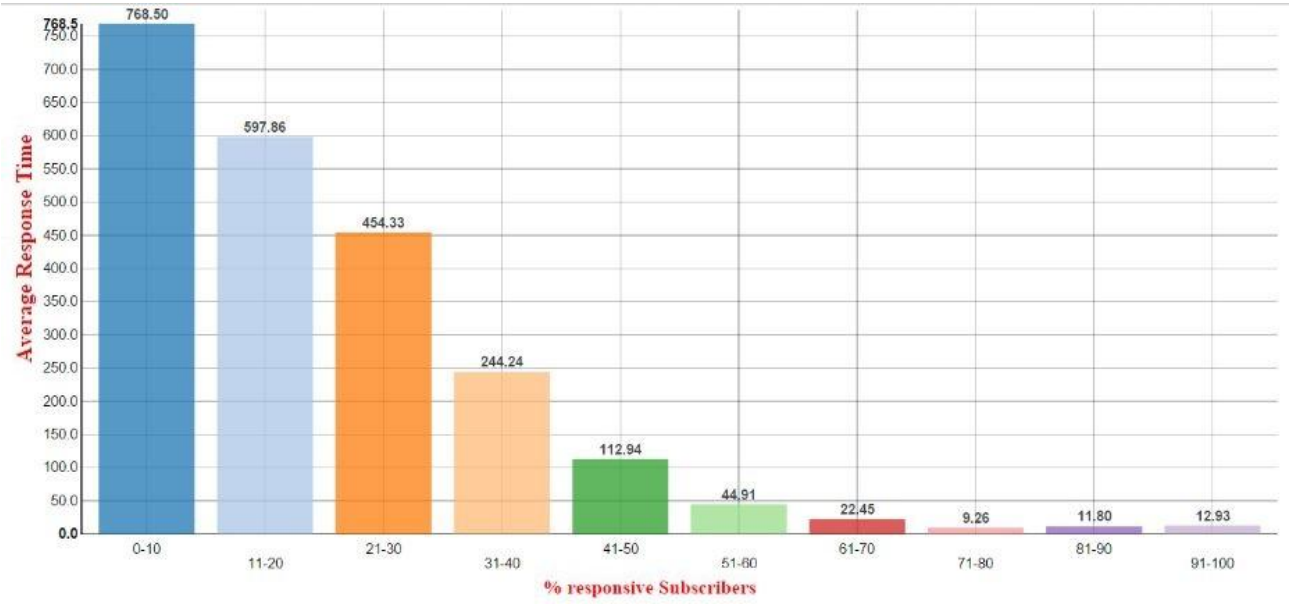


Figure: 3

% responsive subscribers has a greater impact on the response time. As the % of responsive subscribers increases the response time decreases drastically which is evident from above figure 3.Y axis is the Average response time for the period of data in X-axis

Non-Tag Features:

Figure: 4

From figure 4 it is evident that even Non tag features have an impact on response time. But the impact of non tag features is comparatively lower than that of tag features. It is clearly understood that as the length of the body content increases it increases the response time for expected solution to your post. So if we keep the question precise the probability of getting the response earlier is high. Y axis is the Average response time for the period of data in X-axis

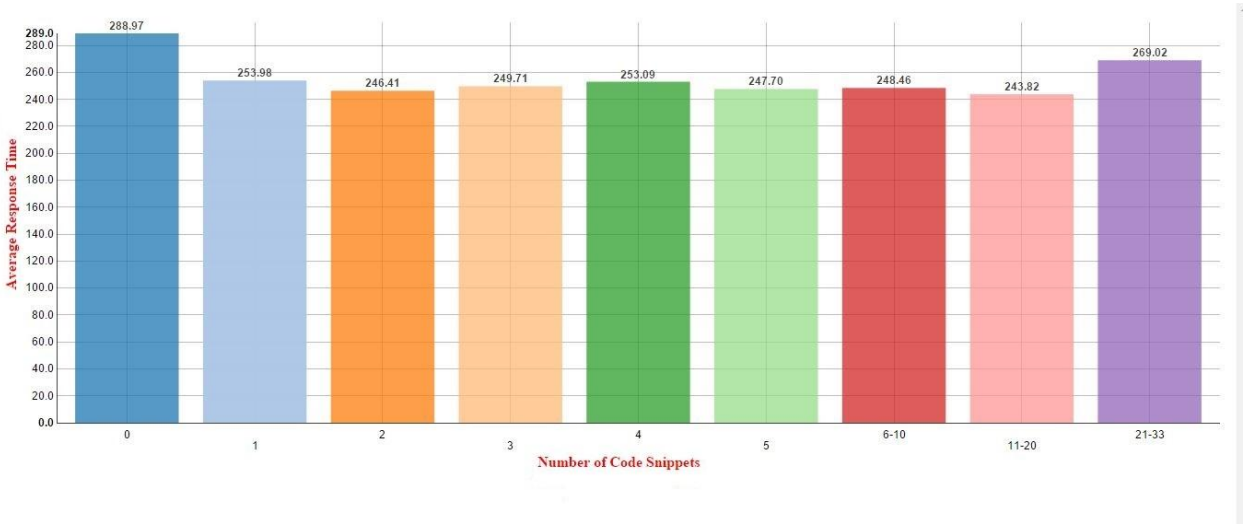


FIGURE: 5

From the above figure we can conclude that if there are Zero or high code snippets in your post you may have to wait little longer to receive the response.

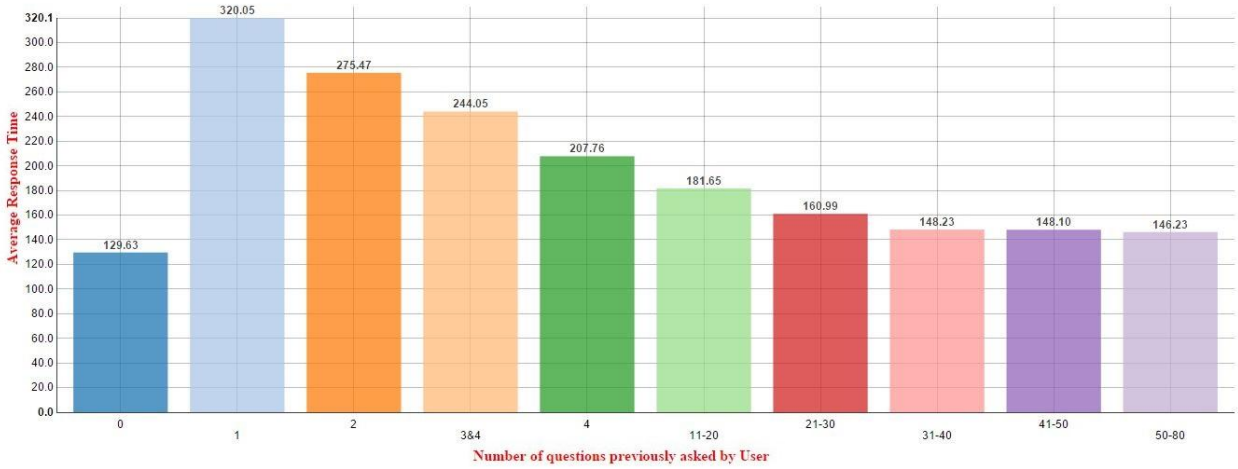


FIGURE: 6

VI. PREDICTION OF RESPONSE TIME

By considering the above stated features we predict the response time of the test set. For predicting the time initially we classified the response time into 2 classes.

1. Response time of questions that are answered within 30 min (<=30 min) are labeled as 1 and
2. Response time of questions that are answered after 30 min (>30min) are labeled as 2.

Here we considered 30 as split because the median answer time of the data (Subset) we considered is 30 min.

Later for providing better insight to user we further analyzed the feasibility of classifying response time into this into three class labels which are

1. Response time of questions which are answered within 30 min (<=30 min) are labeled as 1.
2. Response time of questions which are answered after 30 min and within 5hr (>30 min and <=300 min) are labeled as 2.
3. Response time of questions answered after 300 min (>300 min) i.e. after 5 hrs are labeled as 3.

We then trained this classified models and did a comparative study for binary classification and tertiary classification as mentioned above. Classification accuracies are measured using Confusion matrix and F1 measure and the values are discussed in the Results section of this project document.

We also analyzed the feasibility of predicting response time as a continuous variable using different Regression algorithms. But because of high variance in the values of response time regression algorithms did not yield good results for our proposed set of features. Also we tried approach of classifying data first and then regressing on the top of classification results to see if we can improve regression algorithms performance. For this we need to have best classification accuracies possible. We tried different ensemble classifiers but could only classify to a maximum of 71 percent accuracy which is 3 percent more than accuracies obtained in our base reference paper. But even this approach did not yield good results for predicting response time as continuous variable.

Different classification and Regression models that are used for predicting response time are discussed in detail below.

Classification Models:

We have used different classification models:

Decision Tree[4]:: Decision Trees (DTs) are a non-parametric supervised learning method used for *classification* and *regression*. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Decision trees can computed very fast and easy to interpret even though it tends to overfit.

RandomForest: Random forests[5] are an ensemble learning methods for classification and other tasks, that operate by constructing a multitude of Decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests correct for decision trees' habit of overfitting to their training set. Generally, Ensemble algorithm performs better than normal algorithms and provide better accuracy. So, we used Random Forest classifier for the data classification.

Logistic regression: Logistic regression[6] is direct probability model used for data classification. In this project we used binary logistic model which is used to predict a binary response based on one or more predictor variables, that is , it is used in estimating the parameters of a qualitative response model. The probability describing the possible outcomes of a single trail are modelled, as a function of the predictor variables used by logistic function.

Logistic regression mainly used for binary classification. As we are performing binary classification for the data we used Logistic regression.

SVM: **S**upport vector machines[7] (SVM) are supervised learning models used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one

[4] http://en.wikipedia.org/wiki/Decision_tree
[5] http://en.wikipedia.org/wiki/Random_forest
[6] http://en.wikipedia.org/wiki/Logistic_regression
[7] http://en.wikipedia.org/wiki/Support_vector_machine

category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Given our dataset is huge SVM usually do not overfit the data. It gives best results for the data with less number of classes.

AdaBoost: AdaBoost[8] (Adaptive Boosting), is used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. It can be less susceptible to the over fitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing (i.e., their error rate is smaller than 0.5 for binary classification), the final model can be proven to converge to a strong learner.

We selected Ada Boost as it is less susceptible to over fitting and weighted sum of weak learners is considered.

Gradient Boost: Gradient Boosting Classifier[9] builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n classes regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function. Binary classification is a special case where only a single regression tree is induced.

KNN Classification[10]: Matlab function *'knnclassify'* is used to achieve k nearest neighbor binary classification. We used 'euclidean' distance measure. Optimal value of k was computed by plotting different values of k against accuracy obtained through Confusion Matrix. This was done for two-class classification by classifying the 'first answer time' of training set into two classes with labels 1 and 2. Class 1 corresponds to posts with first answer time <= 30 and class 2 with first answer time > 30.

Discriminant Analysis: Linear Discriminant Analysis[11] is a classification algorithm that uses a linear combination of features to categorize two or more classes. LDA works when the measurements made on independent variables for each observation are continuous quantities. Matlab function *'classify'* is used to achieve the binary classification based on 'first answer time'

Regression Models:

[8] http://en.wikipedia.org/wiki/AdaBoost
[9] http://scikit- learn.org/0.13/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html
[10] http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[11] http://en.wikipedia.org/wiki/Linear_discriminant_analysis

Support Vector Regression:
For the project, to predict the first answer time we used SVR Regression[12] for our data set. We used scikit library in python for implementing the algorithm and we used radial basis kernel for our regression model, penalty parameter(C) as 1 and kernel coefficient (gamma) as 0.1

Random Forest Regression:
For the project, to predict the first answer time, we used another regression model called Random Forest Regression[13] on our data set. We used scikit library in python for implementing the algorithm and we used weak learner as Decision Trees and number of trees in forest (n_estimators) as 200.

Ada Boost Regression:
For the project, to predict the first answer time, we used another regression model called Ada Boost Regression on our data set. We used scikit library in python for implementing the algorithm and we used weak learner as Extra Tree Classifier, number of weak learners (n_estimators) as 200 and we used Ada Boost algorithm as 'SAMME', which a discrete boosting algorithm.

Gradient Boosting Regression:
For the project, to predict the first answer time, we used another regression model called Gradient Boosting Regression on our data set. We used scikit library in python for implementing the algorithm and we used number of weak learners (n_estimators) as 500, learning rate as 0.05, maximum depth of individual estimator (max_depth) as 4 and loss function as 'ls', which means least square regression, this function will be optimized while performing the regression.

Linear Regression:
Linear Regression is a technique used to predict the value of a dependent variable using one or more independent variables. The case when prediction is done for only one dependent variable, it is called Simple Linear Regression. For more than one dependent or explanatory variable, it is called Multiple Linear Regression or Multivariate Linear Regression.

Neural Networks:
Neural Networks[14][15] are a family of statistical learning algorithms that can learn non-linear functions from continuous or discrete variables. A simple neural network can be interpreted as layers of sigmoid units.

[12] http://en.wikipedia.org/wiki/Support_vector_machine
[13] http://en.wikipedia.org/wiki/Random_forest
[14] http://en.wikipedia.org/wiki/Artificial_neural_network
[15] http://www.cs.cmu.edu/~tom/10701_sp11/slides/NNets-701-3_24_2011_ann.pdf

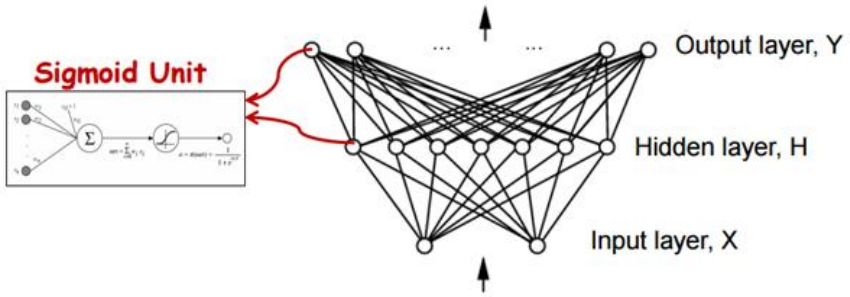


Fig: 7 Artificial Neural Network

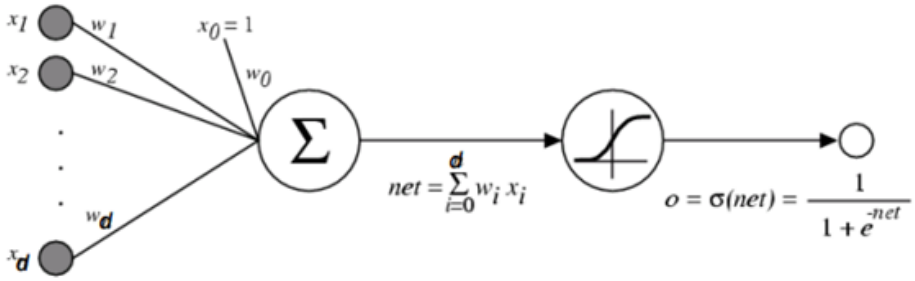


Fig: 8 ANN computation

After each neuron weighted and transformed by the sigmoid function, the activations of these neurons are then passed on to other neurons. This process is repeated until finally, an output neuron is activated.

ARCHITECTURE AND PARAMETERS:

A feed forward network with sigmoid hidden neurons are used in this study. The network is trained with Levenberg-Marquardt back propagation algorithm.

The number of layers is varied from 10 to 200 and the number of neurons in each layer is fixed as 22 as there are 22 features. The cross validation error is measured and the results are compared. It is found that the best CV results are observed with 100 layers.

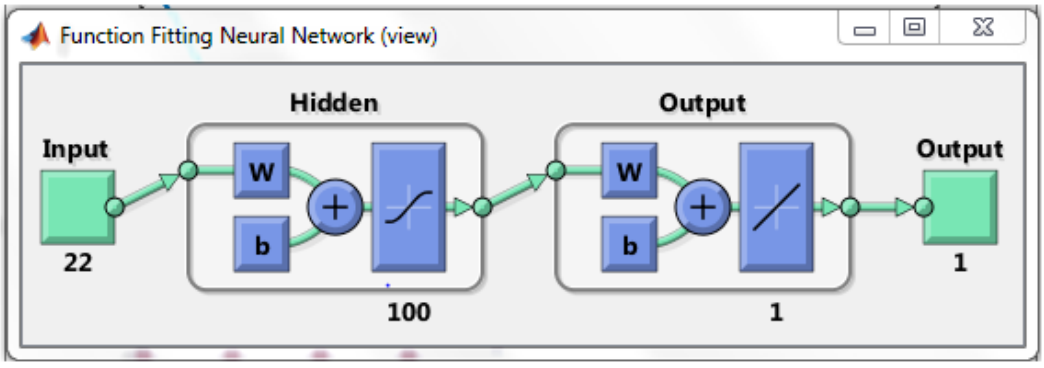


Fig: 9 Function Fitting Neural Network

K-NN:

k-Nearest Neighbors algorithm is a Machine Learning algorithm used for both classification and regression. It considers the k closest training examples in the feature space based on the distance metric used. The basic idea is to determine the missing value/ label of a sample based on what values/labels its closest neighbors have. Like in figure K.1.1, class of the unknown record can be determined to be '+' based on its 3 neighbors.

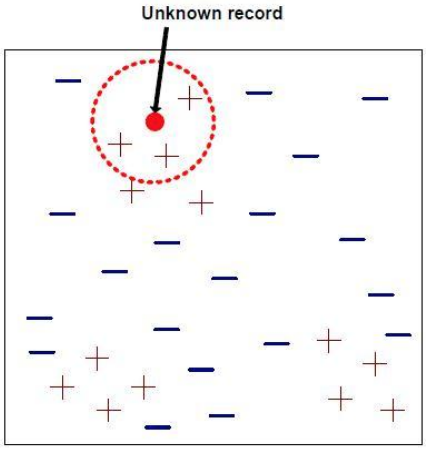


Fig 10: Classification of unknown source using kNN

For the project, we used kNN regression to determine the response time of a question. We used Matlab's method "*knnsearch*" to find the k-closest neighbors and then tried various ways to figure out the response time from its neighbors.

For kNN we need to decide on the k (the number of neighbors we are looking for) and the distance measure used.

Determining k for kNN:

Determining the right value of k is crucial, as too many or too few neighbors will increase the error due to bias variance trade-off. We used small set of training samples and used 70% of it for training and 30% for validation. We predicted the response time of the validation set using kNN-regression with varying value of k. We computed root mean square error (RMSE) for each value of k used. The RMSE were plotted against k to decide a good value of k. Basically, a good value of k is when the error doesn't change much when the value of k is changed. As we can observe from the plot in fig K.1.2, that a good value of k can be 25.

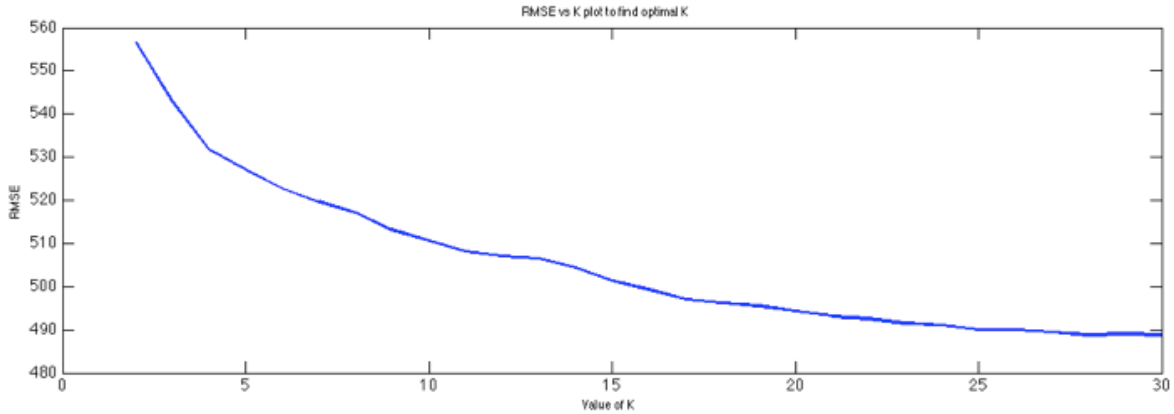


Fig 11: RMSE vs K plot

Determining Distance Measure for kNN:

Distance metric is also an important attribute to achieve good performance. Two points can be very close when distance measure, d_1 is used whereas can be far apart when distance measure, d_2 is used. We used a fixed value of k and used different distance measures to compute RMSE. euclidean gave fairly low error as compared to other distance metric like Mahalanobis, Minkowski and Manhattan.

Different Approaches:

We used different approaches to predict the response time of the test sample from k neighbors.
A. Average Time: In this approach, the predicted response time is computed by taking an average of the response time of the k-nearest neighbors.
B. SD Average Time: We observed that some neighbors for some test samples are outliners as compared to other neighbors of that test samples. To omit these outliners from the average we used standard deviation. So, we computed the mean and standard deviation, and considered only those neighbors that are standard deviation away from the mean. The results did not improve significantly.
C. Weighted Average Time: In this approach, we gave more weightage to neighbors that are closer to the test sample as compared to the other neighbors. The inverse of distance is used as the weight while computing the average response time. This technique also was not able to improve the results.
D. Max-Min Normalization: The features were max-min normalized so that all features have values between 0 and 1. Then KNN was applied on this normalized feature set. This improved the results a little bit so we used it for further experiments.
E. Principle Component Analysis (PCA): PCA is a dimensionality reduction algorithm, and was performed on the max-min normalized feature set to transform it into latent space. Most significant latent features covering maximum variance were then used for KNN-regression. Unexpectedly, it worsened the results. One of the possible reasons for this could be because the number of dimensions of the feature set was already less.

VII. EVALUATION METRICS

1. **Root Mean Square Error:** RMSE represents the sample deviation of the differences between predicted and observed values. The formula used to compute rmse is:

   RMSE = $\sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y)^2}{n}}$.

   where, y is the observed value and y cap is the predicted value.

2. **Confusion Matrix:** Confusion Matrix[16] is a tabular layout that enables easy visualization of the performance of a classifier. The below figure shows the layout of a confusion matrix. The cell named c11 (called true positives) is the count of data points which belong to Class A and are classified as Class A whereas c12 (referred as false negatives) is the number of data points classified as Class B but actually are of Class A. Similarly, c22 (called the false positives) is the number of data points classified as Class A but are of Class A and c22 (called true negatives) is the count of data points correctly predicted to be of Class B.

		Predicted Class	
		Class A	Class B
Actual Class	Class A	c_{11}	c_{12}
	Class B	c_{21}	c_{22}

Fig 12: Tabular representation of a 2x2 confusion matrix

The classification accuracy is defined as ratio of total number of data points correctly classified to the total number of data points.

Classification Accuracy (a) = (c11 + c22) / (c11 + c12 + c21 + c22)

MEAN ABSOLUTE ERROR:

The MAE measures accuracy for continuous variables. MAE is the average over the verification sample of the absolute values of the differences between forecast and the corresponding observation. The MAE is a linear score which means that all the individual differences are weighted equally in the average.

[16] http://en.wikipedia.org/wiki/Confusion_matrix

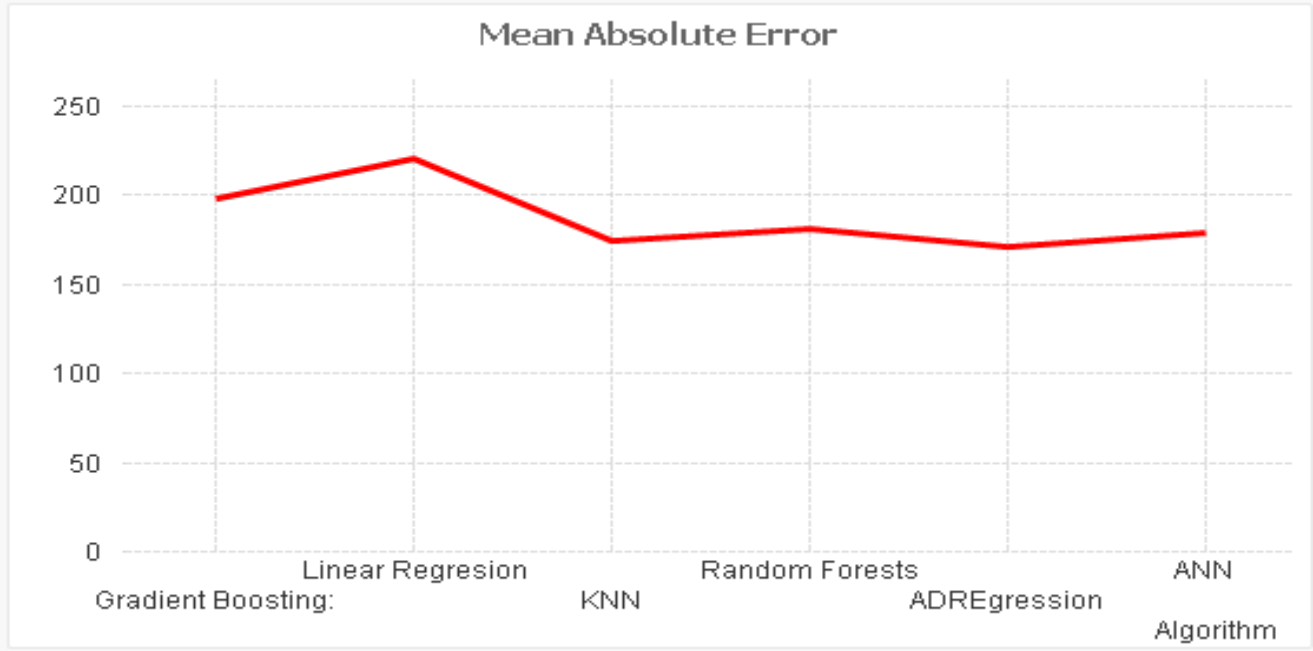


Fig E 1.0

The above graph depicts the Mean absolute error for different regression algorithms listed in section(**VI**)

RMSE:

RMSE measures the difference between forecast and corresponding observed values are each squared and then averaged over the sample. Finally, the square root of the average is taken. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE is most useful when large errors are particularly undesirable

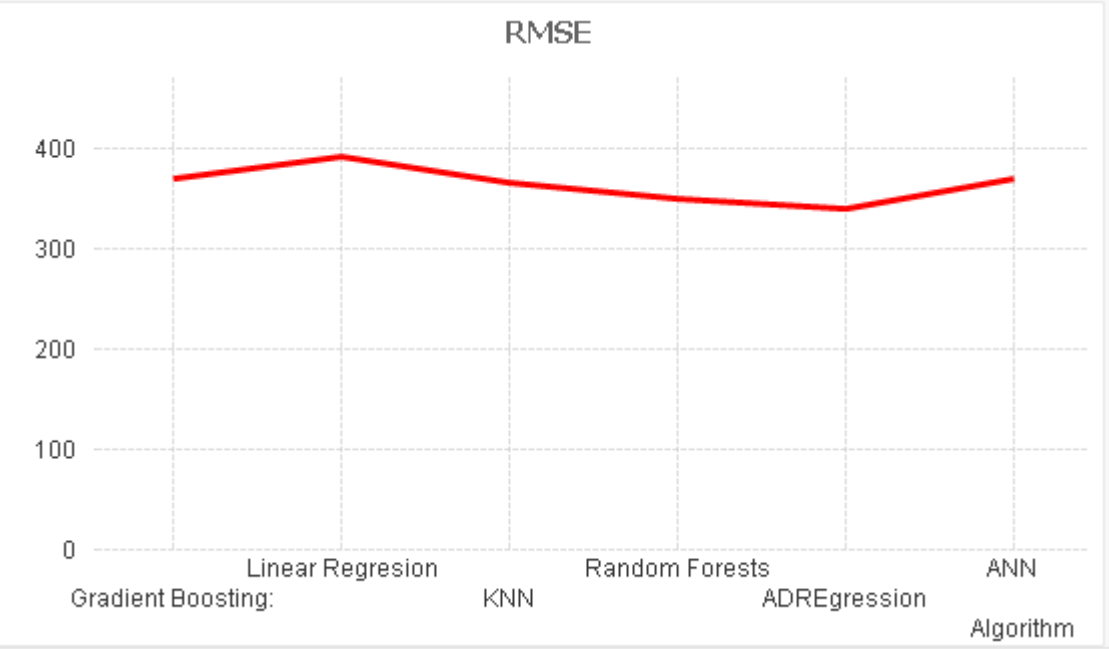


FIG E 1.1

The above graph depicts the Root mean square error for different regression algorithms listed in section(**VI**)

VIII. RESULTS:

The below graph depicts the accuracy values of different classification algorithms that are discussed above in section (**VI**). It is evident from the graph that Gradient Boost Classifier has outperformed all the other algorithms with an classification accuracy of 69.32 for all the features we have extracted listed in Feature Analysis Section(**IV**)

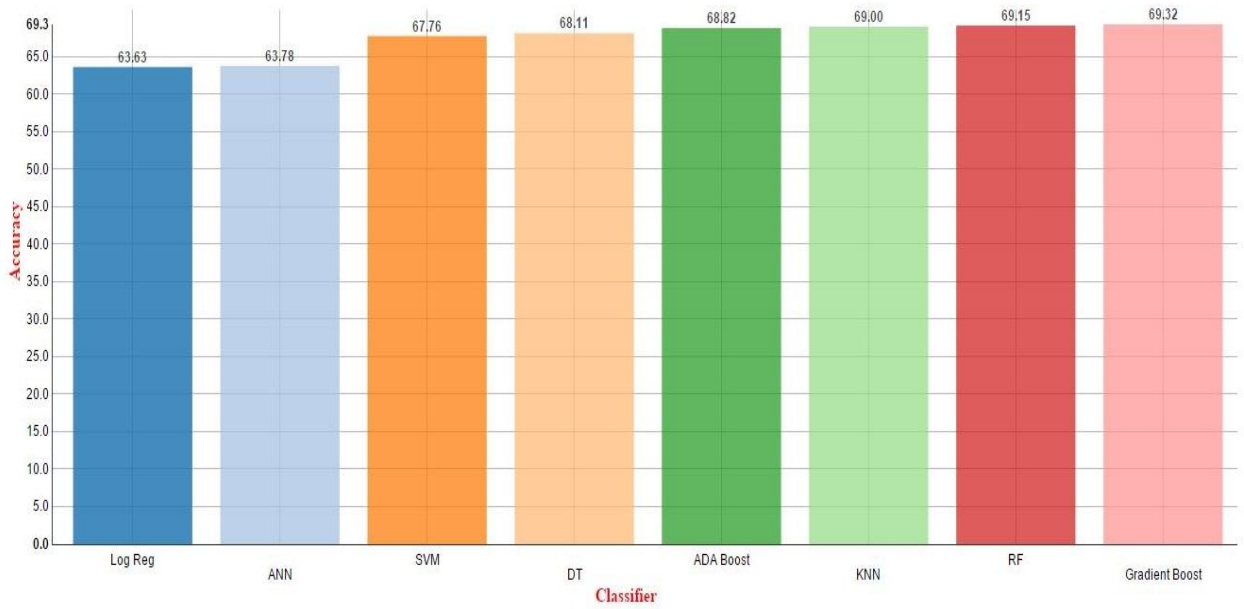
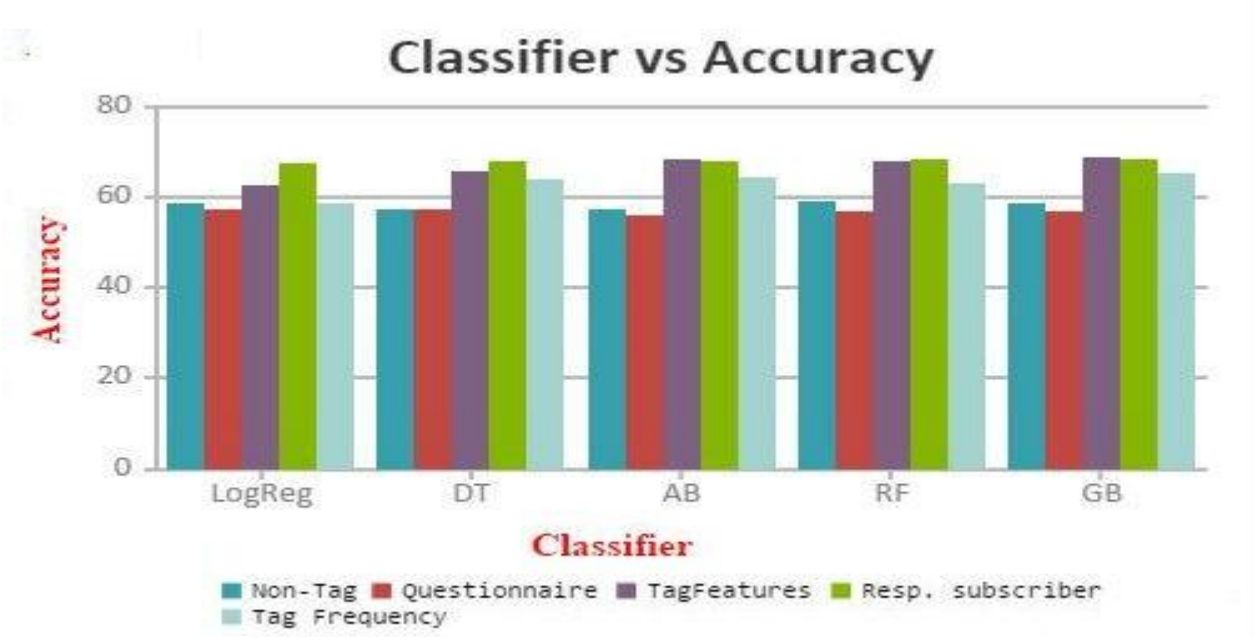


Fig R.1



Classification using Linear Discriminant Analysis:

For the entire feature set:
Classification Accuracy obtained: 68%

For tag based features:
Classification Accuracy obtained: 70%

For non tag based features:
Classification Accuracy obtained: 56%

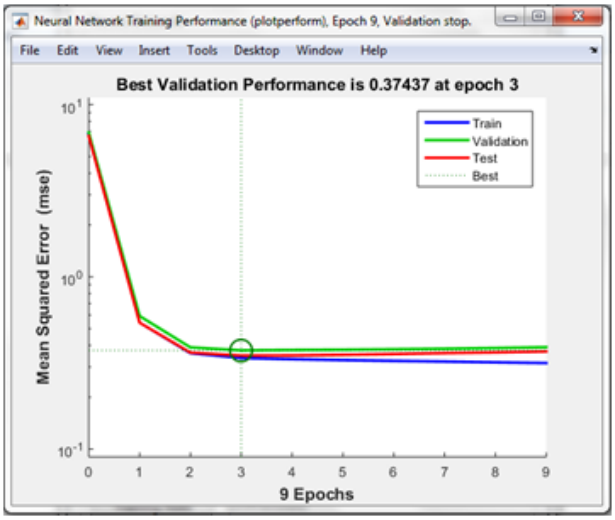
2) KNN Classification:

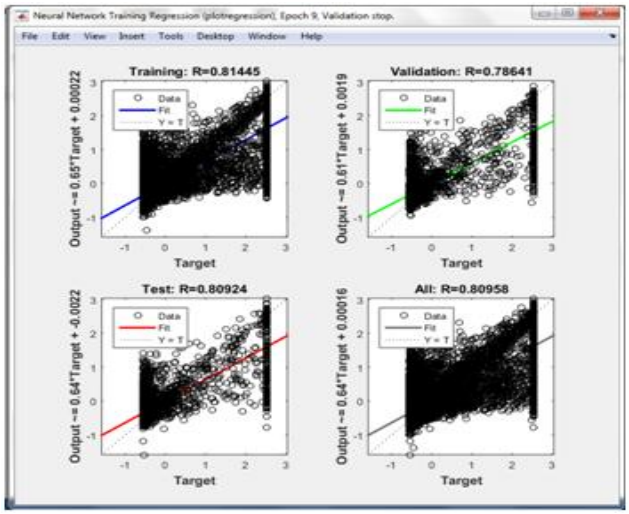
For the entire feature set:
Optimal K used: 12
Classification Accuracy obtained: 64%

For tag based features:
Optimal K used: 16
Classification Accuracy obtained: 69%

For non tag based features:
Optimal K used: 8
Classification Accuracy obtained: 54%

3) Classification Results using Artificial Neural Networks





The above plots show that

1. Test set Mean Squared Error of 0.37437 is achieved at third iteration.
2. Correlation of target vs output. R is the correlation coefficient of target vs output. R=1 is the ideal result. Note that R = 0.8 for testing.

IX. Conclusion

Stack Overflow is a very interesting website to perform mining. It has got huge heterogeneity in its data. When we explored the dataset we got many interesting results as discussed in Data and Feature analysis section of this report. Response time prediction is quite challenging, as there can be many different factors that can impact it. Though we are able to classify response time and

achieve accuracy of about 70 percent, we have to still analyze and come up with new features to improve our classification and regression accuracy.

X. Future Work

Gray areas**:** Areas that are not so popular in Stack Overflow now, but has future scope to become popular. We identified such areas in 2 different perceptive. One perceptive will be Gray areas are those areas (Tags in Stack Overflow terminology) which hasn't received any answers to the posts (zero Subscribers) that belong to a particular tag for the past 3 months. We have identified 2601 such tags which is almost 10 % (9.58 in precise) of the total number of tags. So if Stack Overflow can make its current users to answer those areas by rewarding them with higher bounty there is a possibility of increased user engagement. The second perceptive will be Gray areas are those areas which has certain threshold number of subscribers, but unanswered questions in those areas are comparatively high than answered questions. So these areas can be emphasized by Stack Overflow for increased user engagement. For identifying these areas we have taken subscribers for a tag threshold to 100 and unanswered to total posts ratio >0.5 and have identified 83 such areas.

In our future work we are also planning to add new features to increase the classification accuracy in classifying response time. Gray area can also be one of the feature. If the Tags columns of a particular posts contains gray areas, there is a high chance that the post will remain unanswered for more than a day. We are now analyzing the thresholds for identifying gray areas better.

INDIVIDUAL CONTRIBUTIONS:

Akarsh Cholaveti: System Analysis, Feature Extraction, Non-Tag Features, Random Forest Regression, Metrics calculation, Ensemble classification, Report generation.
Vamsi Deepak Darbha: ETL, System analysis, Data Analysis, Feature Extraction, Data Preprocessing, SVM classification, Report generation.
Vineel Vutukuri : System analysis, Feature Extraction(User Features), Feature Analysis, AdaBoost classification/Regression Report generation.
Srikanth Vemulakonda : System Analysis, ANN classification/Regression, Report generation.
Prerna Satija : System Analysis, KNN classification, KNN regression, Report generation, Discriminant Analysis.
Surbhi Aggrwal: System Analysis, Linear Regression.

XI. References:

[1] Vasudev Bhat Adheesh Gokhale Ravi Jadhav Jagat Pudipeddi Leman Akoglu : Min(e)d Your Tags: Analysis of Question Response Time in StackOverflow

[2] YuanYao,HanghangTong,FengXu,JianLu:Predicting longterm impact of CQAposts:a comprehensiveviewpoint.KDD2014:1496--- 1505

[3] D. Movshovitz-Attias, Y. Movshovitz-Attias, P. Steenkiste, and C. Faloutsos, "Analysis of the reputation system and user contributions on a question answering website: Stackoverflow." in ASONAM, 2013,pp. 886–893.