ENS 491-492 – Graduation Project Final Report

Project Title:

Posted your question to Q&A community websites.
Now what? Just wait?
Towards developing the next step...

Group Members:

23883 Mehmet Utku Eray
20510 Zeynep Seda Birinci
24142 Deniz Bozkurt
23914 Selen Özcan

Supervisor: Reyyan Yeniterzi

2020 May

Sabancı Üniversitesi

Contents

1 EXECUTIVE SUMMARY

The web has changed how individuals provide, search and offer information. However, the searched results may not provide a precise answer to the user's search and it might be tedious to review every single one of them, without having the assurance of finding the ideal answer.

Community Question Answering websites are built to provide the searched information more efficiently by utilizing an interface for users to exchange and share knowledge. Stack Overflow is one of the most popular Q&A (Question and Answer) community websites to find answers related to pretty much every aspect of coding. Q&A community websites are widely used to provide information to users about their questions.

In this project there are 4 machine learning tasks to improve Stack Overflow site in terms of user experience and these improvements are implemented by a user friendly web-site.

First task is, If an inquirer asks a question on the Q&A community websites, predicting whether the inquirer would receive an answer to her/his question or not. Second is, If an inquirer asks a question on the Q&A community websites, predicting the expected time for the question to get answered. Third is, If an inquirer asks a question on the Q&A community websites, predicting the context based on similar questions asked on Q&A community websites. Last task is from responders point of view, providing information about whose questions they can answer. Front-end objective of this project is to develop a web-site implemented with Javascript, html, css and Python in order to demonstrate the improvement on a user interface. Django and bootstrap libraries were used to improve web-site.

2 PROBLEM STATEMENT

Although Q&A community websites like Stack Overflow has become popular, there are many things that can be considered as problems. For a start, one of the most common problems is that some questions do not get an answer even though there are many users actively using the platforms. According to our research, there are 19 million questions asked so far and only 70% of which has been answered even though there are 11 million users on this platform. (“Stack exchange”, 2019) Another issue is that, there is an uncertainty regarding when the questions will receive an answer. From our research we have seen that, there are some questions yet to be answered which are several months and even several years old. Lastly, we want users of our system to be able to answer questions that are more related to them. Q&A community websites such as Stack Overflow does this by suggesting questions based on tags users choose.

Up until now, there have been several researches focused on problems mentioned above to come up with solutions regarding the issues. One of them deeply investigates the issue of unanswered questions, proposing a method to predict how long a question will remain unanswered and when it will receive an answer by considering several criteria such as inquirer’s credibility, topic of the question etc and building a simple classification model. (Asaduzzaman, Mashiyat, Roy, Schneider, 2013) However it doesn’t answer the other questions raised in our paper about the correctness of the answers and the model suggested in the paper is not developed well enough for practical uses. Therefore, further work needs to be conducted to introduce a well-built solution.

Another research solely focuses on how to select the best answer for a particular question by taking both question-answer and answerer’s data into account to propose a model. (Nagwani, Sahu, Verma, 2016) The research’s aim is to find the best answer for the inquirer to choose among all the answers provided for that specific question, not to find the most correct or informative answer, therefore it satisfies inquirer’s needs but it may not be enough to come up with a scientifically accurate answer. Hence, the problem of finding the most relevant answer still remains unsolved.

Lastly, according to Analyzing Bias in CQA-based Expert Finding Test Sets , Yeniterzi & Callan states that “ ... two potential sources of bias in test set construction. The first one is a temporal bias caused by voting activities which are performed before receiving all answers. Another similar bias is the presentation bias which is caused by the initial ranking of the replies, when users only look at the top ranked replies, vote for one, ignore the rest of the replies and leave the page.” (2014, p.1) So, to minimize the risk of bias it is planed to analyze the users’ profile about their participation on other questions asked before to see if there is any similarity between the question and the history of the user that answered the question. This might give a clue about the correctness of the answer to the inquirer and we can return a percentage value on every answer given to the question, for the inquirer to evaluate and choose amongst the answers rather than ending up with a single best answer. However this approach is not being used in this project for reasons that will be given in the following sections.

2.1 Objectives/Tasks

2.1.1 Inquirer's point of view: "Will my questions get answered?"

In this task the goal is to predict whether the question will get an answer or not, at the time of the questions being asked. It has been planned to resolve this task by constructing a supervised learning model. To use text formatted data, questions and titles of the questions, as vectors Doc2Vec method has been used from gensim library. To make a prediction on answers, Random Forest Classification model has been used.

2.1.2 Inquirer's point of view: "When will my question get answered?"

In this task the goal is to predict when the question will get an answer by calculating how much time is expected to elapse after posting a question on the platform. Our approach focuses on constructing a supervised learning model based on classification models. In order to built the classification model, response time data divided into classes. Text format data, questions and titles of questions, represented as vectors by the help of Doc2Vec method from gensim library.

2.1.3 Inquirer's point of view: "Are there any similar questions asked on Stack Exchange Q&A community websites?"

Finding similar questions to one's own that has received a valid answer is often as effective as receiving an answer to one's own question. The aim of this task is to provide the inquirer with the most similar questions to their own question and let the inquirer interpret the answers in order to find a solution to their own problem. Stack Exchange utilizes tags to filter out similar questions while the approach taken in this task is to generate a model with Doc2Vec and find out most similar questions by training the model with titles,bodies and tags of questions and finally inferring a vector for each question from the generated model. Lastly, the cosine similarity of the vectors are compared in order to find out the most similar questions.

2.1.4 Responder's point of view: "Which questions am I more likely to answer correctly?"

This task is based on responders' expectations. The information about whose questions they can answer will be provided. To store data as vectors Doc2Vec method will be used from gensim library.The task consists of three parts.First part is suggesting the responder questions based on their profile. For each responder, the questions that they have answered will be stored as one vector, the other vector will be questions vector that are not answered yet. After storing the data as vectors, these vectors will be compared according to their cosine similarities. And most related ones will be provided to the responder.For the second part, each question that has been answered by the responder so far will be compared with the questions in the data set and the most similar ones will be provided to the responder for answering it. Regarding the final part, in the data sets there are tags on the questions which represent their related fields. These tags will be used to rank the related questions for each responder.

2.2 Realistic Constraints

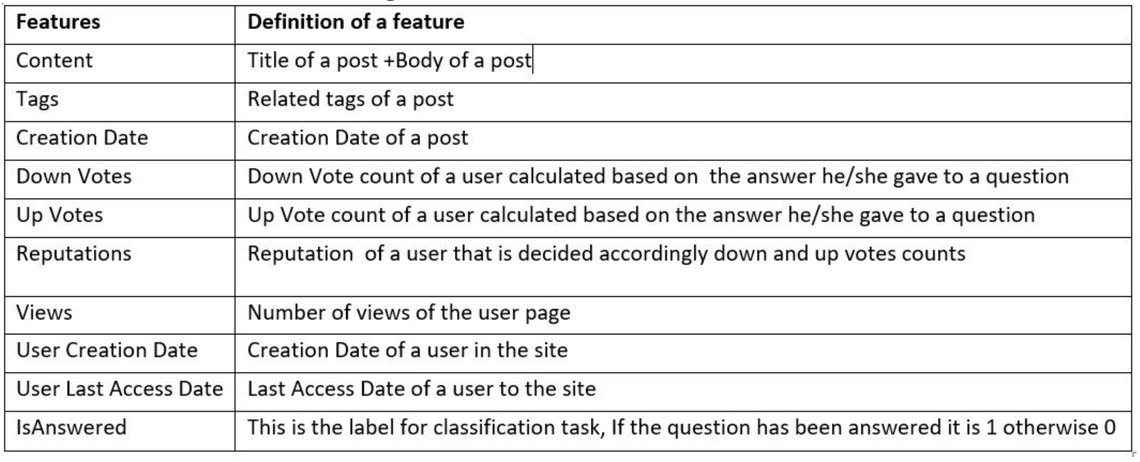
There are several realistic constraints in this project. These are Economic, Manufacturability and Sustainability. There is no budget for this project but are some costs throughout the development. Computers that will be used while developing and server costs are the main parts for the economic constraint. Since the project will be a web based program with machine learning aspect, development will be cheap but it will require user to have a computer or a smartphone to be able to use the application. Since users have to use a computer or a smartphone to be able to ask questions on Q&A community websites, there shouldn't be any problems for them to use this application which solves the manufacturability. constraint. Finally to keep sustainable, there will be constant updates on the application and will keep training the algorithm for machine learning to deliver the best possible user experience. After making the application fully dynamic and automated with an API for Q&A community websites, the system should be self sufficient and no need to update the application besides better user interface or new features.

3 METHODOLOGY

3.1 Implementation of “Will my question get answered?”

For this task, it was decided to merge 2 data sets (posts and user data) of ai.stackexchange to get better results. These 2 data sets are merged by their joint feature (user id). Intersection of User Id’s was found. New data frame was created according to this intersection. Features that are used in final data set are shown in the Figure 1 below:

Figure 1: Definition of features

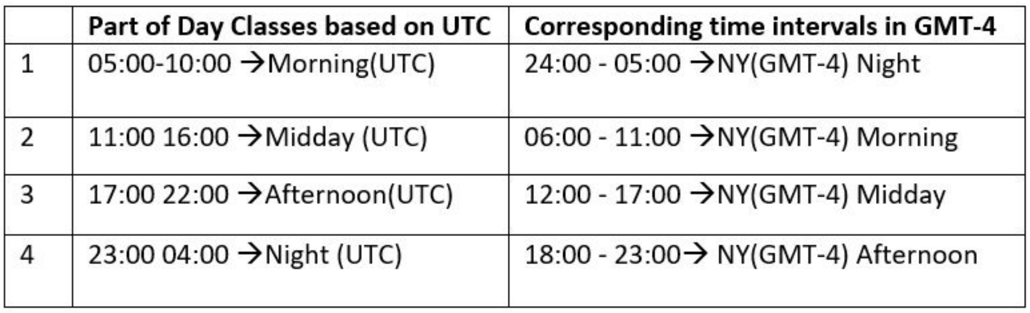


Features	Definition of a feature
Content	Title of a post +Body of a post
Tags	Related tags of a post
Creation Date	Creation Date of a post
Down Votes	Down Vote count of a user calculated based on the answer he/she gave to a question
Up Votes	Up Vote count of a user calculated based on the answer he/she gave to a question
Reputations	Reputation of a user that is decided accordingly down and up votes counts
Views	Number of views of the user page
User Creation Date	Creation Date of a user in the site
User Last Access Date	Last Access Date of a user to the site
IsAnswered	This is the label for classification task, If the question has been answered it is 1 otherwise 0

It was observed that the data set was not evenly distributed according to the answered/ not answered questions. It resulted in failure of the prediction at first try of machine learning model. So the data set was shuffled properly with respect to answered/ not answered questions.

Many machine learning algorithms cannot operate on labeled data directly. They require all input variables and output variables to be numeric. For this reason, date formatted features were converted to numeric values in order to be used in the model. Firstly, label encoding was used to convert categorical features into numerical values. The date data of Stack Exchange is stored according to Coordinated Universal Time (UTC) Greenwich. Four time intervals were defined to divide a day and encode numerically. Time intervals are shown in the Figure 2 below.

Figure 2: Part of the Day Classes



	Part of Day Classes based on UTC	Corresponding time intervals in GMT-4
1	05:00-10:00 →Morning(UTC)	24:00 - 05:00 →NY(GMT-4) Night
2	11:00 16:00 →Midday (UTC)	06:00 - 11:00 →NY(GMT-4) Morning
3	17:00 22:00 →Afternoon(UTC)	12:00 - 17:00 →NY(GMT-4) Midday
4	23:00 04:00 →Night (UTC)	18:00 - 23:00→ NY(GMT-4) Afternoon

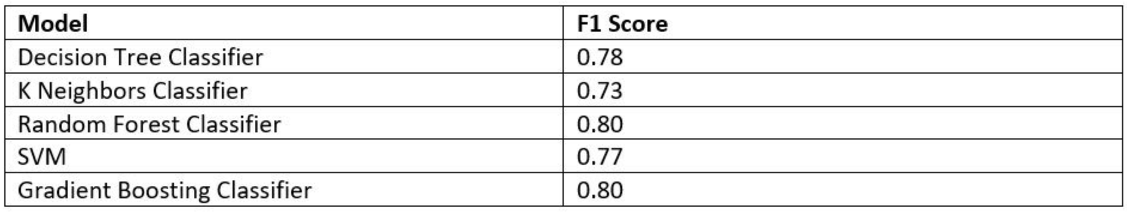
After categorizing the date format features with label encoding, one-hot encoding was implemented to allow efficient use of these features in the final training data set. For each date type feature this process was repeated. For this machine learning task, information of creation date of a post, creation date of a user and last access date of a user was processed.

There are 2 text format features that will be used in this machine learning task. One of them is tags data of posts, the other one is union of title and body of posts. These 2 cases were processed solely. For tags data of posts, CountVectorizer method was used. The method is from scikit-learn library and offers a simple way of both tokenizing a set of text documents and creating a vocabulary of recognized words but also encoding new documents using that vocabulary. The encoded vector is returned with the length of the entire vocabulary and the total number of times each word appears in the document. After building vectors, original tags feature has been omitted and new corresponding vectors has been added to data frame for each tags. (See APPENDIX A for the data frame format for Tags feature's before and after this operation)

In order to prepare text data (title + body of questions) of posts, Doc2Vec method from gensim library was used. NLTK library has been used in order to delete stop words and further preprocessing of text data. Vector representations with Doc2Vec were created for each individual post. For this specific problem optimal vector size was chosen as "100". After creating vectors for each post, each vector has been added to their corresponding post's row. Thus, new data frame has appended 100 new columns for each post. (See APPENDIX B for the data frame format for Content feature's before and after this operation)

After all these operations, training data set has been finalized. For the test data set, same process has been implemented. Later, classification methods have been tried for prediction on answer. Table shown below indicates the F1 score of each classifier have been used. According to Figure 3, Random Forest Classifier and Gradient Boosting Classifier have the highest F1 score. Since, Random Forest Classifier was more consistent in each test run, model constructed with Random Forest Classifier.

Figure 3: F1 score comparison of Classifiers on test set

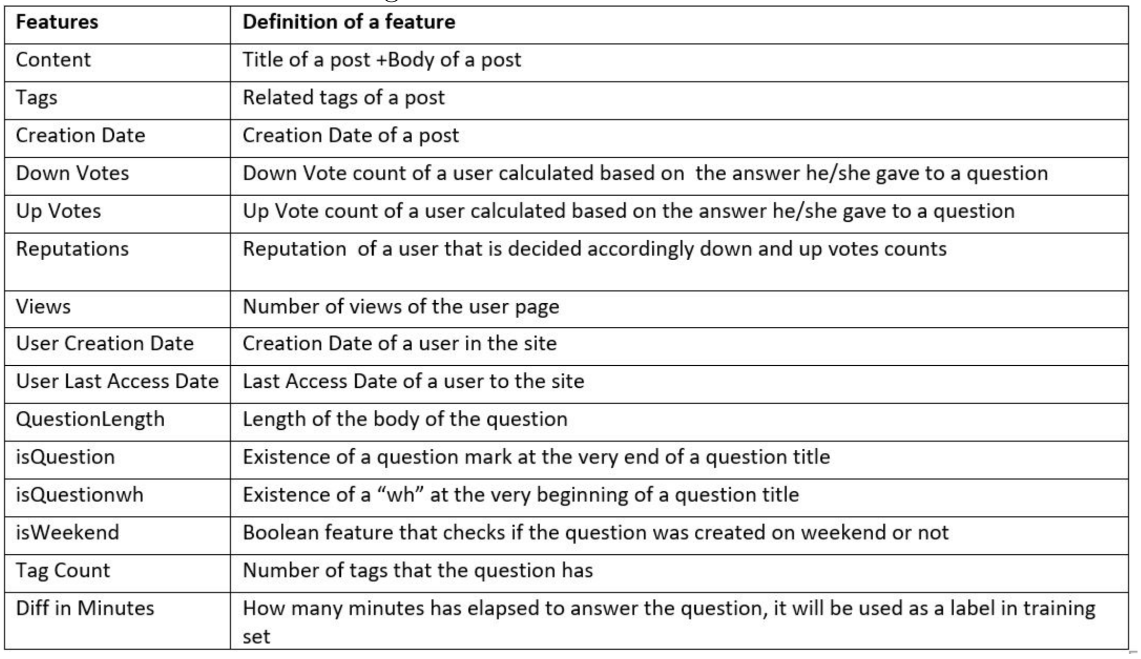


Model	F1 Score
Decision Tree Classifier	0.78
K Neighbors Classifier	0.73
Random Forest Classifier	0.80
SVM	0.77
Gradient Boosting Classifier	0.80

3.2 Implementation of "When will my question get answered?"

For this task, posts and user data sets from ai.stackexchange data are used and they are merged by their joint feature (user id). Features that are used in final data sets are shown in the Figure 4 below:

Figure 4: Definition of features



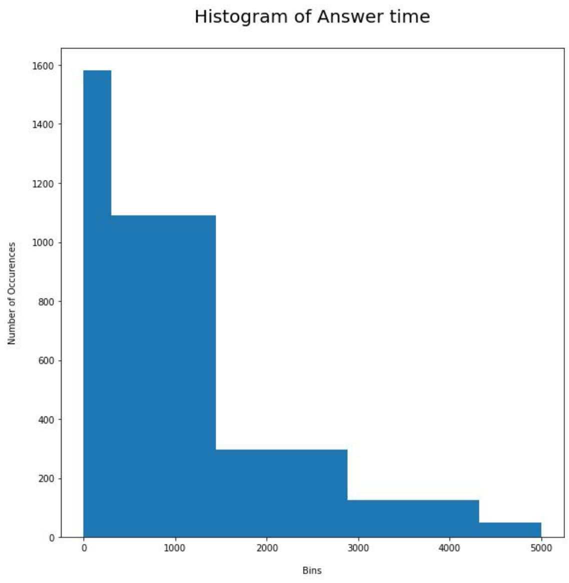
Features	Definition of a feature
Content	Title of a post +Body of a post
Tags	Related tags of a post
Creation Date	Creation Date of a post
Down Votes	Down Vote count of a user calculated based on the answer he/she gave to a question
Up Votes	Up Vote count of a user calculated based on the answer he/she gave to a question
Reputations	Reputation of a user that is decided accordingly down and up votes counts
Views	Number of views of the user page
User Creation Date	Creation Date of a user in the site
User Last Access Date	Last Access Date of a user to the site
QuestionLength	Length of the body of the question
isQuestion	Existence of a question mark at the very end of a question title
isQuestionwh	Existence of a "wh" at the very beginning of a question title
isWeekend	Boolean feature that checks if the question was created on weekend or not
Tag Count	Number of tags that the question has
Diff in Minutes	How many minutes has elapsed to answer the question, it will be used as a label in training set

For the date format features, same preprocessing has been used with the first task. First, label encoding has been done and categorized data into 4 parts which indicates different parts of day then one-hot encoding implemented. Also same text preprocessing and classification methods have been used with the first task in order to use text format data. For Tags feature Countvectorizer method was used. Title and body of the questions were merged and created Content feature. Doc2Vec method was implemented on Content data in order to represent text data as vectors.

Outliers in response time in minutes have been omitted in order to make a better prediction on answer time. Interquartile Range Rule was used to find outliers. After this operation, maximum time to answer a question was determined as 4933 minutes $\approx$ 82 hours $\approx$ 3.5 days.

Later, regression methods have been tried for prediction on answer time. Since, prediction was not credible (mean square error was too large). It has been decided to turn this regression task into classification task. For this transformation response times were categorized as first 5 hours, 5 hours-1 day, 1-2 days, 2-3 days and 3-4 days. Distribution of response time histogram is shown in Figure 5 below:

Figure 5: Response time distributions (first 5 hours, 5 hours-1 day, 1-2 days, 2-3 days, 3-4 days)



According to histogram, it is clear that the response time data is quite imbalanced. To balance the response time data, imblearn library was used and 2 methods have been tried: Random over sampling and ADASYN (Adaptive synthetic sampling approach for imbalanced learning) over sampling. Random oversampling simply replicates randomly the minority class examples. ADASYN generates samples of the minority class according to their density distributions.

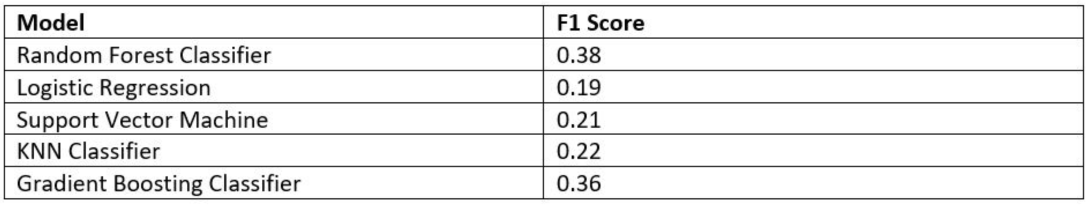
Data distribution of response time after oversampling with different methods are shown in the Figure 6 below.

Figure 6: Data Distribution on test data set after Oversampling

Over Sampling Method	**1st class (0-5hours)**	**2nd class (5 hours-1 day)**	**3rd class (1-2 days)**	**4th class (2-3 days)**	**5th class (3-4 days)**
Random Over Sampling	347	347	347	347	347
ADASYN	347	375	347	351	350

After the data became more balanced, several Classifiers have been implemented on categorized data set. Classifiers and their corresponding overall F1 scores are shown in the table. Random forest Classifier has the highest F1 score overall, so the model was constructed with this Classifier. See Figure 7.

Figure 7: F1 score comparisons of models that have been implemented



Model	F1 Score
Random Forest Classifier	0.38
Logistic Regression	0.19
Support Vector Machine	0.21
KNN Classifier	0.22
Gradient Boosting Classifier	0.36

3.3 Implementation of "Are there any similar questions asked on Stack Exchange Q&A community websites?"

Methods used for this task is to download and load the data dump generated by Stack Exchange for their Q&A websites and generate a xml tree. This xml tree is stored in to a pandas dataframe then this dataframe's unnecessary columns are discarded and left the dataframe with only Title, Body, Tags and Date columns. This smaller dataframe then split in to two sets for training and testing purposes. After generating two corpuses for training and testing, training corpus is used for training the Doc2Vec model. After using test corpus for testing the model, there have been several adjustments on different parameters for the Doc2Vec model to obtain best possible results and by training the model with titles, bodies and tags of questions, model returns most similar questions depending on the inferred vector. There is also another model generated with TFIDF with the same approach and comparing two models resulted in different similarities for same test corpus. This difference favored in Doc2Vec model and the reason for this is discussed in Results and Discussion section. Since the analysis of both models showed Doc2Vec is better for the project purpose, Doc2Vec model has been implemented to complete this task.

3.4 Implementation of "Which questions am I more likely to answer correctly?"

This task is based on responders' expectations. The methodology used for this task is similar to the one used "Which questions am I more likely to answer correctly?" task. Since the model generated in the previous task is the best case scenario for this data set with Doc2Vec, there wasn't any need for further analysis. There are three parts to this task, which are suggesting responder's questions based on their overall profile, displaying similar questions individually for each question that the responder have answered and finally displaying questions based on responder's preferred tags. There are three different models used to solve this task but each of the models are similar in the way they work. Both suggesting questions based on responder's profile and suggesting questions based on each question that the responder has answered utilizes the same model. Regarding searching questions based on preferred tags,this part differentiates from other models with the way it filters questions. This model displays questions by generating results using the responder's profile just like the other option and it filters out the questions that do not have the preferred tags.

3.5 Implementation of Front-End

Implementation of the front-end switched to utilizing Django 3, which is a Python framework that enables to develop a server and create the website as a python web app. Each page of the web application can be generated through template views written in Python and the data transfer between views can be done by passing parameters through URLs which is a more common approach than the other methods. Authentication is done via Stack exchange API which is a free service and necessary one for our system to work properly. After authentication phase done, user will be redirected to profile page where the users' data from Stack Overflow is displayed. User interface consists of three main parts, there is the homepage, authentication/login page and pages after logged in. There is a profile page which displays the account type, reputation, asked and answered questions by the user. When a user clicks on the asked or answered questions by themselves, it forwards the user to the original Stack Overflow page of the question. There are also two more pages named Ask Questions and Answer Question which is not finished yet.

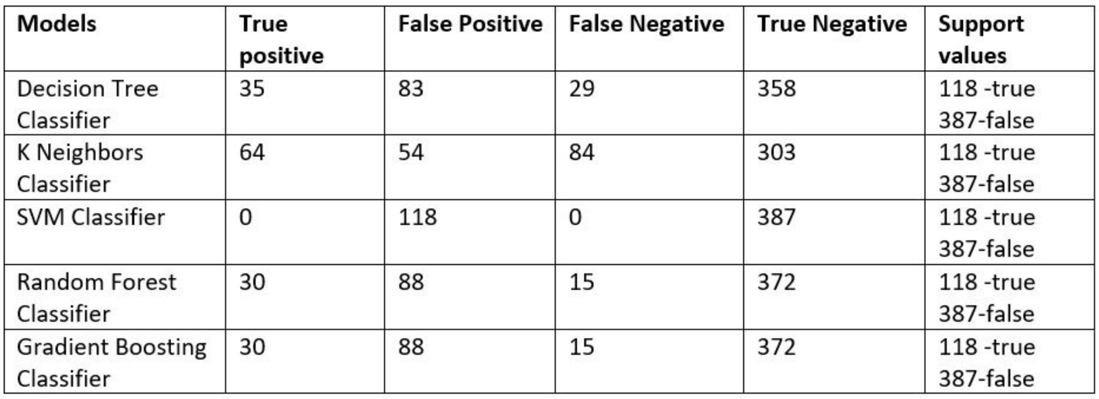
Furthermore, a markdown text editor and also a preview for the user is implemented. Question can be submitted to inferred by the model and passed to results page which displays similar questions with verified answers, unverified answers and without answers to the question asked by the user. The answers are sorted with respect to the cosine similarity between the question asked by the user and the relevant questions in the database. See APPENDIX C.

In conclusion, for the optimal user experience, the front-end has been prepared with Django 3 framework using Python3 for server-side and paging, Bootstrap4 framework using HTML5 and CSS for designing user interface and finally JavaScript and jQuery for functionalities of user interface and data transferring between views.

4 RESULTS & DISCUSSION

"Will my question get answered?" task has been completed. Our model is capable of predicting whether the question will be answered or not. In classification methods the highest accuracy score has been obtained from Random Forest Classifier with 80% accuracy. This task has also been implemented on the web site. The confusion matrix of each method that has been tried, demonstrated in the Figure 8 below:

Figure 8: Confusion Matrix table of each model

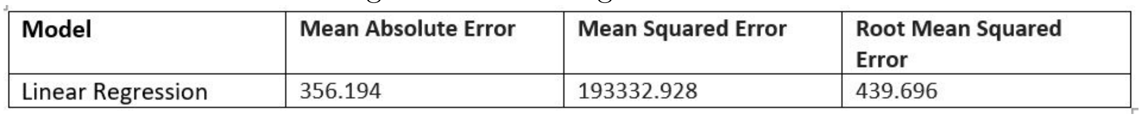


Models	True positive	False Positive	False Negative	True Negative	Support values
Decision Tree Classifier	35	83	29	358	118 -true 387-false
K Neighbors Classifier	64	54	84	303	118 -true 387-false
SVM Classifier	0	118	0	387	118 -true 387-false
Random Forest Classifier	30	88	15	372	118 -true 387-false
Gradient Boosting Classifier	30	88	15	372	118 -true 387-false

True values (0) indicates the questions that are predicted to be not answered. False values (1) indicates the questions that are predicted to be answered. True positive values indicates the number of questions that are predicted to be not answered correctly. True Negative values indicates the number of questions that are predicted to be answered correctly. According to the table K-neighbors Classifier is the best model to predict questions that will not be answered. However, its True Negative value is the smallest. So, K-neighbors Classifier is not the best model overall. Decision Tree Classifier, has the second highest number of True positives but its False Negative value is higher compared to Random Forest, SVM and Gradient Boosting Classifiers. Which indicates it is not the best model to predict the questions to be answered. The best models overall are chosen to be Gradient Boosting Classifier and Random Forest Classifier. Since Random Forest Classifier was more consistent in multiple test runs, it was chosen for the given task.

"When will my question get answered?" task has been finalized. The task has been identified as a regression task at the beginning. However, mean square error of the regression model was quite high to predict significantly. The error results of linear regression model are shown in the Figure 9 below:

Figure 9: Linear regression model



Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Linear Regression	356.194	193332.928	439.696

In the Figure 9, error results are based on predictions in minutes. Mean absolute error is equal to "observed value - real value on average" and it is equal to 356.194 which is $\approx$ 6 hours. According to error results, regression model is not satisfiable to make precise predictions.

Thus, the problem has been transformed into a classification task. Firstly, Random Forest Classifier has been implemented to predict the categorized response times. It predicted the response time categories with 83% accuracy. However, there was denseness on the response time's 0-1 day category. And the model was predicting all the questions' response times in that category. Thus, 0-1 day response time category divided into 0-5 hours and 5 hours-1 day in order to make more precise predictions. Comparison between differently categorized response time distribution histograms are shown in Figure 10 and Figure 11 below.

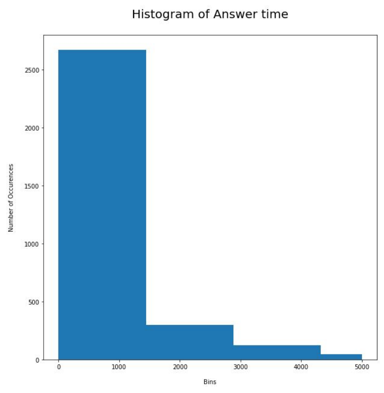


Figure 10: Response time distribution when data is classified as 0-1 day, 1-2 days, 2-3 days, 3-4 days

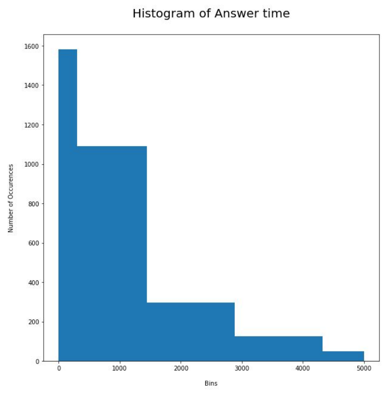
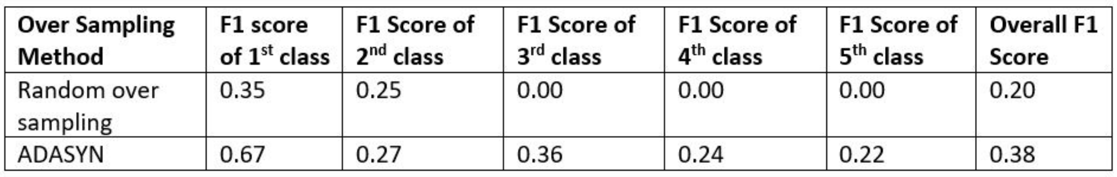


Figure 11: Response time distribution when 0-1 day class divided into 0-5 hours and 5 hours-1 day

As it can be seen from response time distribution histograms, the data is still imbalance. ADASYN oversampling method was chosen to balance response time since it resulted higher F1 score on classification model. Comparison of results between Random oversampling and ADASYN oversampling on Random Forest Classifier is shown in the Figure 12 below:

Figure 12: Comparison of Random over Sampling and ADASYN over sampling

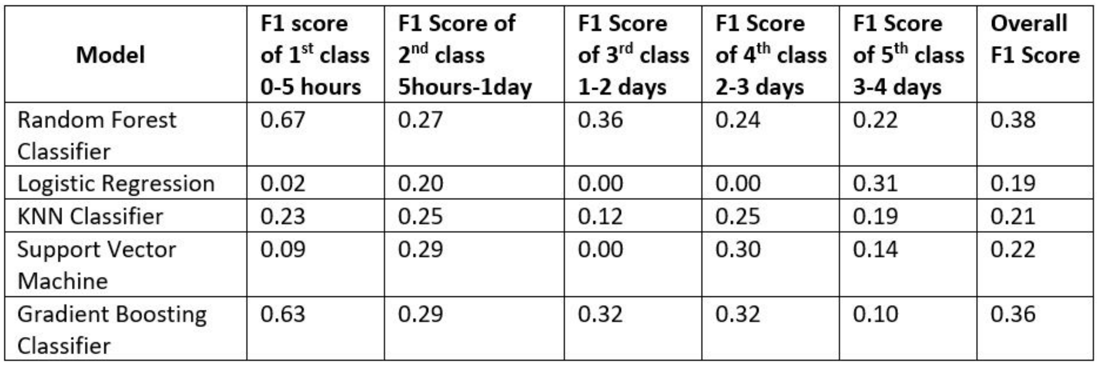


Over Sampling Method	F1 score of 1st class	F1 Score of 2nd class	F1 Score of 3rd class	F1 Score of 4th class	F1 Score of 5th class	Overall F1 Score
Random over sampling	0.35	0.25	0.00	0.00	0.00	0.20
ADASYN	0.67	0.27	0.36	0.24	0.22	0.38

In Figure 12, from the Classification report it can be said that the model constructed with random over sampling fails to predict 3th (1-2 days), 4th (2-3 days) and 5th (3-4 days) classes, since their corresponding F1 scores are 0.00. Also the model constructed with random over sampling makes predictions on 1st and 2nd class with lower F1 scores compared to the results on ADASYN over sampling method. Thus, it can be concluded that ADASYN over sampling method is a better balancing technique for this task.

After, choosing the best balancing technique, other classifiers have been tried. Classification report of each of them are shown in the Figure 13 below:

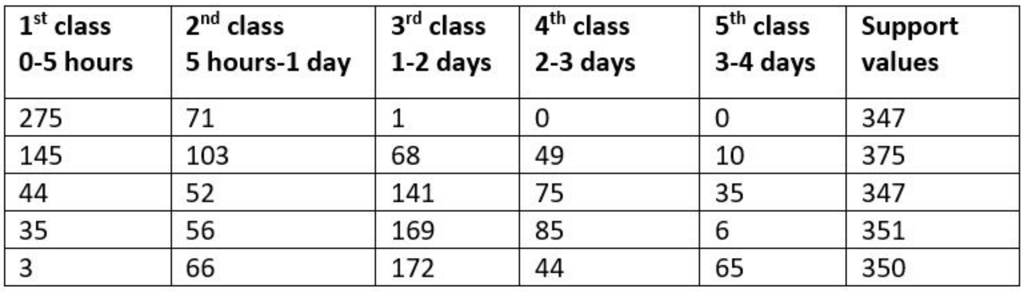
Figure 13: Comparison of different Classifiers based on their F1 scores



Model	F1 score of 1st class 0-5 hours	F1 Score of 2nd class 5hours-1day	F1 Score of 3rd class 1-2 days	F1 Score of 4th class 2-3 days	F1 Score of 5th class 3-4 days	Overall F1 Score
Random Forest Classifier	0.67	0.27	0.36	0.24	0.22	0.38
Logistic Regression	0.02	0.20	0.00	0.00	0.31	0.19
KNN Classifier	0.23	0.25	0.12	0.25	0.19	0.21
Support Vector Machine	0.09	0.29	0.00	0.30	0.14	0.22
Gradient Boosting Classifier	0.63	0.29	0.32	0.32	0.10	0.36

According to the Figure 13, the model that predicts the first class (0-5 hours response time) best is Random Forest classifier with 0.67. However, Support Vector Machine and Gradient Boosting Classifier predicts the second class (5hours to 1 day) better. Nevertheless, overall Random Forest Classifier has more consistent F1 score results. Since it is capable of predicting each classes and has the highest overall F1 score. Thus, final model has been constructed with Random Forest Classifier on the data that had been over sampled with ADASYN method. The model still can be improved since F1 scores of each class and overall F1 score is low. The table shown below in the Figure 14 is the confusion matrix of the final model of this task.

Figure 14: Confusion matrix of the final model on ai.stackexchange data test set



1st class 0-5 hours	2nd class 5 hours-1 day	3rd class 1-2 days	4th class 2-3 days	5th class 3-4 days	Support values
275	71	1	0	0	347
145	103	68	49	10	375
44	52	141	75	35	347
35	56	169	85	6	351
3	66	172	44	65	350

Support values indicates that the number of values in the classes in ascending order. According to confusion matrix in Figure 14 there are 347 values in the first 5 hours, 275 of them predicted correctly, 71 of them predicted in 2nd class, 1 of them predicted in the 3rd class. On the other hand, prediction of the 5th class (3-4 days) is not good as the prediction of 1st class(0-5 hours). There are 350 values in 5th class, 65 of them predicted correctly, 44 of them predicted in the 4th class (2-3 days), 172 of them predicted in the 3rd class (1-2 days) which indicates our model is not capable of predicting the 5th class well, 66 of them predicted in 2nd class and 3 of them predicted in 1st class (0-5 hours), which indicates that our model is capable of separating 1st and 5th classes.

One of the tasks that was written in the progress report which is “Inquirer’s point of view: If I receive multiple answers to my question, which are/is better solution/s?” has been changed. The changed task was a feature for the inquirer which determined the better answer for his/her question by comparing the answers for the question. The methodology was to first vectorize the answers to the question and create a user answer vector from the previous answers given by the users that have answered the inquirer’s question. Finally the answer vectors were supposed to be compared to the users’ answer in order to predict the better answers amongst the answers given by the users. This task was replaced by finding similar questions on the website task which was referred to in Tasks section and Methodology section.

“Are there any similar questions asked on Stack Exchange Q&A community websites?” task has been completed. Our model is capable of finding the questions that are similar to the given question. Throughout the development, there has been two methods used for this task and these are Doc2Vec and TFIDF. These two machine learning models resulted in different similarity results for same training and test sets. On average, Doc2Vec performed better since its results matched the related questions listed by the Stack Exchange itself more accurately and context-wise the resulting questions were more similar. Therefore, this task is accomplished by utilizing Doc2Vec.

“Which questions am I more likely to answer correctly?” task has been completed. This task required the same methodology in the “Are there any similar questions asked on Stack Exchange Q&A community websites?” task. Therefore, same Doc2Vec model is utilized for suggesting questions that can be answered by the user. This time model utilizes every question that the user answered and the tags that user select, and the model returns top ten similar results for user to see and if possible answer.

5 IMPACT

Q&A community websites are widely used by many users regardless of the experience they grant due to huge flow of information they provide. The target of these communities is to make the knowledge sharing and exchanging processes for users easier under large number of categories. All inquirers who post questions in Q&A community websites expect to receive an answer in short periods of time and are frustrated when their questions remain unanswered. Unfortunately, there are a lot of existing issues that make this process extremely difficult; these problems are expressed clearly in previous sections. Our project proposes several models to solve these issues that would make the usability of Q&A community websites convenient, allowing every single type of user to overcome the frustration and concern of not receiving an answer and provide ways to improve their posts. Responders will be more engaged to share their knowledge while being able to access questions that appeal their interests easily. Both sides would be able to use the platforms at ease, while contributing to the flow of information and increasing number of posts available online. Not just inquirers and responders benefit from these sites, every single user online, such as programmers, academics, scientists, advanced tech users and just ordinary users, etc. takes advantage of these posts to solve their own problems based on existing ones, therefore models developed in our project can be used to facilitate other works.

6 ETHICAL ISSUES

Main aspect of the project is based on using the Stack Overflow data published online. Even though the data is accessible by everyone on the internet, there might be some issues while using it for research purposes due to the reason that it's going to be used without asking for permission. However there shouldn't be any ethical concerns related to this case since it's stated on the website hosting the data that users give permission for the data to be used in any shape or format, transformed and built upon the material for any purpose. For the front-end part of the project, there is a required login function for the system to be able to access user's Stack Exchange data and enables machine learning models to use this data for user's sake. However, neither front-end nor back-end stores the user's private or public data in another database,they only store some data as Cookies which is required for data passing between the front-end and the back-end. Furthermore, there should not be any other considerations regarding the legal/ethical aspects of the project.

7 PROJECT MANAGEMENT

7.1 Initial and Final Project Plans

Initial project plans have been defined as exclusive data analysis before starting the main tasks. Developing models for 4 main problems as they were indicated in objectives/tasks part.The initial plan consisted of developing the front-end software for demonstrating models and combining the machine learning tasks with the web-site that has been developed for this project. Furthermore, making improvements on model accuracy's and web-service. Initially, all these tasks have been planned to be finished within 14 weeks of course scope. However, there have been problems throughout the project development that led to changes in our plan. These problems and solutions have been discussed in the following subsections.

7.2 Task 1: : "Will my question get answered?"

For this task, initially the posts data set was planned to be used. However, features that have been provided were not enough to make good predictions. So, it has been decided to merge posts and users data sets of ai.stackexchange data set. It was planned to merge these data sets from the user id of the post's owner since they would intersect. However,the provided data set had many features similar to user id such as "Parent Id", "Owner user Id"," Id", "Last Editor User Id", "Account Id" and "userid". Identifying each of them took some time since there is no official document of the site for the definition of each feature. To demonstrate the relation between features a diagram has been provided. It indicates how each feature connects to the other one in different data sets of ai.stackexchange data(Badges, Comments, Posts, Users,Tags,Votes) . See APPENDIX D for the diagram. Nevertheless, the task has been finalized on time.

7.3 Task 2: "When Will my question get answered?"

At the beginning of the project this task has been defined as a regression problem. However, the results that have been reached out haven't been satisfying. For this reason, the problem has been transformed into classification task to get better results. After this transformation, still the model's score (F1 score) to predict response time classes were low. It was detected that the classes were not distributed in a balanced way. To prevent imbalance in data distribution, oversampling methods were applied.

7.4 Front-End

Implementation of the front-end started with utilizing a framework by Google called Firebase and it led to the division of the front-end in to three separate parts. First part is the user interface which uses Bootstrap 4 framework, HTML5 and CSS, second part is utilizing Javascript and jQuery to add functionalities to the website and handling the data transfer between the user interface and Firebase. Third and final part is so-called middle-ware and it is a Python 3 script which solves the data transfer issue between Firebase and the back-end of the project. Eventually, Firebase led to several complications and slowed down the user interface as well as complicating the middle-ware script while adding new functionalities. Therefore, it was swapped with Django 3 framework to solve all the problems regarding to the usage of Firebase and increasing the capabilities that can be given to a user in the user interface. Django 3 is a Python framework that enables to develop a server and create the website as a python web app. Switching from Firebase methodology to Django led to dropping the middle-ware part of the project and merging the back-end into the front-end. Now, each page of the web application can be generated through template views written in Python and data transfer between views are done by passing parameters through URLs and Cookies which are more common approaches than the old method. Features learned from this change is to plan the project while considering future implementations and problems that can be encountered throughout the development and try to use less frameworks and libraries to encourage minor transitions between different coding languages and providing a more stable build.

7.5 Learnings From the Project

Developing a project with many tasks and problems has benefited us greatly regarding different skills and aspects. Planning tasks thoroughly ahead and learning how to modify, change and question a viability of tasks throughout the development process has been some of the learnings. Some other lessons learned are better time management, more extensive researching and improving teamwork skills.

8 CONCLUSION AND FUTURE WORK

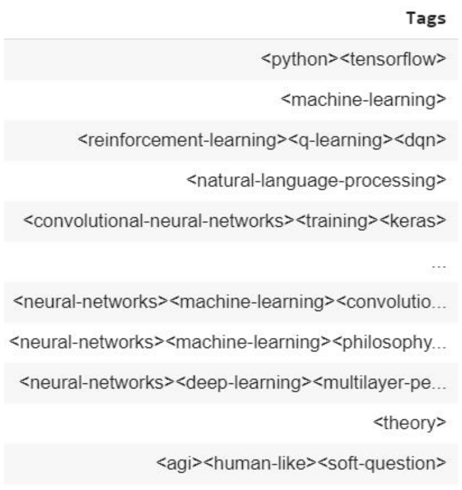
This project aims to improve Stack Overflow site in terms of user utilization. The improvements have been implemented with user friendly web-site for ease of use. In the scope of this project, it has been planned to provide Q&A community websites users information of : whether their questions are going to be answered, estimated answer time of their questions, context based similar questions that have been asked in the site with their question, which questions they are more likely to answer correctly. Machine learning models have been implemented in order to make predictions on these information. As it indicated in Results& Discussion, models are capable of making these predictions. Yet, there is a need of improvement in the tasks to make predictions more precise.

Moreover, for the future, other machine learning tasks can be defined to improve Stack Overflow site regarding user utilization to continue this project. Prediction can be made on the answer quality in the Q&A community websites. An another task could be predicting the best answers to a question. Further, prediction of whether a question will be closed could be an other task.

Finally, for both front-end and back-end, optimizations can be achieved to improve the speed of user interface. There are several features that can be added to the front-end and these features are currently in development. For example, there is a drop-down button added to the navigation bar, which is the left side of the screen, and this button will allow user to select the Q&A community websites that he/she is wants to work on. See APPENDIX C. Since this feature causes several bugs and complications in the flow of the front-end, it has been decided to be not urgent can wait for a fix in the future. Another development in progress is the Answer Question page, currently users can select a tag or select any questions that they want to answer in the Suggestions and Similar Questions section. These sections require better filtering, and Doc2Vec model requires better optimization in order load the page faster. This is currently the slowest loading page in the front-end. As the conclusion, this project can be expanded according to the needs and demands of users in Q&A community websites.

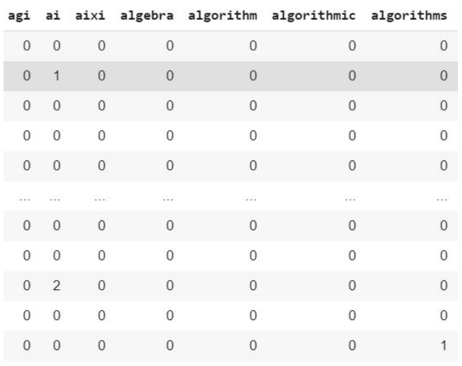
9 APPENDICES

9.1 APPENDIX A



Tags
<python><tensorflow>
<machine-learning>
<reinforcement-learning><q-learning><dqn>
<natural-language-processing>
<convolutional-neural-networks><training><keras>
...
<neural-networks><machine-learning><convolutio...
<neural-networks><machine-learning><philosophy...
<neural-networks><deep-learning><multilayer-pe...
<theory>
<agi><human-like><soft-question>

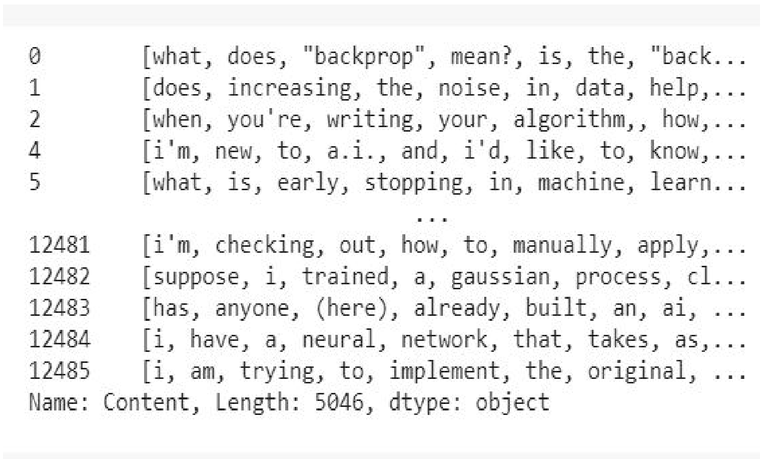
Figure 15: Tags data snippet before processing



agi	ai	aixi	algebra	algorithm	algorithmic	algorithms
0	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
...	...	...	...	...	...	...
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	2	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	1

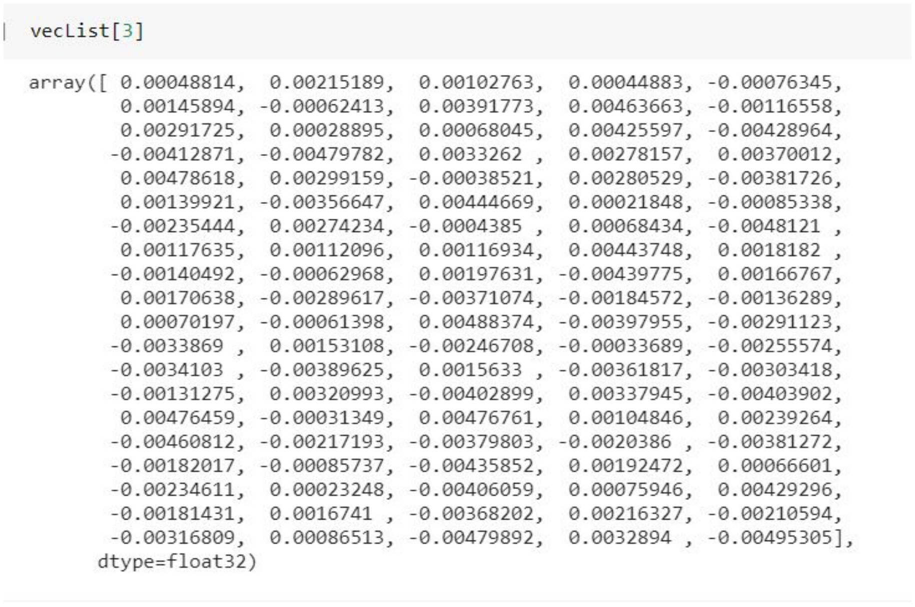
Figure 16: Tags data snippet after processing

9.2 APPENDIX B



```
0        [what, does, "backprop", mean?, is, the, "back...
1        [does, increasing, the, noise, in, data, help,...
2        [when, you're, writing, your, algorithm,, how,...
4        [i'm, new, to, a.i., and, i'd, like, to, know,...
5        [what, is, early, stopping, in, machine, learn...
                                ...
12481    [i'm, checking, out, how, to, manually, apply,...
12482    [suppose, i, trained, a, gaussian, process, cl...
12483    [has, anyone, (here), already, built, an, ai, ...
12484    [i, have, a, neural, network, that, takes, as,...
12485    [i, am, trying, to, implement, the, original, ...
Name: Content, Length: 5046, dtype: object
```

Figure 17: Content(title + body of a post) data snippet before processing



```
vecList[3]
```

```
array([ 0.00048814,  0.00215189,  0.00102763,  0.00044883, -0.00076345,
        0.00145894, -0.00062413,  0.00391773,  0.00463663, -0.00116558,
        0.00291725,  0.00028895,  0.00068045,  0.00425597, -0.00428964,
       -0.00412871, -0.00479782,  0.0033262 ,  0.00278157,  0.00370012,
        0.00478618,  0.00299159, -0.00038521,  0.00280529, -0.00381726,
        0.00139921, -0.00356647,  0.00444669,  0.00021848, -0.00085338,
       -0.00235444,  0.00274234, -0.0004385 ,  0.00068434, -0.0048121 ,
        0.00117635,  0.00112096,  0.00116934,  0.00443748,  0.0018182 ,
       -0.00140492, -0.00062968,  0.00197631, -0.00439775,  0.00166767,
        0.00170638, -0.00289617, -0.00371074, -0.00184572, -0.00136289,
        0.00070197, -0.00061398,  0.00488374, -0.00397955, -0.00291123,
       -0.0033869 ,  0.00153108, -0.00246708, -0.00033689, -0.00255574,
       -0.0034103 , -0.00389625,  0.0015633 , -0.00361817, -0.00303418,
       -0.00131275,  0.00320993, -0.00402899,  0.00337945, -0.00403902,
        0.00476459, -0.00031349,  0.00476761,  0.00104846,  0.00239264,
       -0.00460812, -0.00217193, -0.00379803, -0.0020386 , -0.00381272,
       -0.00182017, -0.00085737, -0.00435852,  0.00192472,  0.00066601,
       -0.00234611,  0.00023248, -0.00406059,  0.00075946,  0.00429296,
       -0.00181431,  0.0016741 , -0.00368202,  0.00216327, -0.00210594,
       -0.00316809,  0.00086513, -0.00479892,  0.0032894 , -0.00495305],
      dtype=float32)
```

Figure 18: Content (title + body of a post)data snippet for one post vector size=100 after processing

9.3 APPENDIX C

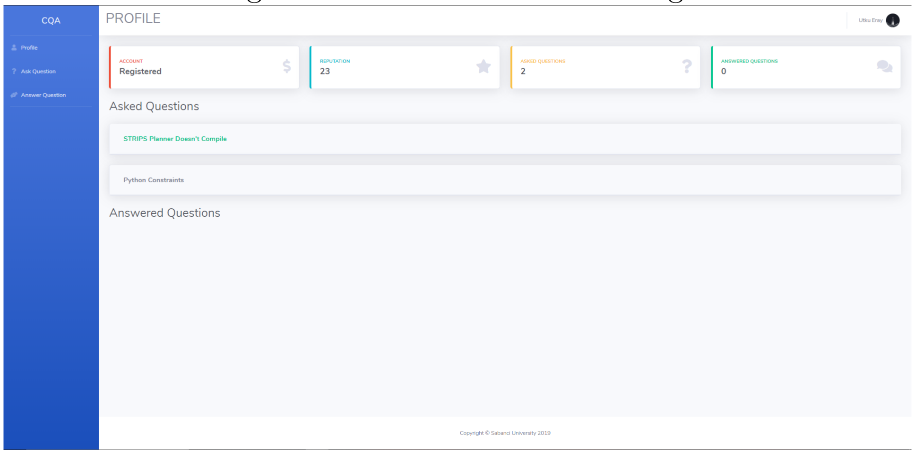


Figure 19: Front-End: Profile Page

ACCOUNT
Registered

REPUTATION
23

ASKED QUESTIONS
2

ANSWERED QUESTIONS
0

Asked Questions

STRIPS Planner Doesn't Compile

Python Constraints

Answered Questions

Copyright © Sabanci University 2019

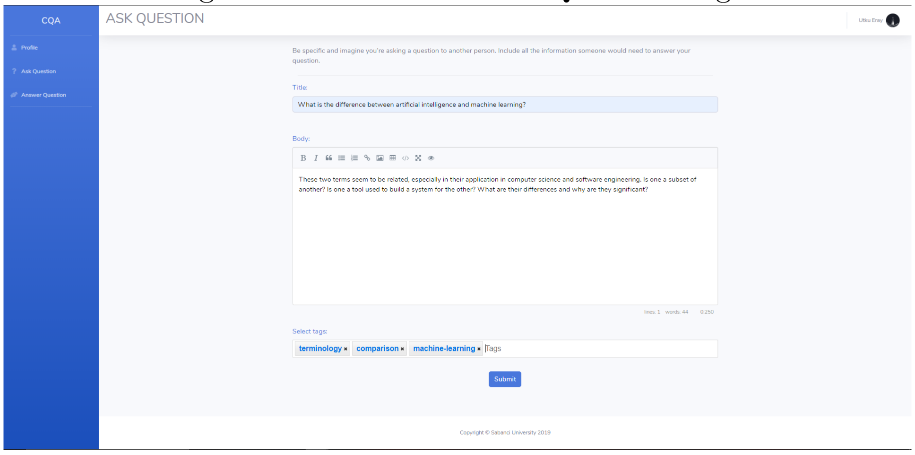


Figure 20: Front-End: Ask Question Page

Be specific and imagine you're asking a question to another person. Include all the information someone would need to answer your question.

Title:
What is the difference between artificial intelligence and machine learning?

Body:
These two terms seem to be related, especially in their application in computer science and software engineering. Is one a subset of another? Is one a tool used to build a system for the other? What are their differences and why are they significant?

lines: 1 words: 44 0:250

Select tags:
terminology × comparison × machine-learning × Tags

Copyright © Sabanci University 2019

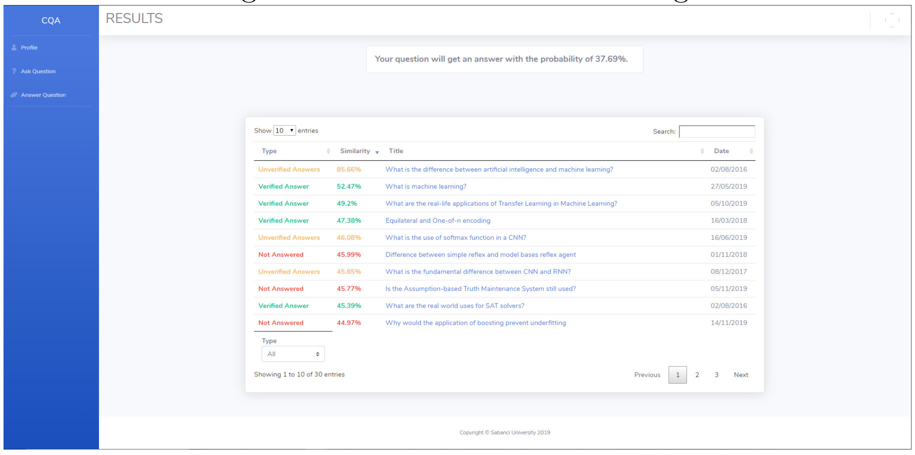


Figure 21: Front-End: Results Page

Your question will get an answer with the probability of 37.69%.

Show 10 entries Search:

Type	Similarity	Title	Date
Unverified Answers	85.66%	What is the difference between artificial intelligence and machine learning?	02/08/2016
Verified Answer	52.47%	What is machine learning?	27/05/2019
Verified Answer	49.2%	What are the real-life applications of Transfer Learning in Machine Learning?	05/10/2019
Verified Answer	47.38%	Equilateral and One-of-n encoding	16/03/2018
Unverified Answers	46.08%	What is the use of softmax function in a CNN?	16/06/2019
Not Answered	45.99%	Difference between simple reflex and model bases reflex agent	01/11/2018
Unverified Answers	45.85%	What is the fundamental difference between CNN and RNN?	08/12/2017
Not Answered	45.77%	Is the Assumption-based Truth Maintenance System still used?	05/11/2019
Verified Answer	45.39%	What are the real world uses for SAT solvers?	02/08/2016
Not Answered	44.97%	Why would the application of boosting prevent underfitting	14/11/2019

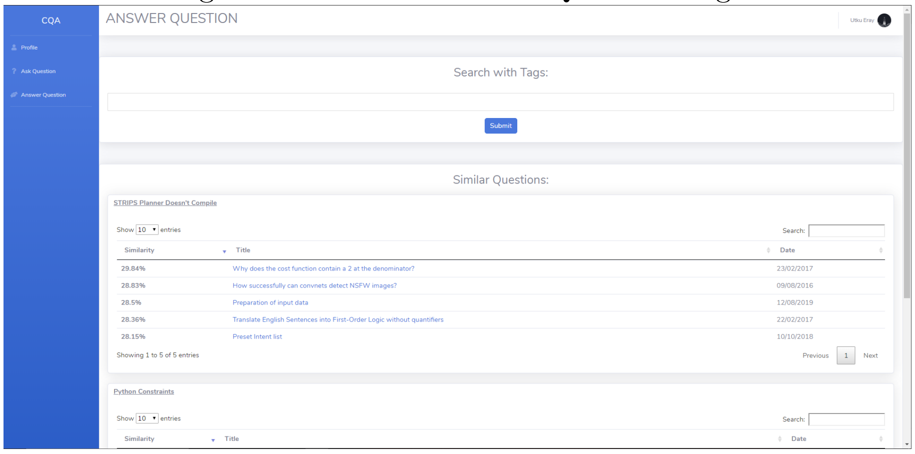
Type: All

Showing 1 to 10 of 30 entries

Previous 1 2 3 Next

Copyright © Sabanci University 2019

Figure 22: Front-End: Ask Question Page



ANSWER QUESTION

Search with Tags:

Similar Questions:

STRIPS Planner Doesn't Compile

Show 10 entries

Search:

Similarity	Title	Date
29.84%	Why does the cost function contain a 2 at the denominator?	23/02/2017
28.83%	How successfully can convnets detect NSFW images?	09/08/2016
28.5%	Preparation of input data	12/08/2019
28.36%	Translate English Sentences into First-Order Logic without quantifiers	22/02/2017
28.15%	Preset Intent list	10/10/2018

Showing 1 to 5 of 5 entries

Previous 1 Next

Python Constraints

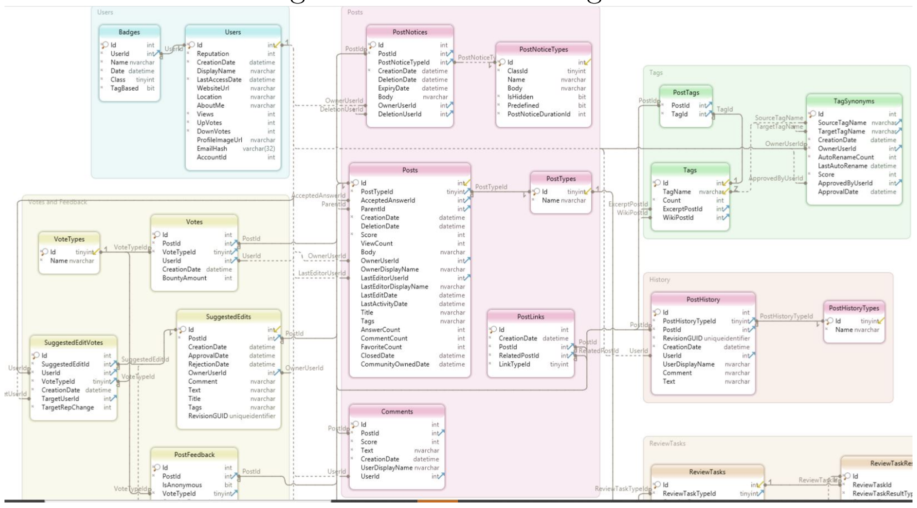
Show 10 entries

Search:

Similarity	Title	Date

9.4 APPENDIX D

Figure 23: Features Diagram



Retrieved from: (https://i.stack.imgur.com/AyIkW.png, n.d.)

10 REFERENCES

- Asaduzzaman, M., Mashiyat, A. S., Roy, C. K., & Schneider, K. A. (2013). Answering questions about unanswered questions of Stack Overflow. 2013 10th Working Conference on Mining Software Repositories (MSR) doi:10.1109msr.2013.6624015

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In NIPS (pp. 3111–3119) . Curran Associates, Inc.

- Sahu, T. P., Nagwani, N. K., & Verma, S. (2016). Selecting Best Answer: An Empirical Analysis on Community Question Answering Sites. IEEE Access, 4, 4797–4808. doi: 10.1109/access.2016.260062

- Where Developers Learn, Share, & Build Careers. (n.d.). Retrieved from: https://stackoverflow.com/.

- Trienes, & Balog. (2019). Papers With Code : Identifying Unclear Questions in Community Question Answering Websites.
Retrieved from https:paperswithcode.com/paper/identifyingunclearquestions-incommunity

- Yeniterzi, & Callan. (2014). Analyzing Bias in CQA-based Expert Finding Test Sets.Retrieved from
http:/www.cs.cmu.edu/afs/.cs.cmu.edu/Web/People/reyyan/publications/SIGIR-2014Yeniterzi.pdf

- Le, Q., & Mikolov, T. (2013). Distributed Representations of Sentences and Documents. Retrieved from https://arxiv.org/pdf/1405.4053.pdf

- Mikolov, T.,Dean, J.,Corrado, G. & Chen, K. (2013). Efficient Estimation of Word Representations in Vector Space.
Retrieved from https:arxiv.org/pdf/1310.4546.pdf

- Cosine similarity. (2007, January 17).
Retrieved from https:/en.wikipedia.org/wiki/Cosine similarity

- Stack Exchange. 2019. Retrieved from https://stackexchange.com/sites

- Nagesh Singh Chauhan (2020, January 10) Random Forest — A powerful ensemble learning algorithm Medium https://towardsdatasciencecom/random-forest-a-powerful-ensemble-learning-algorithm-2bf132ba639d