A project report on

FINDING ANSWERERS ON STACK OVERFLOW

Submitted in partial fulfillment for the course

Social and Information Networks

by

MEGHA NATH (20BCE1581)

ANEGHA JAIN (20BCE1547)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2023

DECLARATION

We hereby declare that the thesis entitled "FINDING ANSWERERS ON STACKOVERFLOW" submitted by us, for the completion of the course, Social and Information Networks (CSE3021) is a record of bonafide work carried out by us under the supervision of Dr Punitha K, our course instructor. We further declare that the work reported in this document has not been submitted and will not be submitted, either in part or in full, for any other courses in this institute or any other institute or university.

Place: Chennai

Date: 12.04.23 Signature of Candidate

Signature of Candidate

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled **"Finding Answerers on StackOverflow"** is prepared and submitted by **Megha Nath** (**20BCE1581**), **Anegha Jain** (**20BCE1547**)to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the course, **Social and Information Networks (CSE3021),** is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for any other course and the same is certified.

Name: Dr. Punitha K Signature of the Faculty

Date: 12.04.23

ABSTRACT

Online forums are used extensively by people to find answers to questions which a simple Google search may not be able to provide. These questions are usually industry-specific and specialized, often requiring a subject matter expert to answer them. Most of such questions asked on online forums suffer from the Free Rider Problem, where users ask questions but do not actively look for questions to answer. Finding appropriate users to answer these questions is one of the main challenges of the Question Answer communities. Propagating these questions to such users thus, becomes very important. In this work, we analyze the StackOverflow database and study some interesting properties that exist in the network structure. Further, we use K-means clustering algorithm and an ensemble of various classification algorithms to find out the most probable group of users who might be able to answer a question posed by a questioner using textual, structural and auxiliary information.

CONTENTS

LIST OF FIGURES 7

LIST OF TABLES 8

CHAPTER 1

INTRODUCTION

1.1 STACKOVERFLOW 8

1.1.1 STACKOVERFLOW SIZE 8

1.1.2 HOW STACKOVERFLOW WORKS 8

1.2 PROBLEM STATEMENT AND PROJECT OBJECTIVE 9

CHAPTER 2

METHODOLOGY

2.1 DATA 10

2.2 CHARACTERIZATION OF STACKOVERFLOW 12

2.3 MODULES AND MODELS USED 15

2.3.1 FUZZY-C MEANS 15

2.3.2 K-MEANS 16

2.3.3 FEATURE EXTRACTION USING RFE 17

2.3.4 MLP 17

2.3.5 ENSEMBLE LEARNING 18

2.4 ANSWER POOL DETECTION ALGORITHM 19

2.4.1 HOW IT WORKS 20

2.4.2 CASCADED CLUSTERING 21

2.4.3 CLASSIFYING THE QUESTION 22

CHAPTER 3

RESULTS

3.1 CLUSTERING RESULTS 23

3.2 CLASSIFICATION RESULTS 24

CHAPTER 4

DISCUSSION

DISCUSSION 26

CHAPTER 5

CONCLUSION

CONCLUSION 28

CHAPTER 6

FUTURE WORK

FUTURE WORK 29

CHAPTER 7

IMPLEMENTATION 30

REFERENCES

REFERENCES 38

LIST OF FIGURES

1 Stack Exchange Data Explorer 11

2 Stack Exchange Data obtained 12

3 Tag Names with Count 13

4 Number of Answers per question 13

5 Latency for the Best Answer across questions 14

6 Comparison of First Answer and Best Answer Latencies 14

7 Clustering of Categories by Thread and Post Lengths 15

8 Elbow plot for Clustering 16

9 Answer pool detection algorithm flowchart 19

10 Cascaded Clustering 20

11 K-Means Clustering Results 22

12 Fuzz-C Means Clustering Results 22

13 Precision 23

14 Recall 24

LIST OF TABLES

1. Summary of Collected Data 11

2. Accuracies of Classification models 23

Chapter 1

Introduction

1.1 STACKOVERFLOW

StackOverflow is a technical online forum. It is the flagship website of the Stack Exchange Network. It is a question and answer website. It was created in 2008 by Jeff Atwood and Joel Spolsky. It features questions and answers on certain computer programming topics. It was created to be a more open alternative to earlier question and answer websites such as Experts-Exchange.

1.1.1 STACKOVERFLOW SIZE

As of March 2022, StackOverflow has over 20 million registered users. It has over 24 million questions and over 35 million answers. Such Q&A sites have globally replaced programming books for day-to-day programming reference and have become an integral part of computer programming today. Based on the type of tags assigned to questions, the top eight most discussed topics on the site are:

- JavaScript
- Java
- C#
- PHP
- Android
- Python
- jQuery
- HTML

1.1.2 HOW STACKOVERFLOW WORKS

Users are capable of posting questions relating to technical queries that they may have. They can also answer questions that they may have interest in or whose subject they might be experts in. An answer gets “upvotes” or “downvotes” from the community if it is judged as correct or incorrect respectively by the members of the community. Here, community members are the users of the Stack Exchange Network.

StackOverflow is a gamified Q&A website where the user whose answers get the most upvotes is awarded with reputation points or badges. Users are able to unlock new privileges with an

increase in their reputation like the ability to vote, comment and even edit other people's posts or questions.

A question or answer posted by a user counts as a post. For a particular question, the answers and comments posted by other users make up a thread. So every question a user posts there is a thread of answers and comments. Users can even link other question threads in a particular question thread.

A question thread suffers "closing" when there is no activity from the poster's side or there have not been answers to the question from any other user. Apart from that, users are also capable of closing a thread if they find an answer in the question thread that satisfies their query. A question may also suffer closing if it is of a broader nature or seems to invite personal opinions, and does not adhere to being tightly focused on a problem. Closing questions is a main differentiation from other Q&A sites like Yahoo! Answers and a way to prevent low quality questions.

All user-generated content is licensed under Creative Commons Attribute-ShareAlike license, version 2.5, 3.0, or 4.0 depending on the date the content was contributed.

1.2 PROBLEM STATEMENT AND PROJECT OBJECTIVE

Given the amount of questions and answers on StackOverflow it is not surprising to know that certain questions on the site are left unanswered or are closed because of no activity associated with the question thread. Although some question threads do suffer from closing due to genuinely receiving no answers as none exist, but most questions suffer due to the Free Rider Problem.

The Free Rider Problem happens when a user promptly asks questions but does not answer questions, that is, a user generates activity only by posting questions and does not answer questions posted by other users despite being a subject matter expert. This is mainly due to users not coming across questions that may be capable of answering. Therefore, it is required to build a technology that will help us propagate such questions to users that may be interested in answering them.

Thus the main objective of our project was to build a model that would propagate such questions to answerers on the StackOverflow website and help diminish the Free Rider Problem.

The kind of answers present on the StackOverflow website is diverse which is why we intend to take the textual information of the question along with the attributes of the questioner obtained from the structural properties of the user's interaction graph and also auxiliary information like the quality of the answers he has given and cluster them by a k-means clustering algorithm. For a new question posed by a user, we can then route the question to answerers for the question in an obtained cluster.

Chapter 2

Methodology

2.1 DATA

Data for our project was collected from the [Stack Exchange Data Explorer website](#).

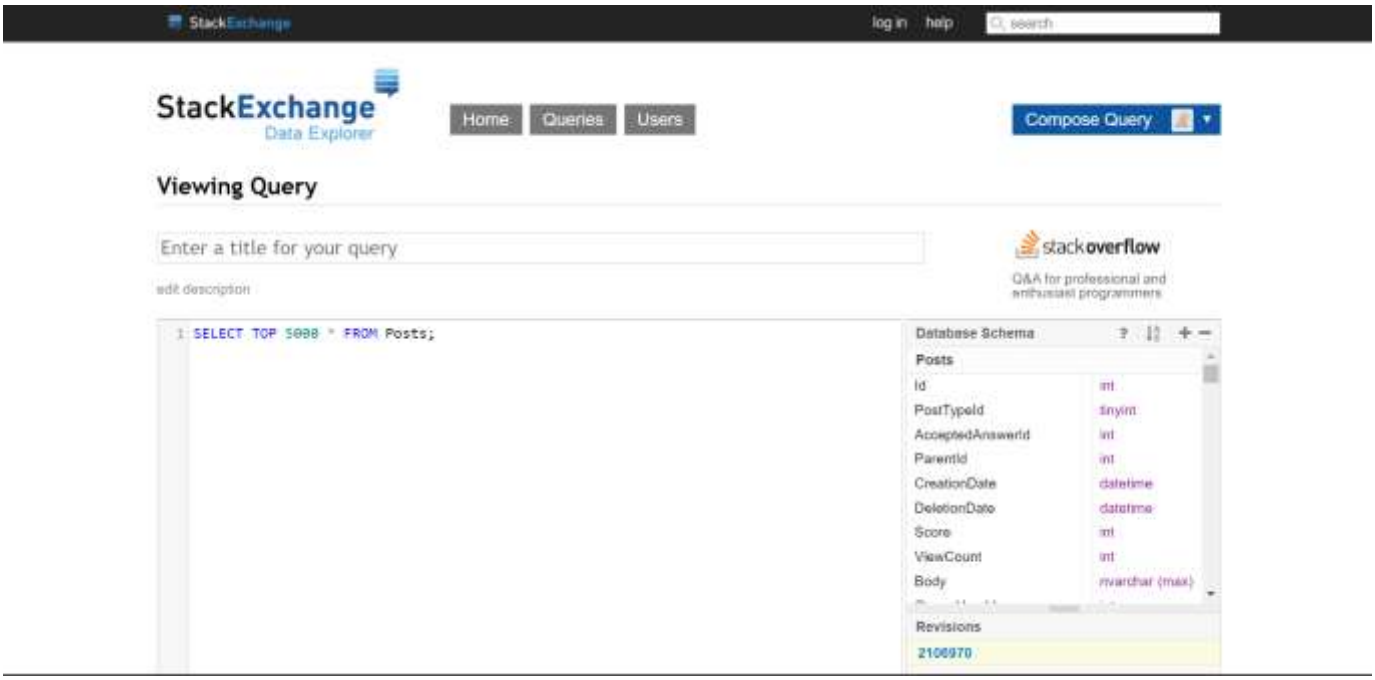

Fig 1. Stack Exchange Data Explorer

Writing a query into the query editor and running the query we got results. The results could be downloaded as a CSV file.

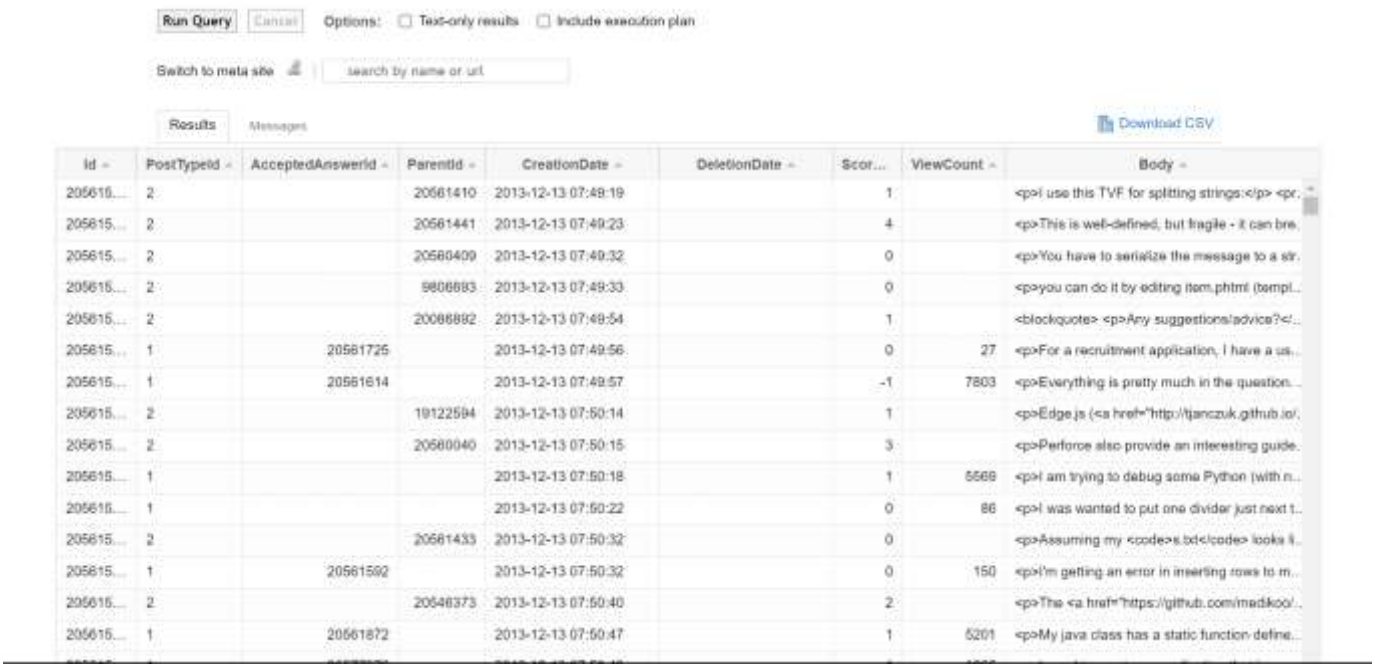

Fig 2. Stack Exchange Data obtained

Summary of the data collected is given below:

No. of Posts	40000
No. of Questions	13395
No. of Answers	26561

Table 1. Summary of Collected Data

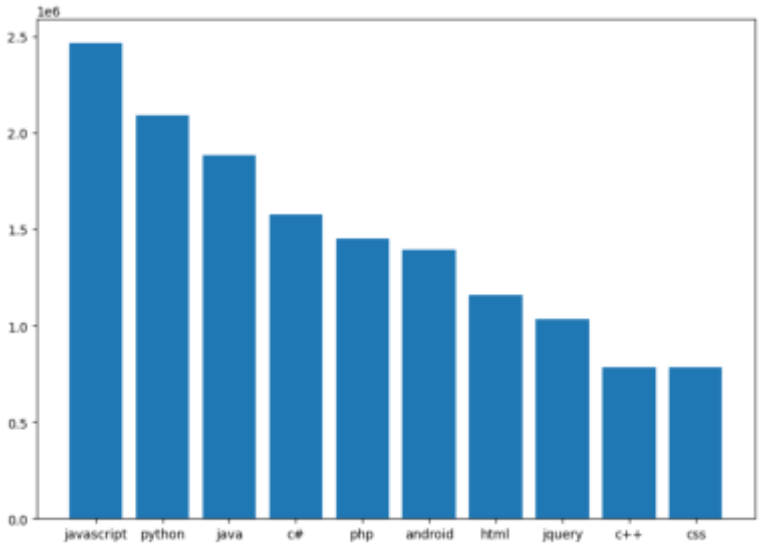

Fig 3. Tag Names with Count

2.2 CHARACTERIZATION OF STACKOVERFLOW

StackOverflow is not necessarily a network structure like Facebook, LinkedIn etc. On StackOverflow all users can either be Questioners or Answerers. A question that is put out on StackOverflow is called a Post and the collection of the question with all of its answers is called a Thread.

In this section we investigate certain properties of StackOverflow Posts:

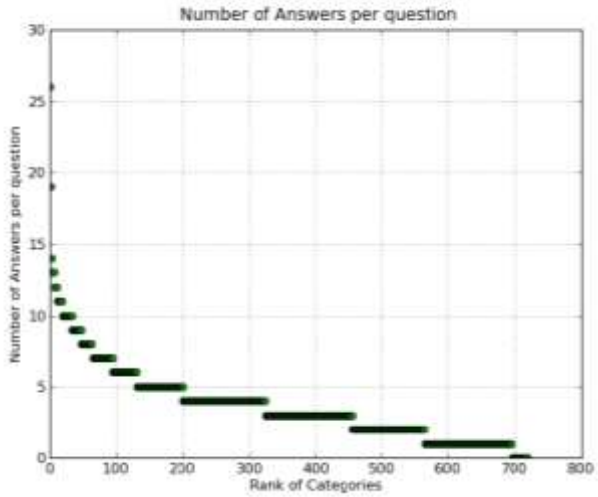

Fig 4. Number of Answers per question

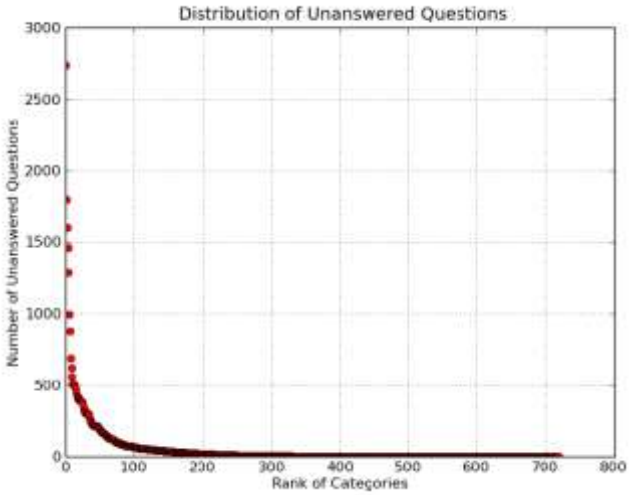

Fig 4. Distribution of Unanswered Questions

Fig 3. and Fig 4. show that it does sometime take an huge amount of time before a question can be answered and this is usually due to the fact that the potential answerers to those questions have not been

able to see them.

Fig 5. Latency for the Best Answer across questions

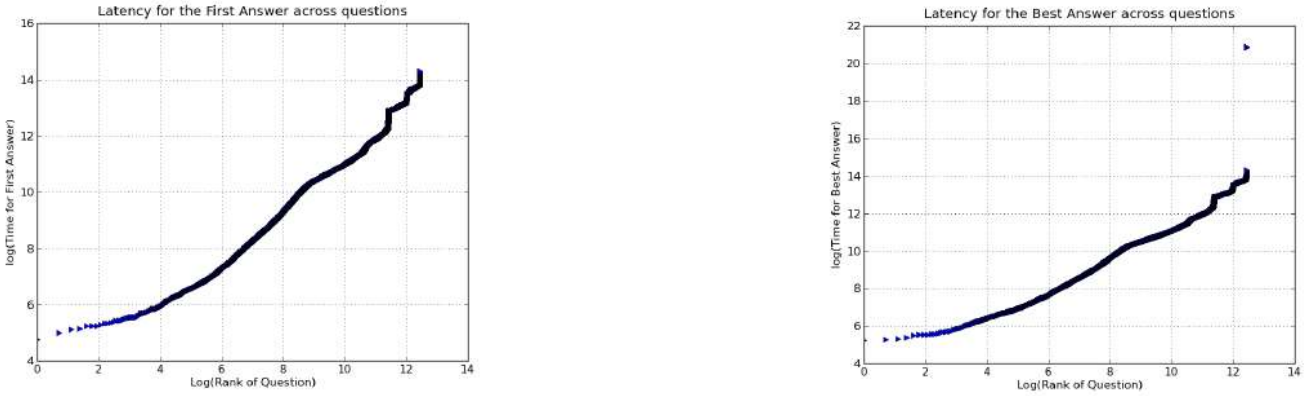

Fig 6. Comparison of First Answer and Best Answer Latencies

Fig 7. shows the result of a K-means clustering on this plot. Here, the categories on the lower green cluster (esp on the left) are categories where there are clear notions of experts, and they provide answers that saturate the discussion. Programming is an example of this category.

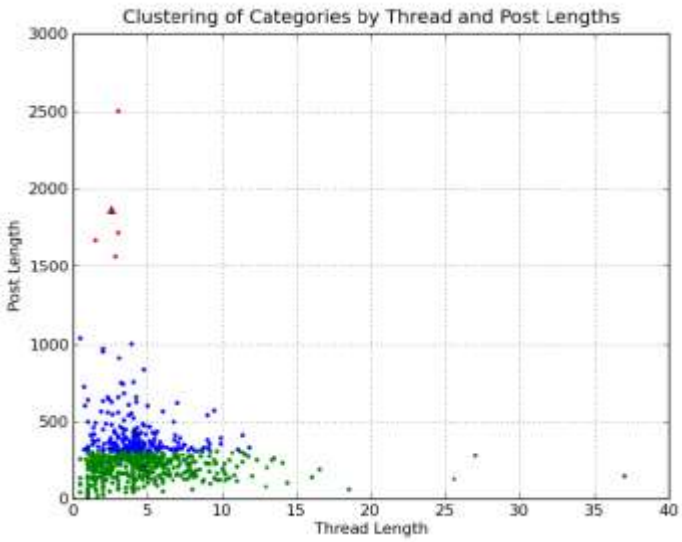

Fig 7. Clustering of Categories by Thread and Post Lengths

2.3 MODULES AND MODELS USED

2.3.1 Fuzzy - C Means Clustering

Fuzzy c-means (FCM) is a data clustering technique in which a data set is grouped into N clusters with every data point in the dataset belonging to every cluster to a certain degree. Data are sorted into different clusters in non-fuzzy clustering (also known as hard clustering), where each data point can only belong to one cluster. Data points may potentially belong to more than one cluster in fuzzy clustering. An apple, for instance, can be either red or green (hard clustering), but it can also be both red and green. (fuzzy clustering). In this instance, the apple can be partially red and partially green. The apple can belong to green [green = 0.5] and red [red = 0.5] rather than to green [green = 1] and not red [red = 0]. These numbers have a normal distribution between 0 and 1, but since they don't correspond to probabilities, they don't have to sum up to 1.

We perform the FCM on the questions extracted from our 40,000 posts dataset. According to the elbow method the best number of clusters was 7.

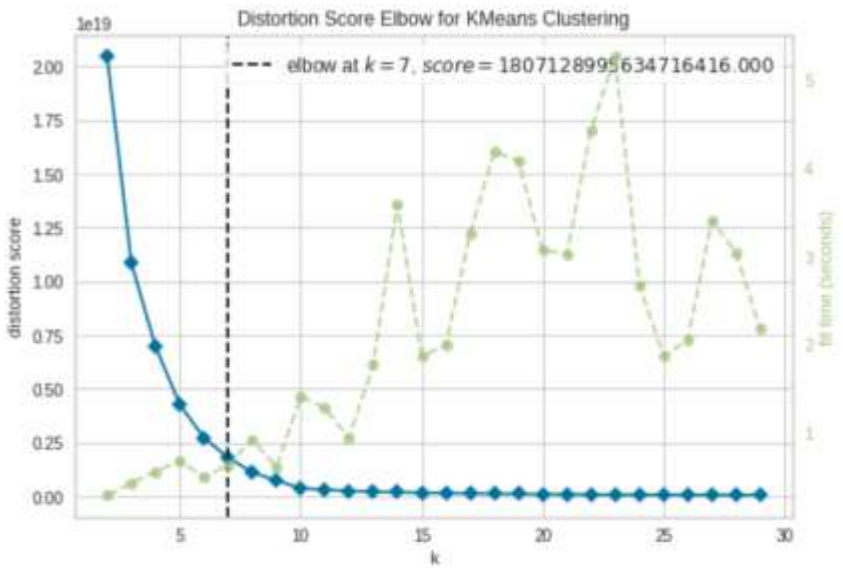

Fig 8. Elbow plot for Clustering

2.3.2 K-Means Clustering

K-means clustering is a method of vector quantization that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. K-means clustering minimizes the within cluster variances which are the squared Euclidean distances.

The three crucial aspects of k-means that make it effective are frequently seen as its three worst flaws:

- As a metric and a gauge of cluster scatter, respectively, variance and Euclidean distance are employed.
- An incorrect choice of k could lead to subpar outcomes as it is an input parameter for the number of clusters. Because of this, it's crucial to do diagnostic checks when using k-means to figure out how many clusters there are in the data set.
- Convergence to a local minimum could lead to unexpected (wrong) outcomes.

The cluster model of k-means is one of its main drawbacks. The idea is based on spherical clusters that can be divided, causing the mean to move in the direction of the cluster centre. Since it is anticipated that the clusters will be of a comparable size, the assignment to the nearest cluster centre is the right assignment.

For our project, K-means was applied with 7 clusters (as found from the Elbow Method) for the questions extracted from the StackOverflow dataset.

2.3.3 Feature Extraction using RFE

Recursive Feature Elimination, or RFE for short, is a popular feature selection algorithm. RFE is popular because it is easy to configure and use and because it is effective at selecting those features (columns) in a training dataset that are more or most relevant in predicting the target variable. There are two important configuration options when using RFE: the choice in the number of features to select and the choice of the algorithm used to help choose features.

In order to score a feature subset, filter approaches substitute a proxy measurement for the error rate. This measurement was chosen since it is quick to compute and captures the value of the feature collection. The mutual information, the pointwise mutual information, the Pearson product-moment correlation coefficient, Relief-based techniques, the inter/intra class distance, or the results of significance tests for each class/feature combination are examples of common measurements. Although filters often use less computing power than wrappers, they create a feature set that is not tailored to a particular kind of prediction model. Due to the lack of tailoring, a filter's feature set is generally more generic than a wrapper's, which results in inferior prediction performance. The feature set is more beneficial for exposing the links between the features because it lacks the assumptions of a prediction model. In lieu of a clear best feature subset, many filters offer a feature ranking, with cross-validation used to determine the ranking's cutoff point. In order to enable the employment of wrapper methods on more complex situations, filter techniques have also been utilised as a preprocessing step. The Recursive Feature Elimination algorithm is another well-liked method that is frequently used with Support Vector Machines to continually build a model and eliminate features with low weights.

For our project, RFE was used to extract the most important features present in the dataset. The features that provided most information and were independent were the following: AcceptedAnswerId, Score, AnswerCount.

2.3.4 MLP

Multilayer Perceptron (MLP) is a fully connected class of feedforward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a chain rule based supervised learning technique called backpropagation or reverse mode of automatic differentiation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

If a multilayer perceptron has a linear activation function in all neurons, that is, a linear function that maps the weighted inputs to the output of each neuron, then linear algebra shows that any number of layers can be reduced to a two-layer input-output model. In MLPs some neurons use a nonlinear activation function that was developed to model the frequency of action potentials, or firing, of biological neurons.

In recent developments of deep learning the rectified linear unit (ReLU) is more frequently used as one of the possible ways to overcome the numerical problems related to the sigmoids.

2.3.5 ENSEMBLE LEARNING

The ensemble methods in machine learning combine the insights obtained from multiple learning models to facilitate accurate and improved decisions. In learning models, noise, variance, and bias are the major sources of error. The ensemble methods in machine learning help minimize these error-causing factors, thereby ensuring the accuracy and stability of machine learning (ML) algorithms.

Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

Empirically, ensembles tend to yield better results when there is a significant diversity among the models. Many ensemble methods, therefore, seek to promote diversity among the models they combine. Although perhaps non-intuitive, more random algorithms (like random decision trees) can be used to produce a stronger ensemble than very deliberate algorithms (like entropy-reducing decision trees). Using a variety of strong learning algorithms, however, has been shown to be more effective than using techniques that attempt to dumb-down the models in order to promote diversity. It is possible to increase diversity in the training stage of the model using correlation for regression tasks or using information measures such as cross entropy for classification tasks.

For our project, an ensemble of Decision Tree Classifier, Random Forest Classifier and K Nearest Neighbors was used. These models were stacked on top of each other for predicting the class that a particular post would belong to. The final estimator applied after the stack was Logistic Regression.

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules. In decision analysis, a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected

values (or expected utility) of competing alternatives are calculated.

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.

k-NN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data can improve its accuracy dramatically. Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (the coefficients in the linear combination). Formally, in binary logistic regression there is a single binary dependent variable, coded by an indicator variable, where the two values are labeled "0" and "1", while the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name.

2.4 ANSWER POOL DETECTION ALGORITHM

The algorithm requires generation of feature vectors from our initial data and converting them into clusters using any Centroid based models (K-Means Clustering). Further using cascaded clustering, we find a specific pool of answerers for our question/feature vectors.

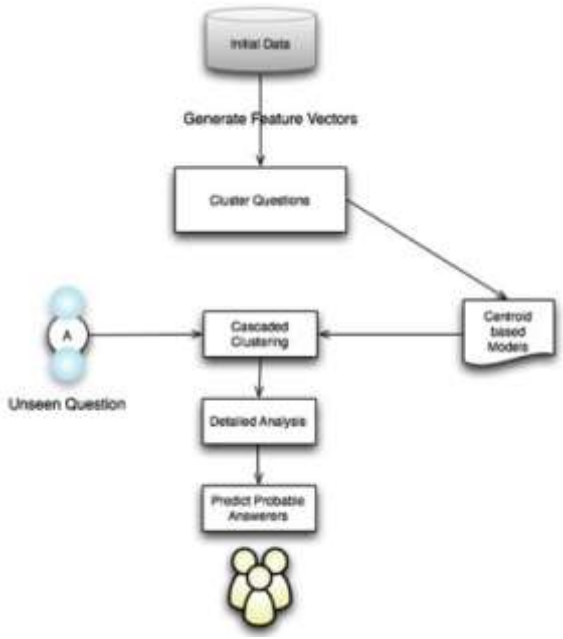

Fig 9. Answer pool detection algorithm flowchart

Answer Pool Detection Algorithm (f_i)

1. *Start*
2. *Cluster f_i into $C_i \epsilon$ Clusters of questions*
3. *From the cluster C_i assign f_i to a cluster $C_{i,j} \epsilon C_i$*
4. ***return***

2.4.1 HOW IT WORKS

The Answer Pool Detection algorithm works by clustering a question asked by any user on the platform. The objective of the algorithm is to identify Free Riders on the website. A Free Rider is a user that usually posts questions but does not answer potential questions that could be of their interest. This usually happens because the user cannot find questions that they could be a subject matter expert of or may have an interest in. An incoming question or a new question posted by a user to the stackoverflow website undergoes clustering under the Answer Pool Detection Algorithm. Important features from the posted question like the date it was posted, tags, content etc, are collected and fed into the model. The model will cluster the question into one of the Question clusters. Within these clusters the new question is clustered again to find the best answer pool. The questions within a cluster have a set of answerers associated with them. These are the answerers that are returned by the algorithm.

The returned answer pool can then be fed into the recommendation system that may be

working behind the scenes for the StackOverflow website. The recommendation system can then propagate the question to the ‘Free Riders’ on the StackOverflow website. These users can finally view and answer the questions that may interest them or they are subject matter experts on. This benefits the Free Riders as well as answering questions gives them privileges on StackOverflow given that StackOverflow is a gamified question and answer website. It will give them privileges such as voting answers, getting badges or editing other users’ questions.

2.4.2 CASCADED CLUSTERING

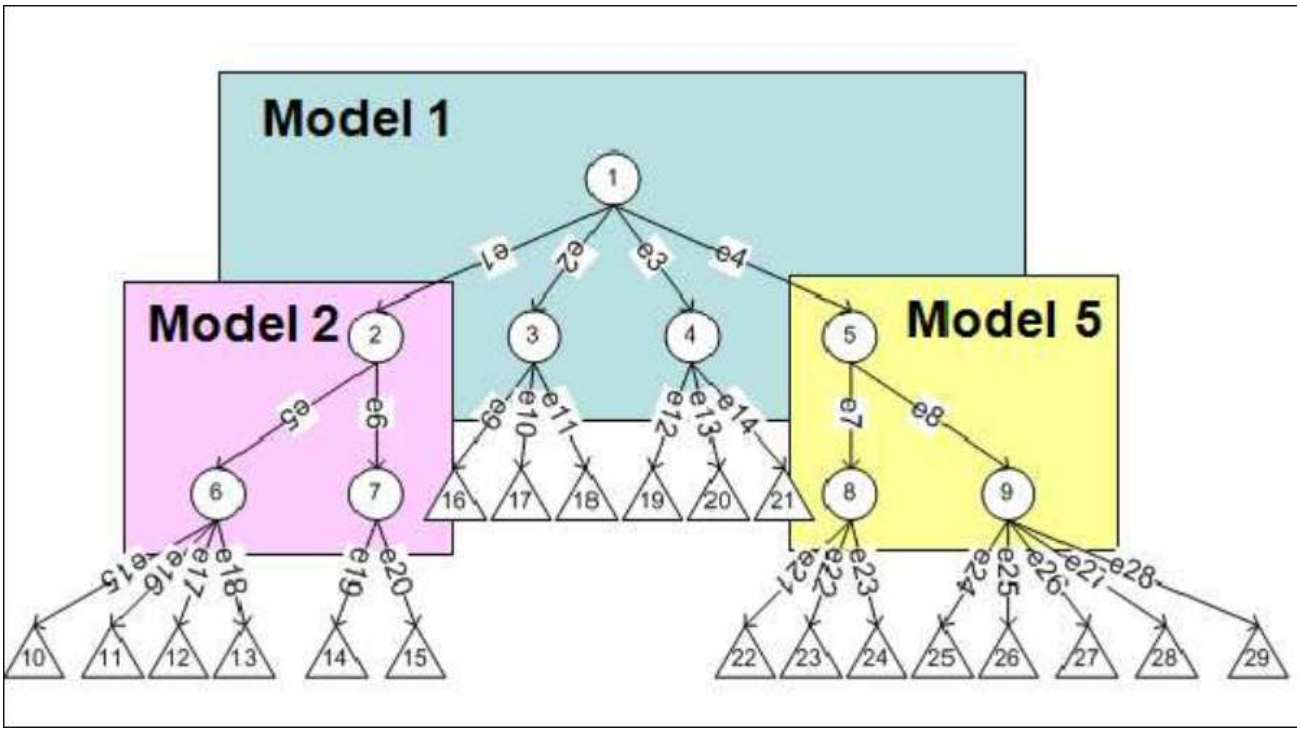

Fig 10. Cascaded Clustering

The key concept involved in the Answer Pool Detection algorithm is Cascaded Clustering. Cascaded Clustering is a type of nested clustering to find a cluster for the data more precisely.

Cascaded clustering was required for our project as our main purpose was to find a pool of potential answerers for the question. This requires great precision to pin down the users that may be interested in answering the question. Although one level of clustering would also give us the answerers but the number of answerer IDs that would be returned would be far too large. The cascaded clustering method ensures that the answer pool returned would contain the most potential answerers for that question. Although larger answerer pool does increase the probability of the question being answered, it would not be feasible for efficient recommendation systems. Frequent wrong recommendations to a user

may decrease the website's reputation as well.

Keeping such points in mind it seems appropriate to use the cascaded clustering method to get down to the best answerer pool for that particular question.

2.4.3 CLASSIFYING THE QUESTION

Our project also tried to do away the time consuming process of clustering every new data point. The main disadvantage with clustering is that it is unsupervised and does not really "learn" anything from the data and clusters data by comparing its distance to every other point in the data.

Furthering the project we employed a Multi Layer Perceptron and Ensemble Learning for categorizing an incoming input data into one of the categories and finding the answer pool from that category. The clusters assigned to each data point during clustering forms the classes during the classification. This way our model ends up "learning" from the dataset and does not spend time calculating distances. Since classification is also supervised, it helps our model to learn with lesser errors.

Now classification will help us classify the question instead of clustering thus reducing time in finding the answerer pool.

From our findings we concluded that the Ensemble Learning approach works better and produces more accurate results as compared to a deep learning approach using Multi Layer Perceptron. Thus, using the Features Extracted using Recursive Feature Elimination and feeding them into the Stacked Classifier we classify the questions and find an answer pool from that class.

Chapter 3

Results

3.1 CLUSTERING RESULTS

In our project we applied 2 types of Clustering. First we used the elbow method to find the optimal number of clusters. After performing the K-Means clustering we got the clusters as shown in Fig 10.

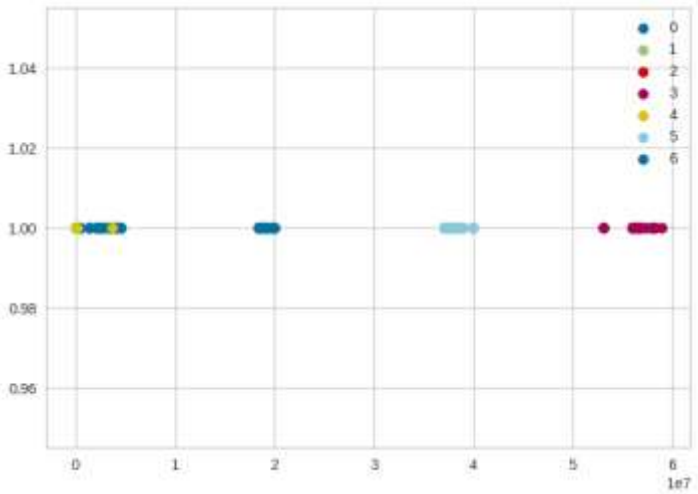

Fig 11. K-Means Clustering Results

Then we performed Fuzzy C-Means Clustering and obtained the clusters as shown in Fig 11.

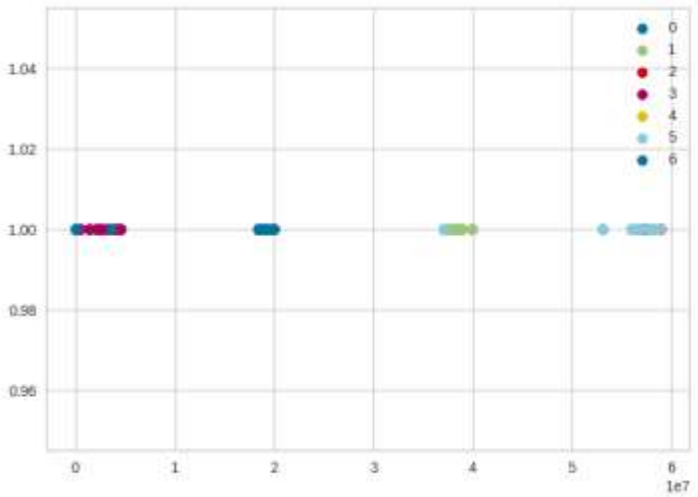

Fig 12. Fuzzy C-Means Clustering

After getting results for both K-Means and Fuzzy C-Means Clustering we applied Cascaded Clustering which gave us our final result of Top N Answerers for a particular question.

3.2 CLASSIFICATION RESULTS

Models	**Accuracies**
Decision Tree Classifier	0.9999253452780887
Random Forest Classifier	0.9998506905561776
K Nearest Neighbors	0.9997013811123553
Stacked Classifier	0.9999253452780887
Multi-Layer Perceptron	0.4075

Table 2. Accuracies of Classification models

As can be observed from the above results, all the classifiers except MLP give us great results however the best result is given by Decision Tree Classifier and Ensemble learning based Stacked Classifier.
The figures below show the precision and recall changes for each cluster/class.

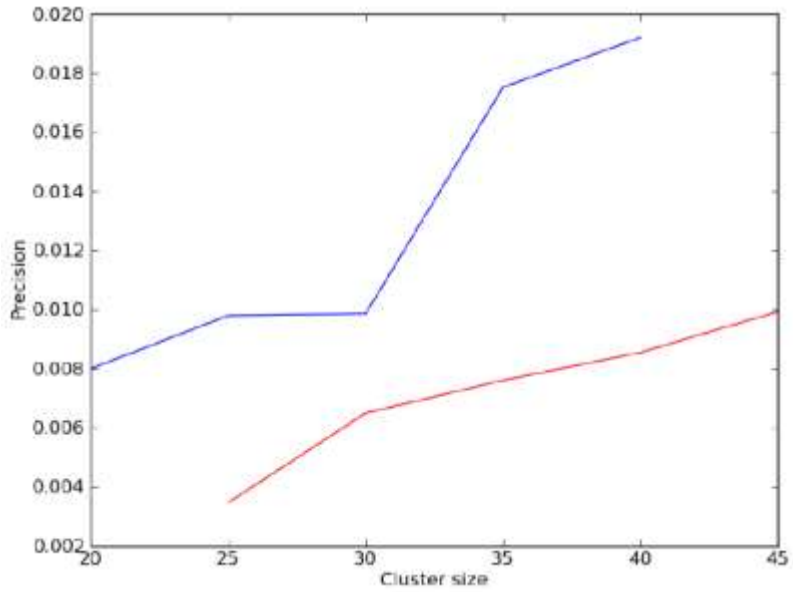

Fig 13. Precision

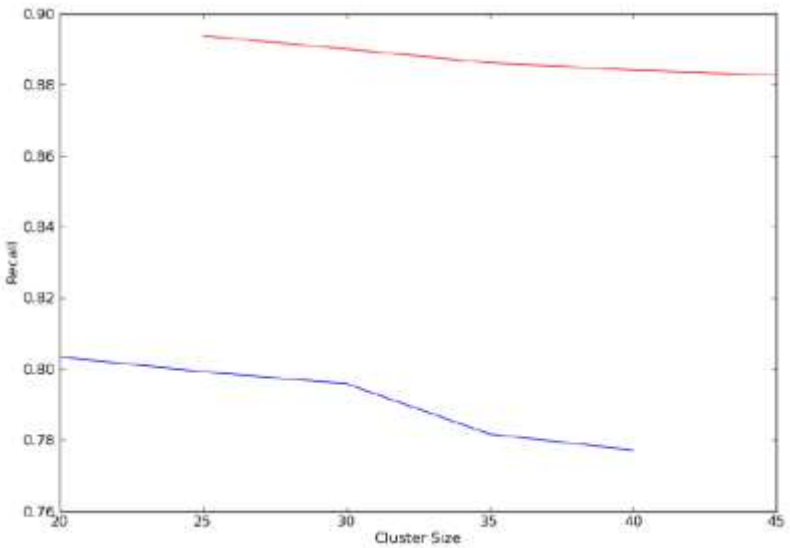

Fig 14. Recall

Chapter 4

Discussion

K-means is an unsupervised machine learning algorithm used for clustering data points into groups or clusters based on their similarities. When you apply the k-means algorithm to your data, you'll end up with a set of clusters, each containing a subset of your data points.Fuzzy C-means is a clustering algorithm that is similar to the k-means algorithm but allows for more flexible clusters. Rather than assigning each data point to a single cluster, the fuzzy c-means algorithm assigns each data point a membership value for each cluster, indicating the degree of membership in that cluster. This membership value is a measure of the similarity of the data point to the cluster center.

In this project we were able to combine both the clustering algorithm results and perform cascaded clustering.Cascaded clustering is a method of performing clustering in a hierarchical fashion, where the output of one clustering algorithm is used as the input to another clustering algorithm. The resulting clusters are organized in a cascaded structure where the output of each clustering algorithm is used as the input to the next.To evaluate the quality of the cascaded clustering results, we can use metrics such as silhouette score or Rand index. Silhouette score measures the quality of individual clusters, while the Rand index measures the similarity between the clusters at different levels of the hierarchy.Overall, cascaded clustering can be an effective method for performing clustering in a hierarchical manner, allowing for more fine-grained clustering of the data.

Using the method of clustering when we feed a question to the system, we get a result as a pool of 10 best answerers for that question.

After we get the results from our clustering we can further specify the category of the questions using the classification algorithm. As we saw in our results the best algorithm that works for classification is our Ensemble learning based stacked classifier.

The main benefit of a stacked classifier is that it can leverage the strengths of multiple base classifiers while mitigating their weaknesses. By combining the predictions of multiple classifiers, the stacked classifier can produce more accurate and robust predictions compared to a single classifier.

Another benefit of stacked classifiers is that they can be more resistant to overfitting. Since the base classifiers are trained on the same dataset, they may all be susceptible to the same biases and noise in the data. However, by combining the predictions of multiple classifiers, the stacked classifier can produce a more reliable prediction that is less affected by noise and biases.

Additionally, stacked classifiers can be more flexible in handling different types of data. By choosing a diverse set of base classifiers, the stacked classifier can adapt to different types of data and learn a more comprehensive representation of the data.

Chapter 5

Conclusion

The technological field is ever changing. New technologies come out every day that make our life easier. Just like the invention of the Q&A platforms made life easier by letting us get answers to specific questions from a community. As we have discussed before, the Free Rider problem causes a lot of loss to a website that thrives on community interaction. Users that frequently ask questions only interact as questioners, although StackOverflow is gamified and offers rewards for answering questions. A method is required to ensure a maximum number of users answer questions so that more community interaction is generated.

Therefore, for our project we developed an algorithm to find appropriate answerers to a question so that the numbers of Free Rider users are minimized. We first employ a simple clustering method using K - means and Fuzzy - C means. The questions in the StackOverflow dataset underwent this clustering so that we could find questions that are potentially similar and group them under the same umbrella. For more precision we developed a cascaded clustering model so that we could find better and more accurate users that could be interested in answering the question. This cascaded model was applied for both K - means and Fuzzy - C means clusters. After the process was over we found that K - Means was better.

We then employed classification mechanisms to reduce time for finding the answer pool. An ensemble learning model was used for finding a class. For the same we deployed it on the Streamlit server and developed a website for the model.

Chapter 6

Future Work

Within the scope of the work that is being proposed, we have polled users for their queries and provided them with the most comprehensive pool of answers we can. To add to the work that has been done, we may go to the StackOverflow system. If there is a recommendation system that proposes the top questions that the answerers should answer, our model can be put into that system, and the answerers will immediately receive the top questions to answer. In the event that this is not the case, it is possible to construct a recommendation system based on the model that was proposed.

Chapter 7

Implementation

Models:

```
import pandas as pd

import numpy as np

data=pd.read_csv("/content/drive/MyDrive/QueryResults (1).csv")

data.head(10)

data.columns

print(data.shape)

print(data.size)

df=data.drop(['DeletionDate','ViewCount','OwnerDisplayName','LastEditorUserId',
'LastEditorDisplayName',

    'LastEditDate', 'LastActivityDate','CommentCount', 'FavoriteCount', 'ClosedDate',
'CommunityOwnedDate',

    'ContentLicense'],axis='columns')

df.head(6)

print(df.shape)

df.dtypes

bool_series = df.isnull()

# filtering data

# displaying data only with team = NaN
```

```
df[bool_series]


df=df.fillna(0)

#df['AcceptedAnswerId']=df['AcceptedAnswerId'].fillna(0)

#df['AnswerCount']=df['AnswerCount'].fillna(0)

df.head()

questions= df[df['PostTypeId'] ==1]

questions=questions.drop(['ParentId'],axis="columns")

questions.shape

answers=df[df["PostTypeId"]==2]

answers.shape


from sklearn.model_selection import train_test_split

questions.columns

usedf = questions.drop(['CreationDate','Body','Title','Tags'],axis='columns')

usedf=usedf.astype(float)

usedf.dtypes

from sklearn.cluster import KMeans

from fcmeans import FCM

from yellowbrick.cluster import KElbowVisualizer

model = KMeans()
```

```
# k is range of number of clusters.

visualizer = KElbowVisualizer(model, k=(2,30), timings= True)

visualizer.fit(df.drop(['CreationDate','Body','Title','Tags'],axis='columns'))        # Fit data to visualizer

visualizer.show()

model = FCM(n_clusters=7) # we use two cluster as an example

model.fit(usedf.to_numpy()) ## X, numpy array. rows:samples columns:features

centers = model.centers

labels = model.predict(usedf.to_numpy())

import matplotlib.pyplot as plt

u_labels = np.unique(labels)

for i in u_labels:

    plt.scatter(usedf.to_numpy()[labels == i , 0] , usedf.to_numpy()[labels == i , 1] , label = i)

plt.legend()

plt.show()



from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters= 7)

label = kmeans.fit_predict(usedf.to_numpy())

import matplotlib.pyplot as plt

filtered_label0 = usedf.to_numpy()[label == 0]

#plotting the results
```

```
plt.scatter(filtered_label0[:,0] , filtered_label0[:,1])

plt.show()

u_labels = np.unique(label)

for i in u_labels:

    plt.scatter(usedf.to_numpy()[label == i , 0] , usedf.to_numpy()[label == i , 1] , label = i)

plt.legend()

plt.show()


usedf['cluster']=label

usedf.head()

ansdf=answers.drop(['CreationDate','Body','Title','Tags'],axis='columns')

ansdf.head()

usedf.rename(columns = {'Id':'QId'}, inplace = True)

ansdf.rename(columns={'ParentId':'QId'}, inplace=True)

def get_answerers(i):

  cluster_i=usedf[usedf['cluster']==i]

  cluster_i=usedf.drop(['cluster'],axis=1)

  kmeans_i = KMeans(n_clusters= 5)

  label_i= kmeans_i.fit_predict(cluster_i.to_numpy())

  answers_i = pd.merge(cluster_i, ansdf, on='QId', how='inner')

  answerers_i=np.array(answers_i['OwnerUserId_y'])
```

```
  return(answerers_i)
list=[]
for i in range(len(get_answerers(2))):
  if(i<=10):
    list.append(get_answerers(2)[i])
  else:
    break
print(list)


questions.head()
questions['class label']=label
touse=questions.drop(['CreationDate','Body','Title','Tags'],axis=1)
from sklearn.feature_selection import RFE
from sklearn.tree import DecisionTreeClassifier
rfe=RFE(estimator=DecisionTreeClassifier(),n_features_to_select=4)
X=touse.drop(['class label'],axis=1)
y=touse['class label']
X.head()
y.head()
X_new=rfe.fit_transform(X,y)
X_new
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X_new,y,test_size=0.2,random_state=1)
from keras.models import Sequential
from keras.layers import Dense
import numpy as np
model = Sequential()
model.add(Dense(15, input_dim=4, activation='relu'))
model.add(Dense(512, activation='relu'))
model.add(Dense(786, activation='relu'))
model.add(Dense(7, activation='softmax'))
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

model.summary()
model.fit(X_train,y_train,epochs=10,batch_size=32)

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import StackingClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X_new, y, test_size = 0.2, random_state = 42)
dtc =  DecisionTreeClassifier()
rfc = RandomForestClassifier()
knn =  KNeighborsClassifier()
from sklearn.model_selection import cross_val_score
```

```
clf = [dtc,rfc,knn]
for algo in clf:
    score = cross_val_score( algo,X_new,y,cv = 5,scoring = 'accuracy')
    print("The accuracy score of {} is:".format(algo),score.mean())
clf = [('dtc',dtc),('rfc',rfc),('knn',knn)]
lr = LogisticRegression()
stack_model = StackingClassifier( estimators = clf,final_estimator = lr)
# score = cross_val_score(stack_model,X_new,y,cv = 5,scoring = 'accuracy')
# print("The accuracy score of is:",score.mean())
```

Website:

```
import streamlit as st
import pickle
from tensorflow import keras
from keras.models import load_model
import random
import numpy
import joblib
import os
# path=os.getcwd()
# path=os.path.join(path,'model2.pkl')
# with open(path, 'rb') as file:
#     model2 = pickle.load(file)

model2=joblib.load('modelsin1.pkl')
print(type(model2))
#model=pickle.load(r"C:\Users\anegh\OneDrive\Desktop\proj\model2.pkl")
st.header("StackOverflow")
pressure=st.text_input("Title:")
temp=st.text_input("Body:")
if (pressure is not None) and (temp is not None):
    num1=random.randint(0,7)
```

```
num2=random.randint(0,7)
num3=random.randint(0,7)
num4=random.randint(0,7)
if st.button("predict"):
    inp = numpy.array([[float(num1), float(num2),float(num3),float(num4)]])
    out = model2.predict(inp)

    st.subheader(d.get(int(out)))
```

References

1. Lada Adamic et al. Knowledge Sharing and Yahoo Answers: Everyone knows Something, WWW 2008

2. Eugene Agichtein,Carlos Castillo et al. Finding High Quality Content in Social Media WSDM 2008

3. Jun Zhang, Mark S Ackerman Lada Adamic Expertise Networks in Online Communities:Structure and Algorithms WWW 2007

4. Lada Adamic Expertise Sharing Dynamics in Online Forums

5. Pawel Jurczyk, Eugene Agichtein Discovering Authorities in Question Answer Communities by Using Link Analysis CIKM 2007

6. Pawel Jurczyk, Eugene Agichtein HITS on Question

7. Kasturi, Iyer, Finding Answerers on Yahoo! Answers

8. Ponzanelli, L., Mocci, A., Bacchelli, A., & Lanza, M. (2014, October). Understanding and classifying the quality of technical forum questions. In *2014 14th International Conference on Quality Software* (pp. 343-352). IEEE.

9. Zhang, H., Wang, S., Chen, T. H., Zou, Y., & Hassan, A. E. (2019). An empirical study of obsolete answers on stack overflow. *IEEE Transactions on Software Engineering*, *47*(4), 850-862.

10. Zhou, G., Lai, S., Liu, K., & Zhao, J. (2012, October). Topic-sensitive probabilistic model for expert finding in question answer communities. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 1662-1666).