Detection of Duplicate Question Pairs

Rohit Kumar 21MT0339
Prateek Singh 21MT0288
SambaKanti Samanta 21MT0353

April 3, 2022

1 Abstraction

We demonstrate a solution to the problem of identifying duplicate questions. We concentrate on a recent dataset of question pairs annotated with binary paraphrase labels and show that the homogeneous ensembling model outperforms most other more complicated neural architectures. Furthermore, when the model is pretrained on a noisy dataset of automatically collected question para- phrases, it results in a higher performance on the task

2 Introduction

Question and Answering sites like Yahoo and Google Answers ex- isted over a decade however they failed to keep up the content value of their topics and answers due to a lot of junk informa- tion posted; thus their user base declined. On the other hand, Quora is an emerging site for the quality content, launched in 2009 and as of 2019, it is estimated to have 300 million active users1. Quora has 400,000 unique topics2 and domain experts as its user so that the users get the first-hand information from the experts in the field. With the advancing repository of the knowledge base, there is a need for Quora to preserve the belief of the users, maintain the content quality, by discarding the junk, duplicate and insincere information. Quora has successfully prevail over this challenge by organizing the data effectively by using modern data science approach to eliminate question duplication.

As reported by Quora there has been many duplicate questions asked by users which actually means the same thing and creates a confusion as people has answered to the same question in different locations . For the instance , suppose take these two questions- "How old are you ? "&"What is your birthday?". Although these questions have no word in common but have the same intent.Hence, Quora would like to group these similar questions together so that all the answers could be presented at the same place and a better user experience overall. We have used the dataset provided by Quora to create a model which can identify duplicate questions posted by the users.

We need to create a machine learning software to understand that specific language, there are plenty of NLP machine learning tutorials out there.Once we have an NLP understanding software you can then run through all the questions and push the output data to another machine learning software, One that can identify contextual similarity patterns from your data.

3 Problem Statement

Identify the duplicate pair of questions using semantic similarity between sentences from a given Quora dataset

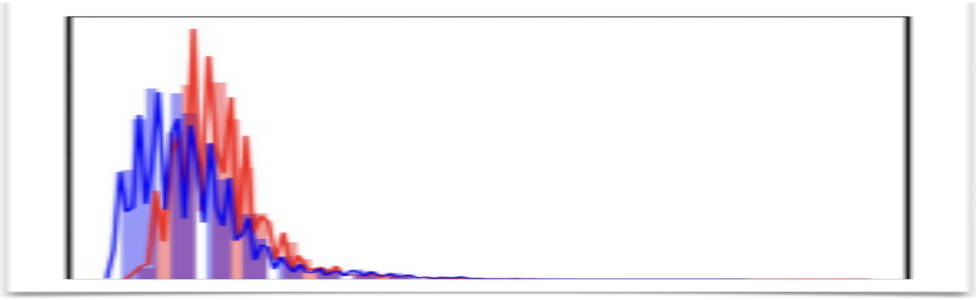
4 Methodology

4.1 Feature Selection

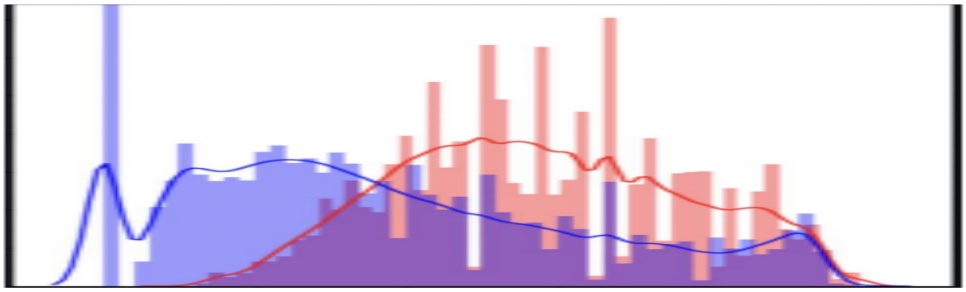
Basic features :

- freq_qid1 = Frequency of qid1's.number of times question1 occur..
- freq_qid2 = Frequency of qid2's.number of times question1 occur.
- q1len = Length of q1.
- q2len = Length of q2.
- q1_n_words = Number of words in Question 1.
- q2_n_words = Number of words in Question 2
- word_Common = (Number of common unique words in Question 1 and Question 2).
- word_Total =(Total num of words in Question 1 + Total num of words in Question 2)
- word_share = (word_common)/(word_Total)
- freq_q1+freq_q2 = sum total of the frequency of qid1 and qid2
- freq_q1-freq_q2 = absolute difference of frequency of qid1 and qid2

But when we do our analysis on these features do not find a strong separation on the result. For example if we consider common words feature,



We can see that it is almost overlapping. Let's see another feature word_share : We can check from below that it is overlapping a bit, but it is giving some classifiable score for dissimilar questions.



So, we need to consider some advanced features into our feature set. Let us consider some features based on tokens and stop words and word count.

4.2 Creating Token Features :

- cwc_min : Ratio of common_word_count to min length of word count of Q1 and Q2

$$cwc\_min = \frac{common\_word\_count}{(min(len(q1\_words), len(q2\_words))} \tag{1}$$

- cwc_max : Ratio of common_word_count to max length of word count of Q1 and Q2

$$cwc\_max = \frac{common\_word\_count}{(max(len(q1\_words), len(q2\_words))} \tag{2}$$

- csc_min : Ratio of common_stop_count to min length of stop count of Q1 and Q2

$$csc\_min = \frac{common\_stop\_count}{(min(len(q1\_stops), len(q2\_stops))} \tag{3}$$

- ctc_min : : Ratio of common_token_count to min lenghth of token count of Q1 and Q2

$$ctc\_min = \frac{= common\_token\_count}{(min(len(q1\_tokens), len(q2\_tokens))} \tag{4}$$

- ctc_max : : Ratio of common_token_count to max lenghth of token count of Q1 and Q2

$$ctc\_max = \frac{= common\_token\_count}{(max(len(q1\_tokens), len(q2\_tokens))} \tag{5}$$

- last_word_eq : Check if last word of both questions is equal or not

$$last\_word\_eq = last\_word\_eq = int(q1\_tokens[-1] == q2\_tokens[-1]) \tag{6}$$

- first_word_eq : Check if First word of both questions is equal or not

$$first\_word\_eq = first\_word\_eq = int(q1\_tokens[0] == q2\_tokens[0]) \tag{7}$$

- abs_len_diff : Absolute length difference

$$abs\_len\_diff = abs(len(q1\_tokens) minus len(q2\_tokens)) \quad (8)$$

4.3 Creating Fuzzy Features :

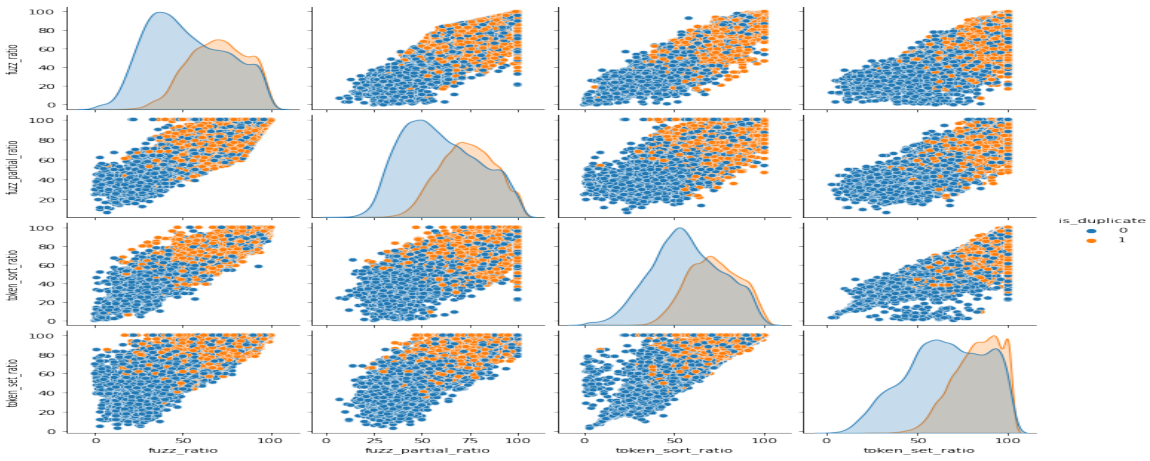
1. fuzz_ratio : Here comes the amazing part. Fuzz ratio rely upon the Levenshtein distance. Commonly saying if the corresponding changes required from one sentance to become other is more, fuzz ratio will be small. ie, fuzz ratio will be almost similar for most similar words. eg: p1 = "mumbai is a great place" p2 = "mumbai is a good place"fuzz ratio = 91

2.fuzz_partial_ratio : In a few cases fuzz ratio cannot solve the issue. fuzz.ratio("YANKEES", "NEW YORK YANKEES") = 60 fuzz.ratio("NEW YORK METS", "NEW YORK YANKEES") =75 Both p1 and p2 mean the same. But their fuzz ratio can be small. So we will find the ratio for partial sentences and it will be high. In such a case, it is called as a fuzz partial ratio. fuzz.partial_ratio("YANKEES", "NEW YORK YANKEES") = 60

3.token_sort_ratio: In some other cases even fuzz partial ratio will fail. fuzz.partial_ratio("LSG vs RCB","RCB vs LSG") = 72 Actually both the sentence have the same value. But the fuzz ratio gives a low result. So a better approach is to sort the tokens and then apply fuzz ratio. fuzz.token_sort_ratio("LSG vs RCB","RCB vs LSG") =100

4.token_set_ratio: There is a other type of fuzz ratio which helps even I cases where all above fails. It is the token set ratio. For that we have to first find the following: t0 = find the intersection words of sentance1 and sentance2 and sort them. t1= t0 + rest of tokens in sentance1 t2= t0 + rest of tokens in sentance2 tocken_set_ratio = max(fuzz_ratio(to,t1),fuz_ratio(t1,t2),fuzz_ratio(t0,t2))

5.longest_substr_ratio : Ratio of length longest common substring to min length of token count of Q1 and Q2. r1→hai, today is a good day r2→No, today is a bad day Here longest common substring is "today is a". So we have longest_substring_ratio = 3 / min(6,6) = 0.5 longest_substr_ratio = len(longest common substring) / (min(len(r1_tokens), len(r2_tokens))



4.4 Used Distance Metrics :

- Euclidean Distance
- Manhattan Distance
- Hamming Distance
- Cosine Similarity

1. Euclidean Distance : The Euclidean distance is a straight-line distance among vectors. For the two vectors x and y, this will be computed as follows:

$$EUC(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

with respect to to the Cosine and Jaccard similarity, Euclidean distance is not used very often in the context of NLP applications.It is suitable for non-stop numerical variables. Euclidean distance is not scale invariant, therefore scaling the data prior to computing the distance is recommended. Additionally, Euclidean distance multiplies the effect of redundant information in the dataset. If I had five variables which are heavily correlated and we take all five variables as input, then we would weight this redundancy effect by five.

((1,2,3)(4,5,6))
These are the two sample points which we will be using to calculate the different distance functions. Let's now calculate the Euclidean Distance between these two points:

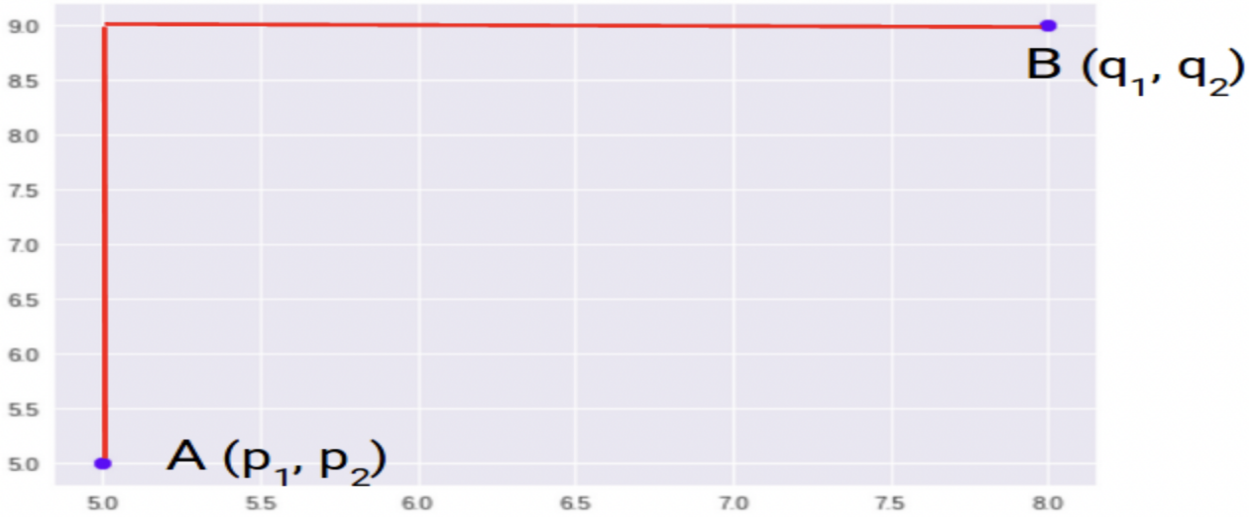
Euclidean distance b/w (1,2,3) and (4,5,6) is 5.1961524

2. Manhattan Distance :

The Manhattan Distance between two points, a and b, with n dimensions is calculated as:

$$\text{Manhattan distance} = \sum_{i=1}^{n} |x_i - y_i|$$

In many ML programs Euclidean distance is the metric of choice. However, for excessive dimensional statistics Manhattan distance is greatest because it yields greater sturdy results.Different from Euclidean distance is the Manhattan distance, additionally called 'cityblock', distance from one vector to another. You can believe this metric as a manner to compute the gap among factors whilst you aren't capable of undergo buildings.




```
Manhattan Distance b/w (1, 2, 3) and (4, 5, 6) is:  9
```

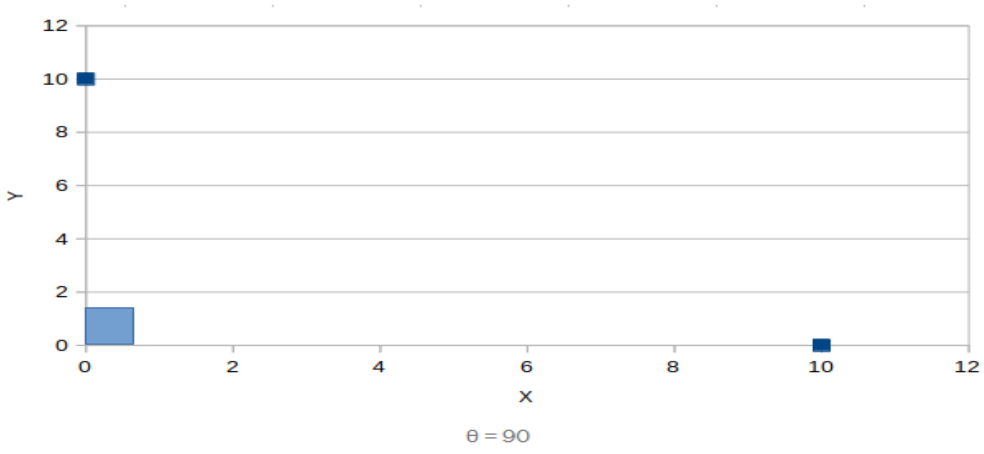
3. Hamming Distance : Hamming distance is a way of similarity measure for comparing two binary data strings. While comparing two binary strings of equal length, Hamming distance is the number of bit positions in which the two bits are different.

The Hamming distance between two strings, p and q is denoted as d(p,q). with respect to calculate the Hamming distance between two strings, and, we perform their XOR operation, (p XOR q), and then count the total number of 1s in the resultant string. Assume there are two strings 11011001 and 10011101. 11011001 10011101 = 01000100. Since, this contains two 1s, the Hamming distance, d(11011001, 10011101) = 2.

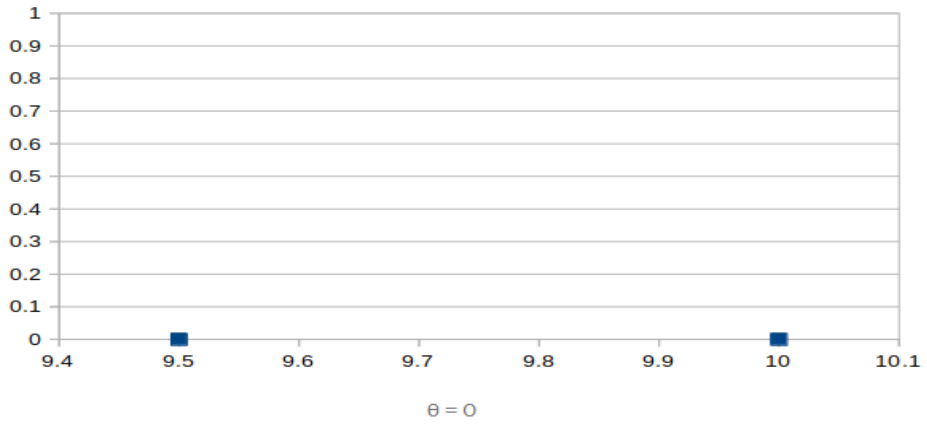
string1 = 'euclidean'
string2 = 'manhattan'

hamming distance between euclidean and manhattan is 7.0

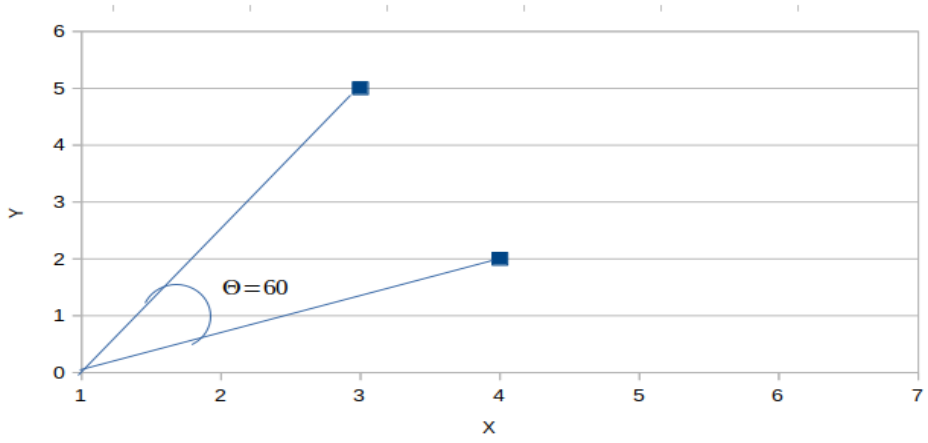
4. Cosine Similarity : Cosine distance Cosine Similarity metric is a way of similarity measure mainly used to find similarities between two data points. When the cosine distance between the data points increases, then cosine similarity, or the amount of similarity decreases, and vice versa. Therefore, Points closer to each other are more similar than points that are far away from each other. Cosine similarity is given by Cos , and cosine distance is 1- Cos .
Example:-



In the above image, there are two data points shown in blue, the angle between these points is 90 degrees, and Cos 90 = 0. Therefore, the shown two points are not similar, and their cosine distance is 1 — Cos 90 = 1.



Now if the angle between the two points is 0 degrees in the above figure, then the cosine similarity, Cos 0 = 1 and Cosine distance is 1- Cos 0 = 0. Then we can interpret that the two points are 100 percent similar to each other.



In the above figure, imagine the value of to be 60 degrees, then by cosine similarity formula, Cos (60) =0.50 and Cosine distance is 1.00- 0.50 = 0.50. Therefore the points are 50 percent similar to each other.

5 Dataset : Quora Question pair dataset

Reference : https://www.kaggle.com/c/quora-question-pairs

- Available Columns: id, qid1, qid2, question1, question2, is_duplicate.
- Class labels: 0, 1
- No. of non-duplicate data points is 255027
- No. of duplicate data points is 149263
- 36.92 percent of question pairs are duplicates and 63.08 percent of questions pair non-duplicate.
- The above ratio denotes that this is an imbalanced dataset but the imbalance here is quite acceptable.
- Most of the questions are repeated very few times. Only a few of them are repeated multiple times.
- One question is repeated 157 times which is the max number of repetitions.
- There is no strict latency requirement
- We would like to have interpretability but it is not absolutely mandatory.
- The cost of misclassification is medium.
- Both classes (duplicate or not) are equally important.

Average number of characters in question1	59.57
Minimum number of characters in question1	1
Maximum number of characters in question1	623
Average number of characters in question2	60.14
Minimum number of characters in question2	1
Maximum number of characters in question2	1169

The following diagram shows an actual snapshot of the few rows from the dataset:

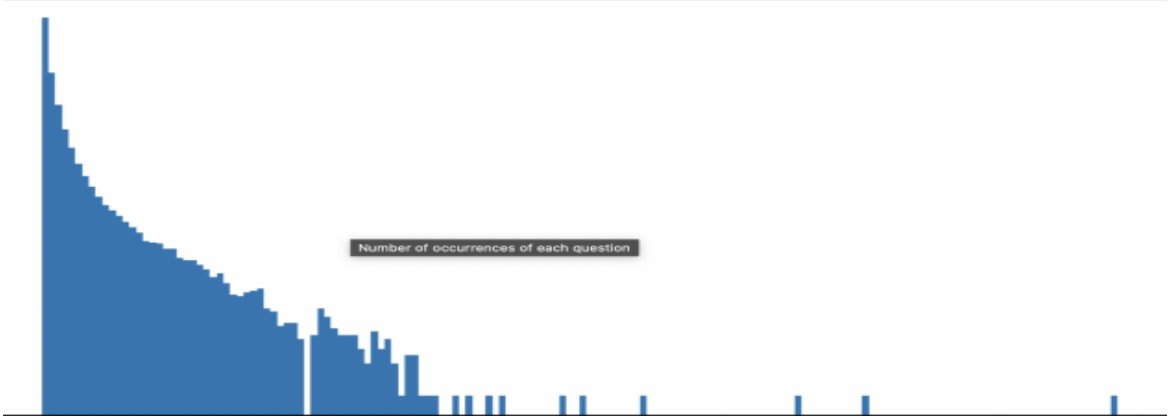
	id	qid1	qid2	question1	question2	is_duplicate
398782	398782	496695	532029	What is the best marketing automation tool for...	What is the best marketing automation tool for...	1
115086	115086	187729	187730	I am poor but I want to invest. What should I do?	I am quite poor and I want to be very rich. Wh...	0
327711	327711	454161	454162	I am from India and live abroad. I met a guy f...	T.I.E.T to Thapar University to Thapar Univers...	0
367788	367788	498109	491396	Why do so many people in the U.S. hate the sou...	My boyfriend doesnt feel guilty when he hurts ...	0
151235	151235	237843	50930	Consequences of Bhopal gas tragedy?	What was the reason behind the Bhopal gas trag...	0

Exploring further into the data, we can find some examples of question pairs that mean the same thing, that is, duplicates, as follows:

How does Quora quickly mark questions as needing improvement?	Why does Quora mark my questions as needing improvement/ clarification before I have time to give it details? Literally within seconds...
Why did Trump win the Presidency?	How did Donald Trump win the 2016 Presidential Election?
What practical applications might evolve from the discovery of the Higgs Boson?	What are some practical benefits of the discovery of the Higgs Boson?

We figured out that Question 1 and question 2 are roughly the same average characters, though we have more extremes in question 2.

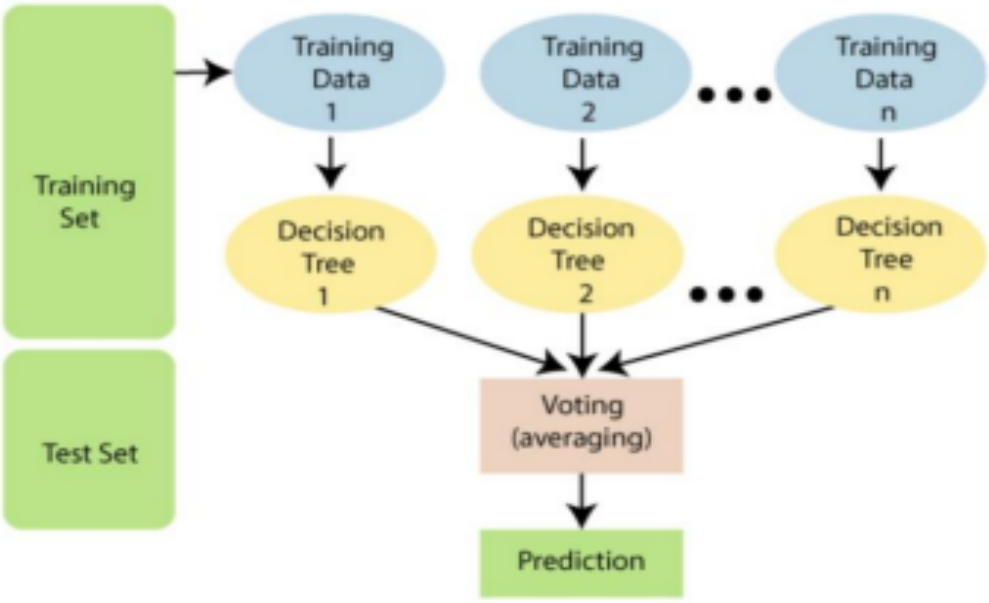
Number of occurrence of each question :



6 Models Implemented /Result and Analysis :

6.1 Random Forest :

It builds decision trees on different samples and takes their majority vote for classification and average in case of regression



The Working process can be explained in the diagram:

- Step-1: Select random K data points from the training set.
- Step-2: Build the decision trees associated with the selected data points (Subsets).
- Step-3: Choose the number N for decision trees that you want to build.
- Step-4: Repeat Step 1 & 2.
- Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Accuracy obtained : 78-80%

		Actual0	Actual1
Confusion matrix :	pred0	[3213,	521]
	pred1	[761,	1505]

Disadvantages of Random Forest

1. Complexity: Random Forest creates a lot of trees (unlike only one tree in case of decision tree) and combines their outputs. By default, it creates 100 trees in Python sklearn library. To do so, this algorithm requires much more computational power and resources. On the other hand decision tree is simple and does not require so much computational resources.

2. Longer Training Period: Random Forest require much more time to train as compared to decision trees as it generates a lot of trees (instead of one tree in case of decision tree) and makes decision on the majority of votes.

Advantages of Random Forest

1. Random Forest is based on the bagging algorithm and uses Ensemble Learning technique. It creates as many trees on the subset of the data and combines the output of all the trees. In this way it reduces overfitting problem in decision trees and also reduces the variance and therefore improves the accuracy.

2. Random Forest can be used to solve both classification as well as regression problems.

3. Random Forest works well with both categorical and continuous variables.
]
4. Random Forest can automatically handle missing values.

6.2 XGBoost :

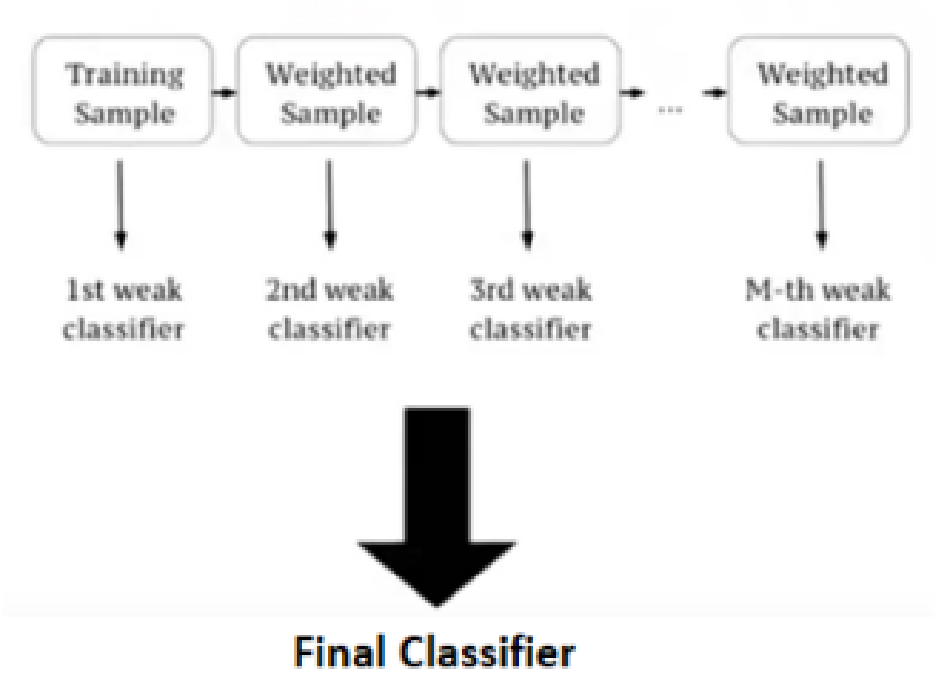
XGBoost provides a wrapper class to allow models to be treated like classifiers or regressors in the scikit-learn framework. This means we can use the full scikit-learn library with XGBoost models. The XGBoost model for classification is called XGBClassifier. XGBoost has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from overfitting. That is why, XGBoost is also called regularized form of GBM (Gradient Boosting Machine).

While using Scikit Learn libarary, we pass two hyper-parameters (alpha and lambda) to XGBoost related to regularization. alpha is used for L1 regularization and lambda is used for L2 regularization.

XGBoost utilizes the power of parallel processing and that is why it is much faster than GBM. It uses multiple CPU cores to execute the model.

While using Scikit Learn libarary, nthread hyper-parameter is used for parallel processing. nthread represents number of CPU cores to be used. If you want to use all the available cores, don't mention any value for nthread and the algorithm will detect automatically.

XGBoost has an in-built capability to handle missing values. When XGBoost encounters a missing value at a node, it tries both the left and right hand split and learns the way leading to higher loss for each node. It then does the same when working on the testing data.



Accuracy obtained : 76-80 %

Confusion matrix : [3145, 648]

[790, 1417]

Advantages :
1.Regularization
2.parallel Processing
3.Can handle missing values
4.Cross Validation
5.Effective tree pruning

Disadvantages: XGBoost does not perform so well on sparse and unstructured data. A common thing often forgotten is that Gradient Boosting is very sensitive to outliers since every classifier is forced to fix the errors in the predecessor learners. The overall method is hardly scalable.

Well XGBoost (as with other boosting techniques) is more likely to overfit than bagging does (i.e. random forest) but with a robust enough dataset and conservative hyperparameters, higher accuracy is the reward

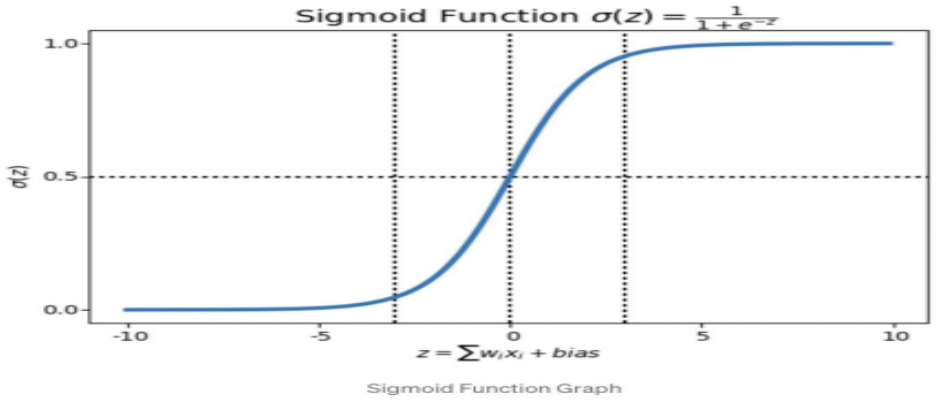
In Xgboost, you have to manually create dummy variable/ label encoding for categorical features before feeding them into the models. Catboost/Lightgbm can do it own their own, you just need to define categorical features names or indexes.

Training time is pretty high for larger dataset, if you compare against catboost/lightgbm.

Few Observations : We clearly see that Random forest is performing better than XGBoost in this scenario. From the confusion matrix we can infer that in case of false positives random forest do a better job. 761 ¡ 790 (smaller is better). Let us try few other popular machine learning models.

6.3 Logistic regression :

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1.In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.



Sigmoid Function Graph

Accuracy Obtained : 72-73%

Confusion Matrix : [2048, 480]

[623, 849]

Disadvantages : The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables. It can only be used to predict discrete functions. Hence, the dependent variable of Logistic Regression is bound to the discrete number set. Non-linear problems can't be solved with logistic regression because it has a linear decision surface. Linearly separable data is rarely found in real-world scenarios.

6.4 KNN :

K Nearest Neighbour or KNN algorithm falls under the Supervised Learning category and is used for classification and regression. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm
The K-NN working can be explained on the basis of the below algorithm:

◦ Step-1: Select the number K of the neighbors

◦ Step-2: Calculate the Euclidean distance of K number of neighbors

◦ Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

◦ Step-4: Among these k neighbors, count the number of the data points in each category.

◦ Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

◦ Step-6: Our model is ready.

Accuracy Obtained : 71-72%

Confusion Matrix : [1993, 535]

[641, 831]

Advantages of KNN Algorithm:

◦ It is simple to implement.

◦ It is robust to the noisy training data

◦ It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

◦ Always needs to determine the value of K which may be complex some time.

◦ The computation cost is high because of calculating the distance between the data points for all the training samples.

6.5 SVM :

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.

However, primarily, it is used for Classification problems in Machine Learning.The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.

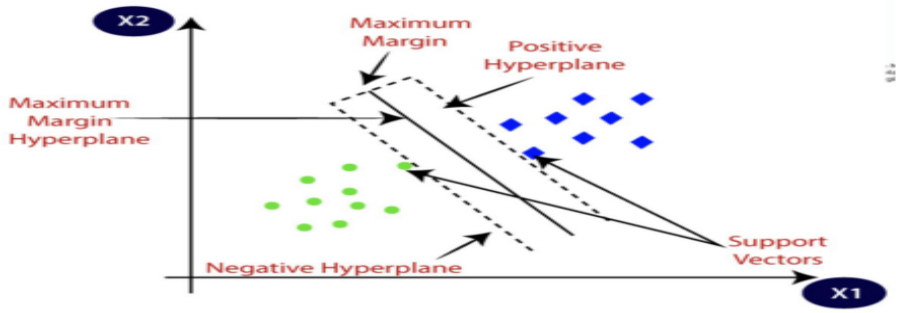
This best decision boundary is called a hyperplane.SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

6.6 SVM Kernel:

The SVM kernel is a function that takes low dimensional input space and transforms it into higher-dimensional space, ie it converts not separable problem to separable problem. It is mostly useful in non-linear separation problems.

Simply put the kernel, it does some extremely complex data transformations then finds out the process to separate the data based on the labels or outputs defined.



Accuracy Obtained : 75% approx.

Confusion Matrix : [1996, 532]

[486, 986]

Advantages:

• SVM performs well quite properly at the same there's a clean margin of separation among classes.

• SVM is extra powerful in excessive dimensional spaces.

• SVM is powerful in instances in which the variety of dimensions is extra than the variety of samples.

• SVM is quite space efficient

Disadvantages:

• SVM set of rules isn't always appropriate for big statistics sets.

• SVM does now no longer carry out thoroughly whilst the statistics set has extra noise i.e. goal classes are overlapping.

• In instances in which the variety of capabilities for every statistics factor exceeds the variety of schooling statistics samples, the SVM will underperform.

• As the assist vector classifier works with the aid of using setting statistics points, above and underneath the classifying hyperplane there's no probabilistic reason behind the classification.

7 Conclusion :

After testing out 5 models we see that on average Random Forest model shows better result overall most of the cases .

We can conclude that the models based on homogeneous ensembling feature works better on the dataset that we have taken for out project. A universal rule of thumb:

• Homogeneous ensembles use the same feature selection with a variety of data and distribute the dataset over several nodes.

• Heterogeneous ensembles use different feature selection methods with the same data

We are getting around 80 percent accuracy with random forest which is quite good for a ML model, from here on adding new features would result in exponentially slow improvement on the accuracy of the model.

The model could do better if we include few more advanced features like distance features in the future.

Another way, Quora could achieve a better by pre-processing the original question pair dataset. Since knowing the context in which question is asked, a proper replacement of some of the pronouns can be done, and higher accuracy can be achieved.

For example, pronoun like us, we, they can be replaced if the topic under which question exists thus replacing it with their relative context like “American,” “ Programmers ” and “ Prisoners ’ etc. during the pre- processing data stage can help achieve a better result.

As we are unaware in which context questions were asked we could not do such preprocessing on the original dataset.

Increasing the batch sizes usually yields better results, as the task gets harder. It is more difficult to identify the correct duplicate question out of a set of 100 questions than out of a set of only 10 questions.

So it is advisable to set the training batch size as large as possible. I trained it with a batch size of 20000 on 8 GB GPU memory.

8 References :

https://www.sciencedirect.com/science/article/abs/pii/S0031320399000059

https://neptune.ai/blog/tokenization-in-nlp

https://towardsdatascience.com/understanding-random-forest-58381e0602d2

https://machinelearningmastery.com/develop-first-xgboost-model-python-scikit-learn/