International Journal of Future Generation Communication and Networking
Vol. 13, No. 1, (2020), pp. 849 - 868

Quora Question Pairs Similarity Using Logistic Regression and Support Vector Machine

Dr. P V Rama Raju[1], G. Naga Raju[2], N. Nikhil[3], M.Hemanth Gupta[3], Chandan Akella[3], S.Kumara Siddarth[3]

[1]Professor, [2]Asst. Professor, [3]B. Tech Students

[1, 2, 3]Department of ECE, SRKR Engineering College (A), Bhimavaram, India.

Corresponding Author mail ID: pvrraju50@gmail.com[1] , bhanu.raj.nikhil@gmail.com[2]

Abstract:

Quora is an application in which, questions of several aspects are being posted .The questions that are posted are answered by persons who are having good knowledge about that corresponding aspect. Millions of questions are being posted in quora every day and they are all not necessarily identical. This paper explores the task of Natural Language Understanding by exploring the questions in the Quora dataset .We explored the dataset and used various machine learning models(linear and tree based models). XgBoost models, Support Vector Machine(SVM) models, Logistic regression models are used with TF-IDF and Word2vec algorithms to identify the similarity between questions that are posted on Quora .Our finding was that the TF-IDF neural network along with XgBoost has stood out by giving the best performance, outdoing the other complicated models.

Keywords: *TF-IDF, Question pairs, SVM, Logistic regression, Word2vec, XgBoost*

1. Introduction

The dataset provided by quora contains pairs of questions and the task is to determine whether those pairs of questions are similar or not, that is whether they have the same meaning or not.

Quora is a question-and-answer site where questions are asked, answered, edited and organized by its community of users .Duplicate questions are common in these kinds of platforms particularly as the number of questions asked increases. The main aim of this work was to apply various Natural Language Processing (NLP) concepts for feature engineering from the given dataset and apply and compare some machine learning models such as K- Nearest neighbor, Decision Tree, Random Forest, Extra Trees, AdaBoost and Xgboost to predict the similarity. We acquired a highest accuracy of 86.26% with Extra Trees [1]. .The task at hand requires model to be proficient in Natural Language Understanding(NLU) so that the model is able to mimic the humans in terms of layman language in the closest way possible .The task of determining whether the two sentences have the same meaning or not requires the model to capture lexical meaning(vocabulary) and the syntactic meaning(structure) of the sentences .This paper presents the comprehensive set of machine learning models and shows their performance on the dataset .We used simple linear models as our basic foundation .We built and tested Support Vector Machines(SVM),gradient boosting algorithms and several neural networks .Natural language processing (NLP) is a theory-motivated range of computational techniques for the automatic analysis and representation of human language[4].TF-IDF stands for (term frequency-inverse document frequency) is discussed in examining the relevance of key-words to documents in corpus. The study is focused on how the algorithm can be applied on number of documents. First, the working principle and steps which should be followed for implementation of TF-IDF are elaborated [11].

The problem at hand that is identifying similar questions is a binary classification problem on strings of varying lengths .We represents these sentences as numerical inputs so that the learning algorithms can work

ISSN: 2233-7857 IJFGCN

on them. The widely used method is to use hand engineered feature engineering .This method used along with the bag-of-words model enhances the performance.

NumPy , Pandas ,distances ,Seaborn ,Matplotlib ,Word cloud, FuzzyWuzzy ,Natural Language Toolkit(NLTK) etc. are some of the packages that are being used in this model ."Pandas" is a fast and flexible software library, generally used in python programming for manipulation and analysis of numerical data .'NumPy' is a scientific computing python fundamental package that is used for processing a powerful N-dimensional array object ,sophisticated (broadcasting) functions ,useful linear algebra, Fourier transform, and random number capabilities. 'Seaborn' is a Python library for data visualization, with matplotlib as the basis. 'Matplotlib' is a Python 2D plotting library that produces publication quality figures in a variety of paper formats and interactive environments on all platforms. 'Word cloud' is a method of text visualization in which the importance and usage of each word or tag is represented by the differences in font colors and sizes .'FuzzyWuzzy 'is a Python library used mainly used for string matching .It contains a process that find the strings that maintain a similar pattern .'NLTK' provides easy-to-use interfaces to over 50 corpora and lexical resources such as Word Net, as well as a suite of word processing libraries for classification, tokenization, stemming, markup, analysis and semantic reasoning. , wrappers for industrial grade NLP libraries , and an active discussion forum.

2.Related works

Recent works on the Quora question pair similarity is presented below.

Dr. P. V. Ramaraju, et al. [2], Detection of pests within the paddy fields could be a considerable challenge in the agriculture field, so effective measures ought to be developed to combat the infestation and minimizing the utilization of pesticides. The techniques of image analysis Hue Saturation Intensity (HSI) broadly applied to farms and plants. The present paper extends the employment of various image process techniques to discover pests and classify those using neural networks. The neural network is trained using feature extraction techniques to leaf and pest pixels. Finally, pests are detected also the name, lifespan of the pest, disease caused by the pest and the medicine to eradicate the pests are displayed.

Dr. P. V. Ramaraju, et al. [5], the main perspective to invent the “Stock Analysis” is to manage the list of products and to calculate the efficiency of the growth of business .This all information is stored in the cloud to predict the growth rate of an organization

.By using objects, Relation-ships, Visual Force Page, Apex Code .Gone are those days when a shop owner used to manage all his sales and accounts on paper. By this procedure the Manager can track the sales of the items from its shop. The system now gives a clear picture to the manager about the total sales and items available in the stock and will be the best Secured Cloud Application.

Asma Ben et al. [6], One of the challenges in large-scale information retrieval (IR) is developing fine-grained and domain-specific methods to answer natural language questions. One of the promising tracks investigated in QA is mapping new questions to formerly answered questions that are “similar”.

Shiyao Xu et al. [7], in this paper, we propose an ensemble model which based on both word and character level neural networks such as the convolutional neural network (CNN). Our model takes 10-fold cross-validation into account to improve the generalization ability. In the evaluation of the CCKS 2018 shared task three, our model achieves the F1 score of 0.85085 for the opening test data which ranks the second.

Dr. P. V. Ramaraju, et al. [8], Speech has been the most commanding and convenient method of communication. However many problems appears because of speech miscommunications. Hence, in this paper, Speech was divided by statistic method, KNN (K-Nearest Neighbor), which separates vowels from consonants. Later the trained KNN divider was tested using TIMIT test database.

Thales A.P. West et al. [9], In 2018, New Zealand announced an ambitious effort to plant one billion trees by 2028 as part of its climate change mitigation plan and Paris Agreement targets for 2030 and 2050.We identified spatially-explicit drivers of forest gain and the locations most likely to experience afforestation in the country using two distinct spatial modeling frameworks: logistic regressions and artificial neural networks (ANN).These results indicate an overall agreement between the two modelling approaches, despite substantial methodological differences. T. Sridevi, P. Mallikarjuna Rao, P V Ramaraju and G. Nagaraju,[10], Biometric fraud is an area of increasing concern, as the number of deployed biometric systems increases and fraudsters become aware of the potential to compromise them. This paper particularly handles how to incorporate cryptography and steganography in biometric applications.

G. Naga Raju, Dr. P. V. Ramaraju, et al. [12],Understanding the loss of biodiversity and reduction of carbon sequestration capacity that results from deforestation becomes much more difficult. In this paper, color segmentation algorithm is used on image and area of forest in a place over different year is calculated. In this paper, text to speech conversion algorithm is used to give out the result. Simulation is carried by using Matlab R2015a software.

A.Brenning, [13], The predictive power of logistic regression, support vector machines and bootstrap-aggregated classification trees (bagging, double-bagging) is compared using misclassification error rates on independent test data sets. Based on a resampling approach that takes into account spatial autocorrelation, error rates for predicting “present” and “future” landslides are estimated within and outside the training area. The evaluation outside the training area reveals that tree-based methods tend to over fit the data. Peng Wan, et al. [14], Under some reasonable conditions, the addressed networks have $(2m+1)n$ equilibrium points. $(m+1)n$ of which are locally asymptotically stable, and the others are unstable. The attraction basins of the locally asymptotically stable equilibrium points are given in the form of hyper spherical regions. The theoretical results and illustrative example indicate that the activation functions improve the storage capacity of neural networks significantly.

Dr. P. V. Ramaraju, et al. [15], The endeavor hopes to give a sensible response for the traffic signal structure to deny the standard sign timings during emergency regularly .The endeavor uses a microcontroller of 8051 family that is interfaced with the IR sensors and photodiodes balanced in discernable pathway diagram over the store for seeing the thickness. The thickness is surveyed in three stand-out ways low, medium and high as appeared by which the timings are appropriated for sign. The managing supplanted is done using RF advance.

Keith Wurtz, et al. [16], Aspects of the logistic regression procedure that are necessary to evaluate models are presented and discussed with an emphasis on cutoff values and choosing the appropriate number of candidate predictor variables. In order to demonstrate the process of conducting a logistic regression analysis, models are generated using educational background measures .Topics covered include setting up the database, dummy coding, data reduction, multi collinearity, missing cases, setting the cutoff value, interpretation of the results, selecting a model, and the interpretation of odds ratios when they are negative In Lee,et al.[17], Machine learning holds great promise for lowering product and service costs, speeding up business processes, and serving customers better. We then discuss the trade-off between the accuracy and interpretability of machine-learning algorithms, a crucial consideration in selecting the right algorithm for the task at hand.

Iian Rice,et al.[18], Popular dimension reduction and visualization algorithms rely on the assumption that input dissimilarities are typically Euclidean, for instance Metric Multidimensional Scaling, t-distributed Stochastic Neighbor Embedding and the Gaussian Process Latent Variable Model. It is well known that this assumption does not hold for most datasets and often high-dimensional data sits upon a manifold of unknown global geometry.

3. The Proposed Methodology

3.1 Procedure:

1) We did an experimental analysis on the data of the Quora question pairs by performing operations such as finding the number of different questions, checking for duplicates, the number of occurrences of questions, etc.
2) Features such as the fuzz ratio, the fuzz partial ration, the longest common substring, etc were extracted.
3) After performing the functionality extraction, certain visualization techniques such as pair plotting, violin plotting, TSNE, etc were applied.
4) Next, we made the tfidf-w2vec vectorizer on a data pair of question sets, and then we merged each tfidf-w2vec vector with our advanced vectors.
5) Application of a machine learning algorithm such as logistic regression, Support Vector Machines (SVM's), etc. and found a log loss for the train and the test data set.
6) After choosing the best parameters we plotted confusion matrix, precision matrix and recall matrix for each one.
7) We did the same process for TF-IDF vectorizer at the end of this project.

3.2 Block diagram:

The proposed model consists of several steps to be followed and several algorithms used. These steps are represented in the form of a block diagram or flow chart shown in Figure 1.

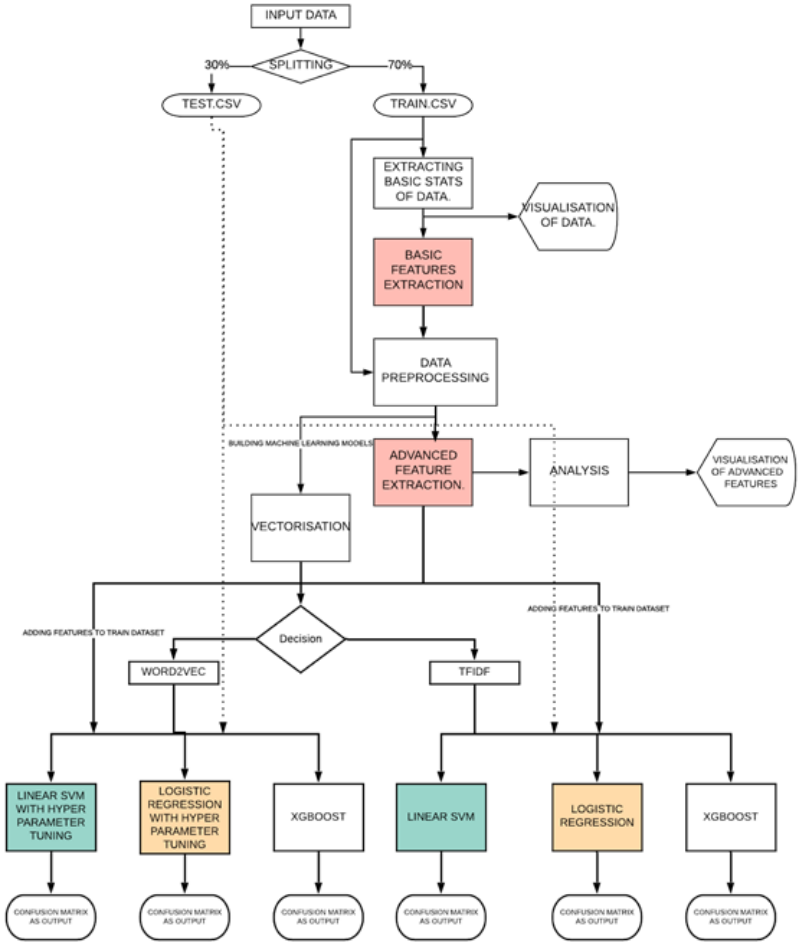


Figure 1: Blocked diagram showing the flow of the model and different models used

ISSN: 2233-7857 IJFGCN

3.2.1 Data:

3.2.1.1 Exploration:

The dataset of Quora Question Pairs, which is provided as part of a Kaggle contest

is having a training set of 404,290 question pairs and a test set of 2,345,795 question pairs.

Since the supplied test set does not contain a label for any pair of questions, the only measure of performance that can be obtained with this test set is accuracy (via an online submission to Kaggle). Thus, our data mining only took into account this training set of 404,290 pairs of questions. Each sampling point has the following fields:

• id: unique ID of each pair

• qid1: ID of first question

• qid2: ID of second question

• question1: text of first question

• question2: text of second question

• is duplicate: are duplicate questions of each other (0 indicates no duplicate, 1 indicates duplicate)

Of the 404,290 pairs of questions, 255,027 (63.08%) have a negative label (0) and 149,263 (36.92%) have a positive label (1), making our dataset unbalanced.

The character set in our dataset was not strictly ASCII; we found that 6,228 questions contained non-ASCII characters, and these questions arose in 8,744 question pairs. There were also two pairs that contained an empty string for one of their questions.

3.2.1.2 Splitting of data:

As mentioned in the subsection (3.2.1), although the question pairs are unique, majority of the question pairs are part of the multiple pairs.

Therefore, to perform the splitting of data, we have adopted two approaches. We have two kinds of splits and for both the splits, the provided training data was split into three parts: 70% for training, 20% for the validation and 10% for the testing.

1. Blind split: dataset is split while preserving the balance (the ratio of duplicates to non-duplicates) in each new dataset. For this approach, we ignored the fact that questions in the training set may also appear in the validation and test sets.
2. Disjoint split: This split preserves the balance of the data, and it additionally ensures that the questions that appear in the training, validation and test sets are different.

Generally blind split is used over disjoint split as disjoint split is a harder task .Since the vocabulary of the test and the validation sets are different from the training set, the model needs to learn more generalizable features which proves to be an additional and fairly harder task .Moreover questions tend to be asked more than once and hence we feel that the blind split is more representative of the real world than the disjoint split.

3.2.2 Reading data and basic statistics

As we deal with big datasets, repetition of data should be taken care of.

We eliminate the duplicate questions from the data set and visualize the repeated questions through a log histogram of repeated questions. The plot of number of times a question appears against the number of questions for that many occurrences is shown in the **Figure:2**.

3.2.2.1 Null value correction:

There is a probability to find null values in the huge datasets and should be deleted or replaced with average values and we found two rows with null values as follows:

```
                                                question2  is_duplicate
105780                                                NaN             0
201841                                                NaN             0
363362  My Chinese name is Haichao Yu. What English na...             0
```

We opted to fill the null value with a empty string i.e; ‘

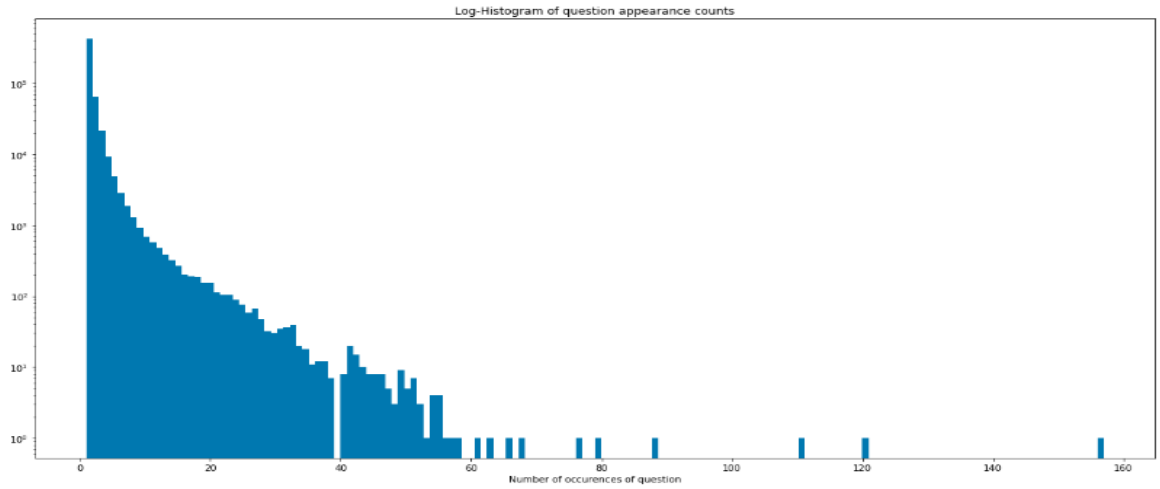


Figure 2. The plot of number of times a question appears against the number of questions for that many occurrences

3.2.3 Extraction of basic features

We construct a few features like:

- **freq_qid1** = Frequency of qid1's
- **freq_qid2** = Frequency of qid2's
- **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 frequency of qid1 and qid2
- **freq_q1-freq_q2** = absolute difference of frequency of qid1 and qid2

The Sample data set after building basic features given above is shown in the Figure 3**.**

q1len	q2len	q1_n_words	q2_n_words	word_Common	word_Total	word_share	freq_q1+q2	freq_q1-q2
35	48	7	9	4.0	16.0	0.250000	2	0
46	57	8	10	4.0	18.0	0.222222	9	3

Figure 3. Sample data set after building basic features

3.2.4 Data Preprocessing:

The data that we have is a human generated text and hence it is not uncommon to have anomalies in the form of non ASCII characters which unnecessarily increases the data size .We used different preprocessing steps to eliminate the non ASCII characters such as digits and punctuation marks .The size of the vocabulary dropped drastically from 175,999 to 94,420.

Ultimately, we tend to use the NLTK. Tokenize for tokenization as a universal preprocessing step. Additionally, for the linear models we tend to remove non-ASCII characters and for the tree-based models we tend to emotional punctuation.

Steps performed in preprocessing are:

- Removing html tags
- Removing Punctuations
- Performing stemming
- Removing Stop words
- Expanding contractions etc.

All the commands are applied only on question1 and question2 column.

3.2.5 Extraction of Advanced Features (NLP and Fuzzy Features):

We include some features to dataset to improve the machine learning model performance.

Features:

- **cwc_min** : Ratio of common_word_count to min length of word count of Q1 and Q2 cwc_min = common_word_count / (min(len(q1_words), len(q2_words))
- **cwc_max** : Ratio of common_word_count to max length of word count of Q1 and Q2 cwc_max = common_word_count / (max(len(q1_words), len(q2_words))
- **csc_min** : Ratio of common_stop_count to min length of stop count of Q1 and Q2 csc_min = common_stop_count / (min(len(q1_stops), len(q2_stops))
- **csc_max** : Ratio of common_stop_count to max length of stop count of Q1 and Q2 csc_max = common_stop_count / (max(len(q1_stops), len(q2_stops))
- **ctc_min** : Ratio of common_token_count to min length of token count of Q1 and Q2 ctc_min = common_token_count / (min(len(q1_tokens), len(q2_tokens))
- **ctc_max** : Ratio of common_token_count to max length of token count of Q1 and Q2 ctc_max = common_token_count / (max(len(q1_tokens), len(q2_tokens))
- **last_word_eq** : Check if First word of both questions is equal or not last_word_eq = int(q1_tokens[-1] == q2_tokens[-1])
- **first_word_eq** : Check if First word of both questions is equal or not first_word_eq = int(q1_tokens[0] == q2_tokens[0])
- **abs_len_diff** : Abs. length difference abs_len_diff = abs(len(q1_tokens) - len(q2_tokens))
- **mean_len** : Average Token Length of both Questions mean_len = (len(q1_tokens) + len(q2_tokens))/2
  fuzz_ratio : FuzzyWuzzy is a library of Python which is used for string matching. Fuzzy string matching is the process of finding strings that match a given pattern. Basically it uses Levenshtein Distance to calculate the differences between sequences.
- **fuzz_partial_ratio** : Fuzzy Wuzzy partial ratio raw score is a measure of the strings similarity as an int in the range [0, 100]. Given two strings X and Y, let the shorter string (X) be of length m. It finds the fuzzy wuzzy ratio similarity measure between the shorter string and every substring of length m of the longer string, and returns the maximum of those similarity measures. Fuzzy Wuzzy partial ratio sim score is a float in the range [0, 1] and is obtained by dividing the raw score by 100.

ISSN: 2233-7857 IJFGCN

- **token_sort_ratio** : Return a measure of the sequences' similarity between 0 and 100 but sorting the token before comparing.
- **token_set_ratio** : ignores duplicated words. It is similar with token sort ratio, but a little bit more flexible.
- **longest_substr_ratio** : Ratio of length longest common substring to min length of token count of Q1 and Q2 longest_substr_ratio = len(longest common substring) / (min(len(q1_tokens), len(q2_tokens))

Post application of features to the dataset the sample data looks like as shown in the Figure 4.

cwc_min	cwc_max	csc_min	...	ctc_max	last_word_eq	first_word_eq	abs_len_diff	mean_len	token_set_ratio
0.999975	0.799984	0.333322	...	0.555549	0.0	0.0	2.0	8.0	87
0.999980	0.833319	0.333322	...	0.599994	0.0	0.0	2.0	9.0	92
0.857131	0.499996	0.999986	...	0.464284	0.0	0.0	13.0	21.5	94

Figure 4. dataset after advanced feature extraction

3.2.6 Visualization

3.2.6.1 Dimensionality reduction:

We use TSNE algorithm for best results in the dimensionality reduction and visualization of data.t-Distributed Stochastic Neighbor embedding (t-SNE) is an unsupervised, on-linear technique primarily used for data exploration and visualizing high-dimensional data. The implementation of TSNE algorithm along with its parameters and values is shown in the Table 1 and the dimensionality reduction is shown in the Figure 5.

parameter	value
n_components	2
init	'random', *# pca*
random_state	101
method	'barnes_hut'
n_iter	1000
verbose	2
angle	0.5

Table 1: implemented TSNE algorithm with following set of parameter and values

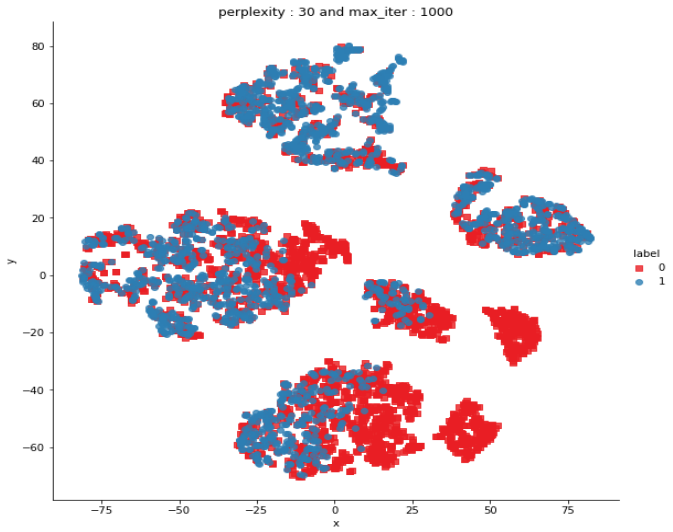


Figure 5. Visualization of dimensionality reduction using TSNE.

3.2.7 Vectorisation:

For other procedures in machine learning models, we convert qid1 and qid2 into vectors so that we can apply classifier regression models to them ,We use two types of vectorization methods, first we derive the TF-IDF scores then we apply the word2vec vectorization process based on the TF-IDF score and for the other vectorization processes, we use the TF-IDF vectorizer.

3.2.7.1 Word2vec using TF-IDF scores:

TF-IDF (term frequency-frequency of reverse document) is a statistical measure that assesses the relevance of a word for a document in a collection of documents. This is done by multiplying two measurements: the number of times a word appears in a document and the reverse frequency of the word document on a set of documents.
TF-IDF for a word in a document is calculated by multiplying two different metrics:

- Term frequency of a word in a document. There are several ways to calculate this frequency, the simplest being the gross number of instances where a word appears in a document. Then there are ways to adjust the frequency, the length of a document, or the raw frequency of the most frequent word in a document.
- Reverse frequency of word documents on a set of documents. This means how common or rare a word is throughout the document. The closer it is to 0, the more common a word. This metric can be calculated by taking the total number of documents, dividing it by the number of documents containing a word and calculating the logarithm.
- So, if the word is very common and appears in many documents, this number will approach 0. Otherwise, it will approach 1. Multiplying these two numbers results in the TF-IDF score of a word in a document. The higher the score, the more relevant this word is in this particular document.

To put it in more formal mathematical terms, the TF-IDF score for the word t in the document d from the document set D is calculated as follows shown in equations: 1, 2, and 3:

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \quad \text{..... (1)}$$

Where:

$$tf(t,d) = log(1 + freq(t,d)) \quad \text{..... (2)}$$

$$idf(t,D) = log(\frac{N}{count(d \in D : t \in d)}) \quad \text{..... (3)}$$

Now we apply word2vec using TF-IDF generated scores for words in the question pairs

- After we find TF-IDF scores, we convert each question to a weighted average of word2vec vectors by these scores.
- here we use a pre-trained GLOVE model which comes free with "Spacy". It is trained on Wikipedia and therefore, it is stronger in terms of word semantics.

3.2.7.2 Word2vec approach to find:

Word2Vec is a method to construct such an embedding. It can be obtained using two methods (both involving Neural Networks): Skip Gram and Common Bag Of Words (CBOW)
Consider the following similar sentences: *Have a good day* and *Have a great day.* They hardly have different meaning. If we construct an exhaustive vocabulary (let's call it V), it would have V = {Have, a, good, great, day}.
Now, let us create a one-hot encoded vector for each of these words in V. Length of our one-hot encoded vector would be equal to the size of V (=5). We would have a vector of zeros except for the element at the index representing the corresponding word in the vocabulary. That particular element would be one. The encodings below would explain this better.
Have = [1,0,0,0,0]`; a=[0,1,0,0,0]` ; good=[0,0,1,0,0]` ; great=[0,0,0,1,0]` ; day=[0,0,0,0,1]` (` represents transpose)
If we try to visualize these encodings, we can think of a 5 dimensional space, where each word occupies one of the dimensions and has nothing to do with the rest (no projection along the other dimensions). This means 'good' and 'great' are as different as 'day' and 'have', which is not true.
Our objective is to have words with similar context occupy close spatial positions. Mathematically, the cosine of the angle between such vectors should be close to 1, i.e. angle close to 0.
Word2Vec is a method to construct such an embedding. It can be obtained using two methods (both involving Neural Networks): Skip Gram and Common Bag Of Words (CBOW)
Deeper into the actual architecture.

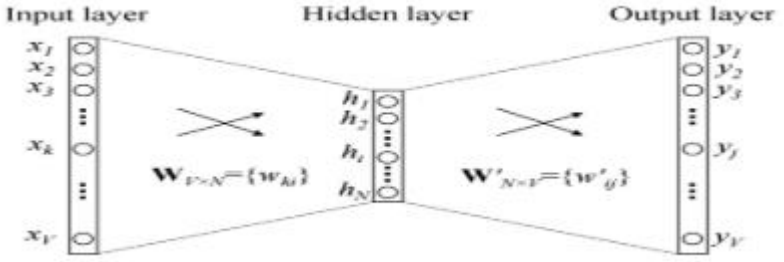


Figure 6: Vector space model for single context words

The input or the context word is a one hot encoded vector of size V. The hidden layer contains N neurons and the output is again a V length vector with the elements being the softmax values. Vector space model for single context words and multiple context words are shown in Figures 6 and 7**.**
Let's get the terms in the picture right:
- Wvn is the weight matrix that maps the input x to the hidden layer (V*N dimensional matrix)

ISSN: 2233-7857 IJFGCN

-W`nv is the weight matrix that maps the hidden layer outputs to the final output layer *(N*V dimensional matrix)*
The hidden layer neurons just copy the weighted sum of inputs to the next layer. There is no activation like sigmoid, tanh or ReLU. The only non-linearity is the softmax calculations in the output layer.
But, the above model used a single context word to predict the target. We can use multiple context words to do the same.

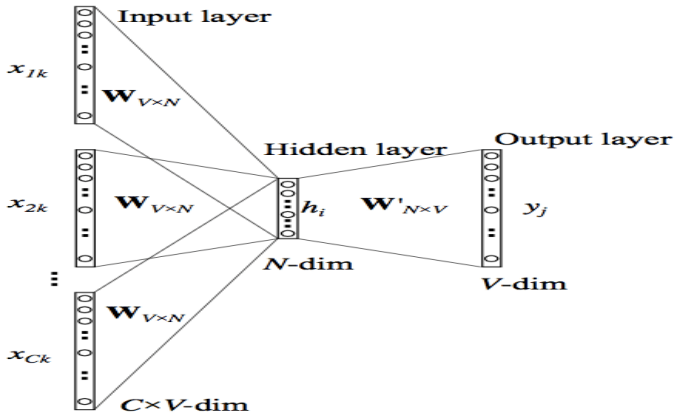


Figure 7: Vector space model for multiple context words

The above model takes C context words. When *Wvn* is used to calculate hidden layer inputs, we take an average over all these C context word inputs.
So, we have seen how word representations are generated using the context words. But there's one more way we can do the same. We can use the target word (whose representation we want to generate) to predict the context and in the process, we produce the representations.

3.2.8 Building Machine Learning models, Obtained outputs and Analysis:

We are dealing with binary classification problem so we use classifier regression models
Linear SVM and logistic regression shows best log-loss and good prediction values
Logistic regression with hyper parameter tuning:

3.2.8.1 Logistic regression:

Logistic Regression is used when the dependent variable (target) is categorical.
For example,

- To predict whether an email is spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time.
From this example, it can be inferred that linear regression is not suitable for classification problem. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to 1.
Model Output = 0 or 1
Hypothesis => t = WX + B
h(x) = sigmoid (t)
The sigmoid function behavior is shown in the figure 8.

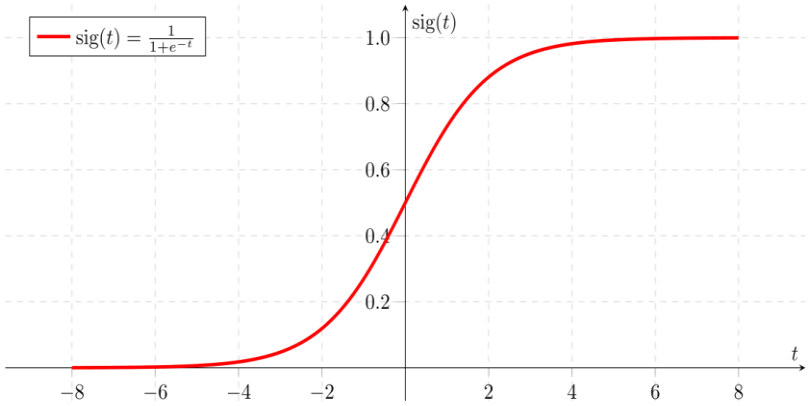


Figure 8: Sigmoid Activation Function in which 't' goes to infinity, Y(predicted) will become 1 and if 't' goes to negative infinity, Y(predicted) will become 0.

3.2.8.2 Analysis of the hypothesis:

The output from the hypothesis is the estimated probability. This is used to infer how confident can predicted value be actual value when given an input X. Consider the below example,

X = [x0 x1] = [1 IP-Address]

Based on the x1 value, let's say we obtained the estimated probability to be 0.8. This tells that there is 80% chance that an email will be spam.

Mathematically this can be written as,

This justifies the name 'logistic regression'. Data is fit into linear regression model, which then be acted upon by a logistic function predicting the target categorical dependent variable.

$h_\Theta(x) = P(Y=1 \mid X; \text{theta})$

Probability that Y=1 given X which is parameterized by 'theta'.

$P(Y=1 \mid X; \text{theta}) + P(Y=0 \mid X; \text{theta}) = 1$

$P(Y=0 \mid X; \text{theta}) = 1 - P(Y=1 \mid X; \text{theta})$

3.2.8.3 Hyper parameter tuning:

In SGD Classifier we need to search for best alpha value so we iterate alpha as follows:

alpha = [10 ** x for x in range(-5, 2)]

We check every alpha value in that range and calculate log loss

For values of best alpha = 10 The train log loss is: 0.5532114928149146

For values of best alpha = 10 The test log loss is: 0.5651686320754837

Total number of data points : 30000

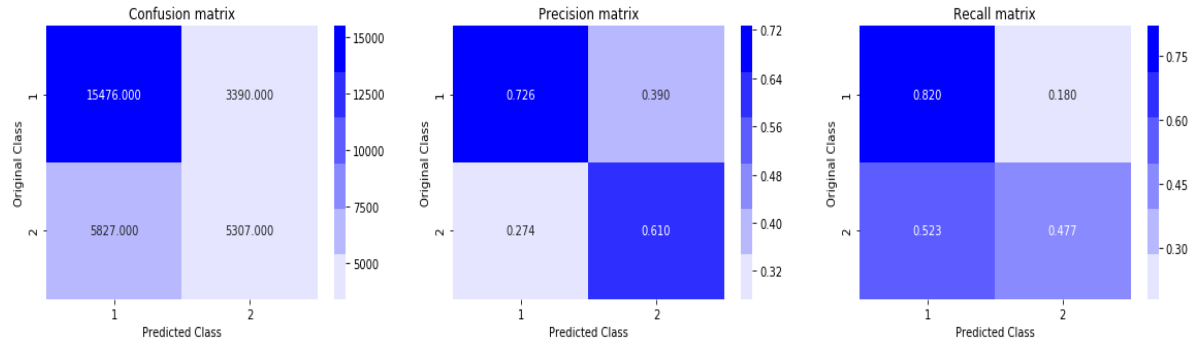


Figure 9: Output for logistic regression model with hyper parameter tuning when word2vec embedding used

3.2.8.4 Linear Support Vector Machine with Hyper parameter tuning :

SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a Hyperplane which separates the data into classes.

According to the SVM algorithm we find the points closest to the line from both the classes.

These points are called support vectors. Now, we compute the distance between the line and the support vectors. This distance is called the margin. Our goal is to maximize the margin. The Hyperplane for which the margin is maximum is the optimal Hyperplane as shown in Figure 10.

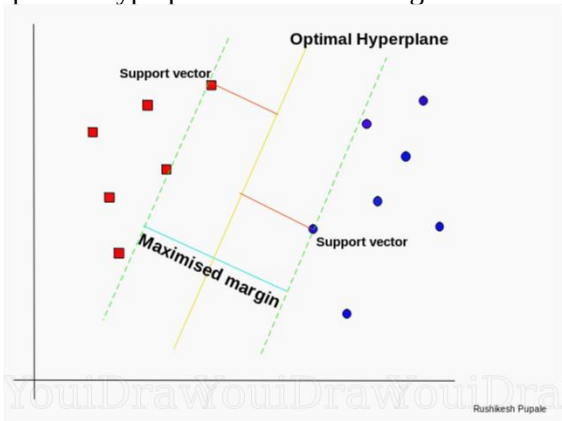


Figure 10: Optimal Hyperplane using the SVM algorithm

Thus SVM tries to make a decision boundary in such a way that the separation between the two classes(that street) is as wide as possible.

Simple, aren't it? Let's consider a bit complex dataset, which is not linearly separable.

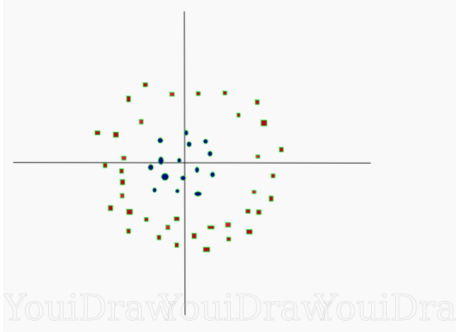


Figure 11:Non-linearly separable data

This data in Figure 11 is clearly not linearly separable. We cannot draw a straight line that can classify this data. But, this data can be converted to linearly separable data in higher dimension. Let's add one more dimension and call it z-axis. Let the coordinates on z-axis be governed by the constraint,

$z = x^2+y^2$

So, basically z coordinate is the square of distance of the point from origin. Let's plot the data on z-axis.

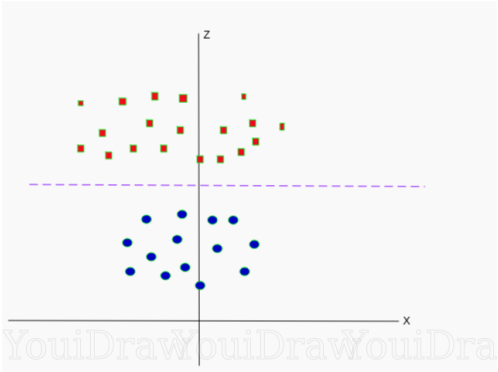


Figure 12: Dataset on higher dimension

Now the data is clearly linearly separable as in Figure 12. Let the purple line separating the data in higher dimension be z=k, where k is a constant. Since, $z=x^2+y^2$ we get $x^2 + y^2 = k$; which is an equation of a circle. So, we can project this linear separator in higher dimension back in original dimensions using this transformation.

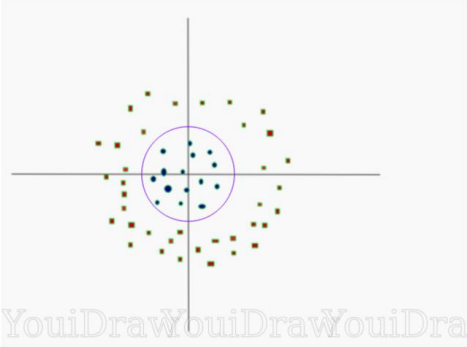


Figure 13: Decision boundary in original dimensions

Thus we can classify data by adding an extra dimension to it so that it becomes linearly separable and then projecting the decision boundary back to original dimensions using mathematical transformation. But finding the correct transformation for any given dataset isn't that easy. Thankfully, we can use kernels in sklearn's SVM implementation to do this job.

Hyperplane:

Now that we understand the SVM logic lets formally define the Hyperplane.

A Hyperplane in an n-dimensional Euclidean space is a flat, n-1 dimensional subset of that space that divides the space into two disconnected parts.

In hyper parameter tuning we get the following alpha and log loss values shown in Figure 14:

For values of best alpha = 0.001 The train log loss is: 0.5304926919931381

For values of best alpha = 0.001 The test log loss is: 0.5362286*553574601*

Total number of data points : 30000

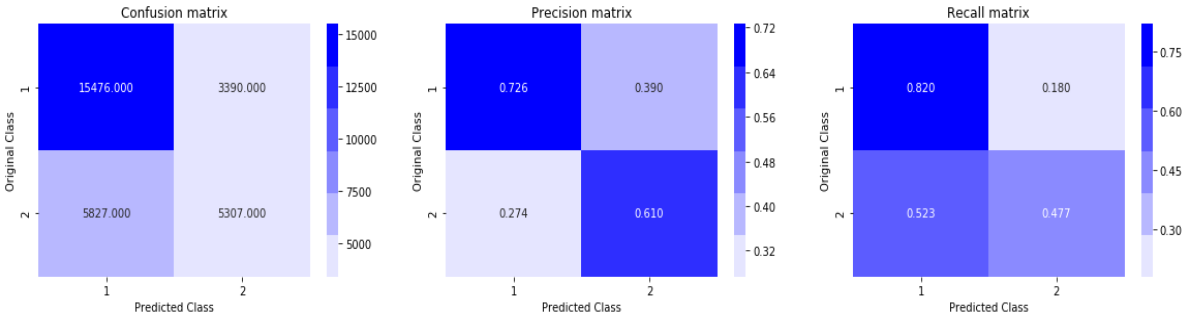


Figure 14: Output confusion matrices for linear svm with hyper parameter tuning when word2vec vectorisation is used

3.2.8.5 XgBoost model and its outputs:

The beauty of this powerful algorithm lies in its scalability, which drives fast learning through parallel and distributed computing and offers efficient memory usage.
XgBoost is an ensemble learning method. Sometimes, it may not be sufficient to rely upon the results of just one machine learning model. Ensemble learning offers a systematic solution to combine the predictive power of multiple learners. The resultant is a single model which gives the aggregated output from several models.
The models that form the ensemble, also known as base learners, could be either from the same learning algorithm or different learning algorithms. Bagging and boosting are two widely used ensemble learners.
Bagging: Consider a single training dataset that we randomly split into two parts. Now, let's use each part to train a decision tree in order to obtain two models.
When we fit both these models, they would yield different results. Decision trees are said to be associated with high variance due to this behavior. Bagging or boosting aggregation helps to reduce the variance in any learner.
Boosting: In boosting, the trees are built sequentially such that each subsequent tree aims to reduce the errors of the previous tree. Each tree learns from its predecessors and updates the residual errors. Hence, the tree that grows next in the sequence will learn from an updated version of the residuals.
The base learners in boosting are weak learners in which the bias is high, and the predictive power is just a tad better than random guessing.

Boosting consists of three simple steps:

- An initial model F0 is defined to predict the target variable y. This model will be associated with a residual (y – F0)
- A new model h1 is fit to the residuals from the previous step
- Now, F0 and h1 are combined to give F1, the boosted version of F0. The mean squared error from F1 will be lower than that from F0 eqn(4):

$$F_1(x) \leftarrow F_0(x) + h_1(x) \quad \text{..... (4)}$$

To improve the performance of F1, we could model after the residuals of F1 and create a new model F2 eqn(5):

$$F_2(x) \leftarrow F_1(x) + h_2(x) \quad \text{..... (5)}$$

This can be done for *'m'* iterations, until residuals have been minimized as much as possible eqn(6):

$$F_m(x) \leftarrow F_{m-1}(x) + h_m(x) \quad \text{..... (6)}$$

model outputs:

Train-log loss: 0.685518 valid-log loss: 0.68558
Multiple eval metrics have been passed: 'valid-log loss' will be used for early stopping.

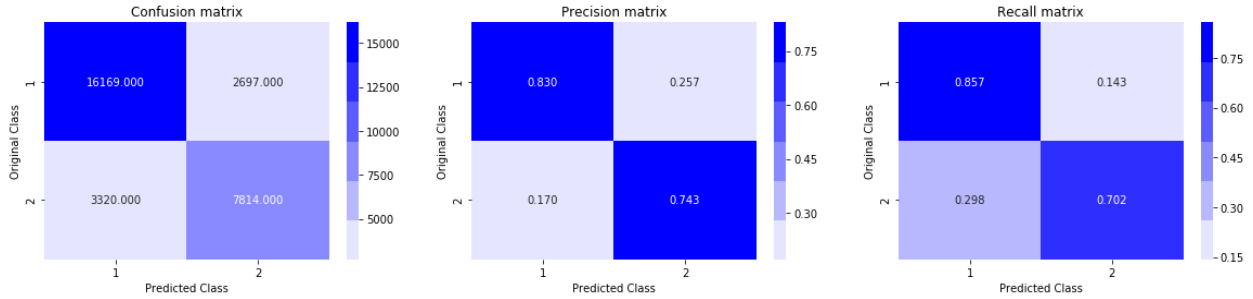


Figure 15: Output confusion matrices for Xgboost model

3.2.8.6 Logistic Regression when TF-IDF features are applied:

Input parameters:
alpha = [10 ** x for x in range(-5, 3)]
alpha=i, penalty='l2', loss='log', random_state=42
method="sigmoid"
For values of best alpha = 0.001 The train log loss is: 0.4972706175797248
For values of best alpha = 0.001 The test log loss is: 0.5001691771613037
Total number of data points : 30000

OUTPUT(shown in Figure 16**):**

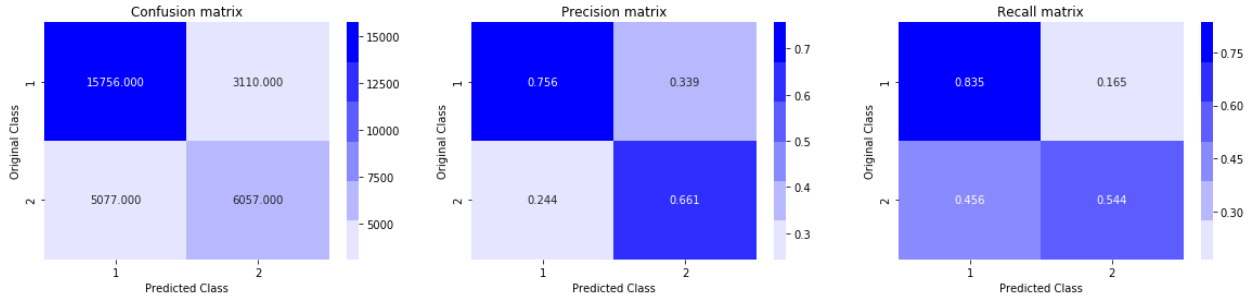


Figure 16: Output confusion matrices for logistic regression with TF-IDF

3.2.8.7 Linear SVM with TF-IDF:

Input parameters:
alpha = [10 ** x for x in range(-5, 4)]
alpha=i, penalty='l1', loss='hinge', random_state=42
method="sigmoid"
For values of best alpha = 1e-05 The train log loss is: 0.5019785812917407
For values of best alpha = 1e-05 The test log loss is: 0.505235521953195
Total number of data points : 30000

OUTPUT: (shown in Figure 17)

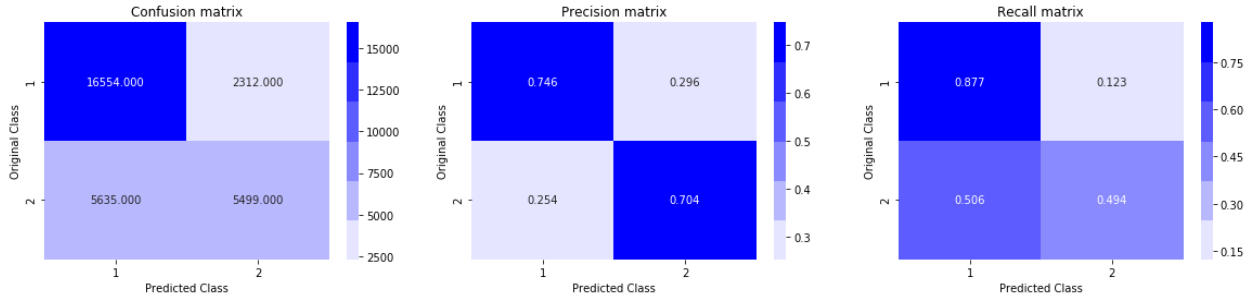


Figure 17: Output confusion matrices for Linear SVM with TF-IDF

3.2.8.8 Xgboost with TF-IDF:

Input parameters:

n_estimators = [50,100,150,200,300,400,500]
learning_rate=0.1, n_estimators=i,n_jobs=-1

OUTPUT: (shown in Figure 18)
learning_rate=0.1, n_estimators=500, n_jobs=-1
The test log loss is: 0.36562741985161873

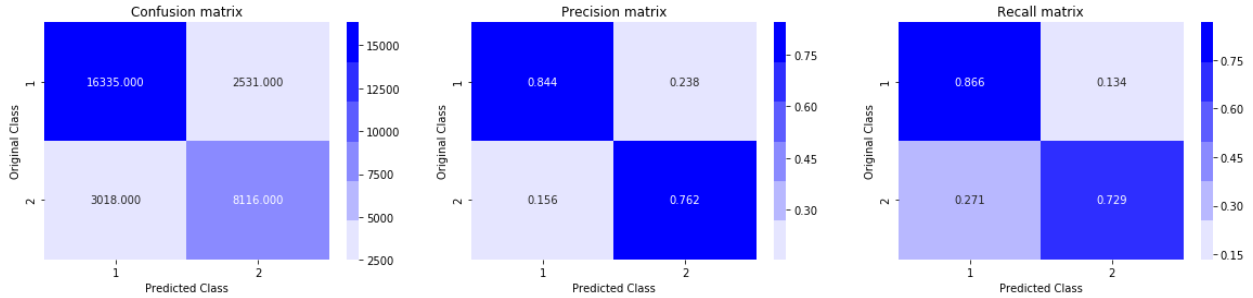


Figure 18: Output confusion matrices for Xgboost with TF-IDF

3.2.8.9 Accuracy analysis for outputs:

Log loss values of the different machine learning models used are analyzed as shown in the Table 2 and the best model is chosen accordingly.

Model	vectorizer	log loss
Logistic regression	TF-IDF w2vec	0.5651
Linear SVM	TF-IDF w2vec	0.5362
Xgboost	TF-IDF w2vec	0.3896
Logistic regression	TF-IDF	0.5001
Linear SVM	TF-IDF	0.5025
Xgboost	TF-IDF	0.3396

Table 2: Accuracy of outputs for different machine learning models used

4. Conclusion:

We tested a large number of machine learning models with the dataset at hand i.e the quora question pairs. XgBoost models, Support Vector Machine (SVM) models, Logistic regression models are used with TF-IDF and Word2vec algorithms for question pair identification, found the log loss and analyzed each model using its confusion matrices. We found out that TF-IDF vectorizer with Xgboost proves to be a reliable model to depend on with log-loss of 33.96%.Also the quora question pairs dataset proves to be an important resource for the further exploration of the field of Natural language understanding. This Quora question pair model can be further developed with lesser log losses and used for a better understanding and application of Natural Language Processing.

REFERENCES:

1.Shashank Pathak, Ayush Sharma, Shashank Shekhar Shukla.Semantic String Similarity for Quora Question Pairs. International Journal of Advances in Science, Engineering and Technology ISSN : 2321 –8991. ISSN : 2321 – 9009.Volume-7,Issue-4 Oct, 2019 ,Page(s): 77-80.

2.Dr. P.V. Rama Raju, P.M.P Gayatri, G. Nagaraju.Detection, Classification of Pest and Disease from Crop Images Using Neural Networks.International Journal Of Advanced Science and Technology(IJAST) ISSN:2005-4328E-ISSN: 2207-6360.Volume-127, june-2019(342-345).

3.T. Mikolov, M. Karafi´at, L. Burget, J. Cernock`y, and S. Khudanpur, Recurrent neural network based language model. International Journal of Advances in Science Engineering and Technology, ISSN(p): 2321 –8991, ISSN(e): 2321 – 9009 .Volume-6, Issue-4, Oct.-2018,

4.Erik Cambria,Bebo White.Jumping NLP Curves: A Review of Natural Language Processing Research. IEEE Computational Intelligence Magazine Volume-9 , Issue-2 , May 2014 ,Page(s): 48 - 57.

5.Dr. P. V. Rama Raju, G. Naga Raju, T. Neelima,A.Aparna. Analysis of Stock by Using Salesforce Methodology.International Journal Of Research ISSN:2236-6124.Volume-8, Issue -4 April-2019(127-132).

6.Asma Ben Abacha and Dina Demner-Fushman.A question-entailment approach to question answering.The National Center for Biotechnology Information (NCBI) Volume-20,Oct-2019().

7.Shiyao Xu, Shijia E, and Yang Xiang.An Ensemble Model Based on Siamese Neural Networks for the Question Pairs Matching Task.CEUR Workshop proceedings Volume-2242/paper10 Page(s):1-6.

8.Dr. P. V. Rama Raju, G. Naga Raju,R.Devi Priya,S.Krishna Sri,T.S.S.Prasad.Speech stratification using KNN method.International Journal of Innovative Engineering and Management Research ISSN:2456-5083.Volume-08, Issue-03 March-2019,Pages: 133–136.

9.Thales A.P. Westa,∗ , Juan J. Mongea , Les J. Dowlinga , Steve J. Wakelina , Richard T. Yaoa ,Andrew G. Dunninghama , Tim Payn.Comparison of spatial modelling frameworks for the identification of future afforestation in New Zealand. Elseiver. 0169-2046. Issue- 15 February 2020,Page(s): 1 - 5

10.T. Sridevi, P. Mallikarjuna Rao, P V Ramaraju and G. Nagaraju.Three Tier security for the Financial Services & e-Commerce Applications.International Journal of Control Theory and Applications ISSN:0974-5572.Volume-10, Issue-26 2017,Pages: 269–277.

11.Qaiser, Shahzad,Ali, Ramsha. Text Mining by the use of TF-IDF to Examine the Relevance of Words to Documents.International Journal of Computer Applications ISSN :0975 - 8887. Volume-181 July-2018,pages:1-5.

12.G. Naga Raju, Dr.P V Rama Raju, Ch.Ramya Priya, G.Padmini Devi,I.Ajay Kanth, A.H.V.Kiran Kumar.International Journal of Innovative Engineering and Management Research ISSN:2456-5083.Volume-07, Issue-05 April-2018,Pages: 1-8.

13.A. Brenning. Spatial prediction models for landslide hazards: review, comparison and evaluation. Natural Hazards and Earth System Sciences.,SRef-ID: 1684-9981/nhess/2005-5-853, Issue: 2005, pages: 853–862

14.Peng Wan, Dihua Sun, Min Zhao, Li Wan, Shuang Jin.Multistability and attraction basins of discrete-time neural networks with non monotonic piecewise linear activation functions.Science direct Volume 122 February 2020,Pages 231-238.

15.P V Ramaraju, G. Nagaraju, V.N.Ganeswarreddy, V.Sai Kanna, B.Sashank.Traffic Controller Based on Density with RF Remote Override.International Journal of Innovative Technology and Exploring Engineering(IJITEE) ISSN:2278-3075.Volume-8, Issue-6S4 April 2019,Pages: 674–678.

16.Keith Wurtz,A Methodology for Generating Placement Rules that Utilizes Logistic Regression, Journal of Applied Research in the Community College,Vol. 15, No. 2, Spring 2008 Pages: 59-61.

17.In Lee,Yong Jae Shin.Machine learning for enterprises: Applications, algorithm selection, and challenges.ELSEVIER/Business Horizons Volume 63, Issue 2, March–April 2020, Pages 157-170.

ISSN: 2233-7857 IJFGCN

18.Iain Rice.Improved data visualisation through multiple dissimilarity modelling.ELSEVIER/Information Sciences,Volumes 370–371, 20 November 2016, Pages 288-302.

ABOUT AUTHORS:

Dr. P.V. RAMA RAJU

Presently working as a Professor of Department of Electronics and Communication Engineering, S.R.K.R.Engineering College, A.P, India. His research interests include Biomedical Signal Processing, Signal Processing, Image Processing, VLSI Design, Antennas and Microwave Anechoic Chambers Design. He is author of several research studies published in national and international journals and conference proceedings.

G. NAGA RAJU

Presently working as assistant professor in Dept.of ECE, S.R.K.R. Engineering College, Bhimavaram, A.P, India. He received B.Tech degree from S.R.K.R.Engineering College, Bhimavaram in 2002 and M.Tech degree in computer electronics specialization from Govt. College of Engg., Pune University in 2004. His current research interests include Image Processing, digital security systems, Signal processing, Biomedical Signal Processing and VLSI Design.

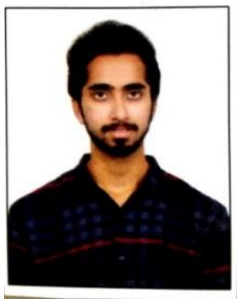
N. NIKHIL

Presently pursuing Bachelor of Technology degree in Electronics and Communication Engineering at S.R.K.R.Engineering College, Bhimavaram, AP, India.

M.HEMANTH GUPTA

Presently pursuing Bachelor of Technology degree in Electronics and Communication Engineering at S.R.K.R.Engineering College, Bhimavaram, AP, India.

ISSN: 2233-7857 IJFGCN



CHANDAN AKELLA

Presently pursuing Bachelor of Technology degree in Electronics and Communication Engineering at S.R.K.R.Engineering College, Bhimavaram, AP, India.

S.KUMARA SIDDARTH

Presently pursuing Bachelor of TEchnology degree in Electronics and Communication Engineering at S.R.K.R. Engineering College, Bhimavaram, AP,India.