1. Introduction

Yelp is the internet's largest social-local platform and has got a lot of attention from various businesses around the world. To ease the process of finding great businesses, the yelp was created in 2004. The dataset has 2, 00,000 pictures, 66, 85,900 reviews, 12, 23,094 tips which are written for 1, 92,605 businesses. For each of the businesses, there are 1.2 million business attributes. The yelp has 135 million monthly users on average and 37 million unique mobile app visitors. Voice of the customer is inarguably an effective indicator of the performances of the businesses.

2. Problem Statement

Classify a given yelp review text into relevant categories.

Yelp users give ratings and write reviews about businesses and services on Yelp. These reviews and ratings help other Yelp users and vendors to evaluate a business or a service and make a choice. Considering the feedback written by the customer on their Shopping experience in Las Vegas we are classifying them into categories.

3. Methodology

About dataset:

There are a total of 6 JSON files that are made public for the Yelp dataset challenge. [1] The JSON files are business.JSON, checkin.JSON, photo.JSON, review.JSON, tip.JSON, and user.JSON. The challenge dataset contains 1,223,094 tips by 1,637,138 users. There are over 1.2 million business attributes like hours, parking, availability, and ambiance. It contains aggregated check-ins over time for each of the 192,609 businesses.

The size of the business.JSON file is 1, 35,039 KB. The business file contains 1, 92,609 tuples and 14 attributes. The attributes in business.JSON files are address, attributes, business_id, categories, city, is_open, latitude, longitude, name, postal_code, review_count, stars, state, hours. Business.JSON file contains the attribute location. It gives detailed information about the location of the business. It consists of address, postal code, city, state. Majority Business reviewed in this dataset is located in the U.K.: Edinburgh, Germany: Karlsruhe, Canada: Montreal and Waterloo, U.S.: Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, and Las Vegas. Each business has a unique business_id. Each business is given a star rating. There are 28,216 businesses with 5-star ranting, 35969 businesses with 4-star rating, 25,996 businesses with 3-star rating, 11,426 businesses with a 2-star rating and 4874 businesses with a 1-star rating. The file contains attributes that provide information about whether the business takes reservations, is the business good for kids, noise level, categories of business, availability of parking facility, restaurant's price range, ambiance. The businesses described in the Yelp dataset belong to

different categories, such as restaurants, shopping, hotels, and travel, etc. The hours' attribute gives information about working hours of business and also the day of working. Hours are using 24 hr clock. The business file also tells whether the business is open or closed, representing in the form of integer values i.e., 0 for closed and 1 for open.

The size of the checkin.JSON file is 3, 99,227 KB. It contains check-ins on business. For a particular business id, the date and time of check-ins are provided.

The review.JSON file consists of 9 attributes. The attributes are business_id, user_id, stars (integer values between 1-5), text, and date, useful, funny, and cool. It consists of reviews given by the user on business. business id of those businesses on which the review is written. The user id of the users who have written or given the reviews. The date on which the review is written. Useful, cool, funny attributes indicate the number of useful, cool and funny votes received on the given review.

The size of the photo.JSON is 25,060 KB. The attributes are photo_id, business_id, caption, label. The photo_id is unique strings with 22 characters. Photos are given caption and label indicating the category to which the photo belongs. The different categories are food, drink, menu, inside or outside.

The size of the tip.JSON file is 233MB. The Tip file consists of 1223094 tuples and 5 attributes. The attributes are Text, date, compliment_count, business_id, user_id. Tips are written by a user on a business. Tips are shorter than reviews and they imply quick suggestions. Tips are based on services, products, place, price, etc. The file consists of date attribute which conveys when a tip was written by a user in YYYY-MM-DD format. The user file consists of 24 attributes. The attributes are user_id ,review_count, yelping_since,votes,elite,average_stars , friends and compliments . User data includes the user’s friend mapping and the metadata that is associated with the user.

Exploratory data analysis

Business.JSON and tip.JSON are the two relevant files for our project. Business.JSON file contains information about business names and categories whereas tip.JSON file contains information about the customer feedback. Business.JSON file is read into a data frame. We checked for the missing values in the attributes of the business.JSON file and got to know that the 'categories' attribute has 0.250248% missing values, hours attribute has 23.27% and attributes have 14.97% missing value. tip.JSON file is read into a separate data frame and the dataset found to be very clean with no missing values.

We plotted the bar graph to realize the top categories in the business dataset. By the bar graph, the inference was drawn that restaurant is the top 1 category, followed by shopping. We also checked out the cities where the majority of the businesses are running. The majority of businesses are located in Las Vegas. Unique businesses are running in Las

Vegas. So, our immediate decision was to choose the business running in Las Vegas. The graph was plotted to realize the top business categories in Las Vegas. From the graph, we inferred that most of the business running in Las Vegas belonged to the shopping category. We checked the number of businesses that are currently active or running in Las Vegas. Among 5355 shopping businesses, 4349 businesses are open and 1006 businesses are closed. Continuing our analysis further, we found out that Venetian Las Vegas is the maximum reviewed business with review count 3499 and around 686 businesses have minimum reviews. Moreover, there are 4055 unique shopping categories in Las Vegas.

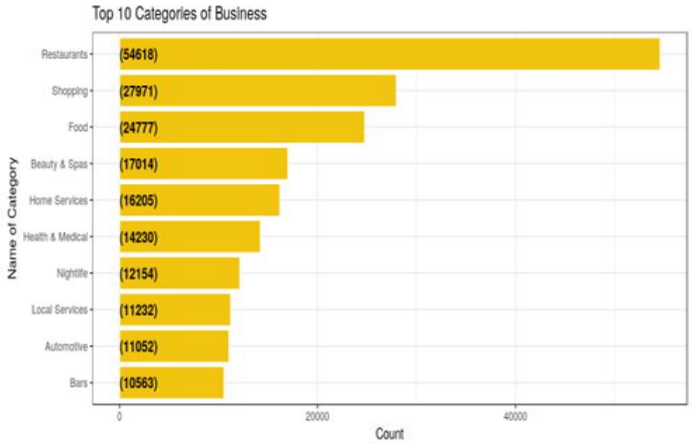

Figure 1 Top 10 categories of business

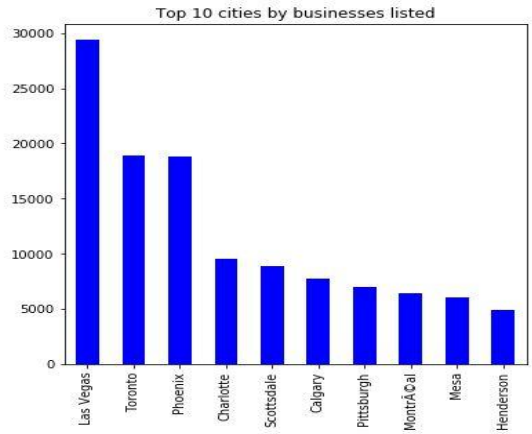

Figure 2 Top 10 cities by businesses

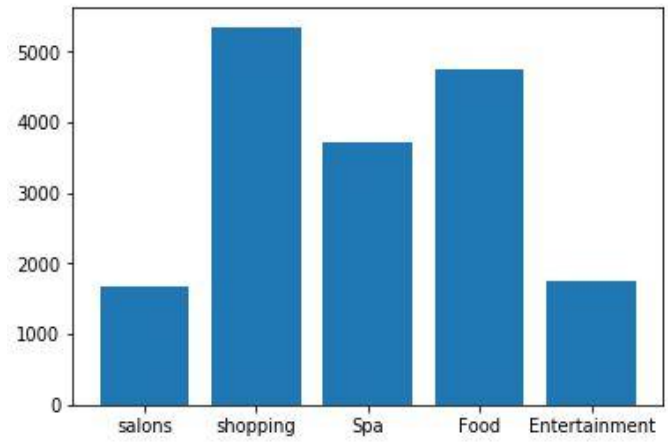

Figure 3 Top business categories in Las Vegas

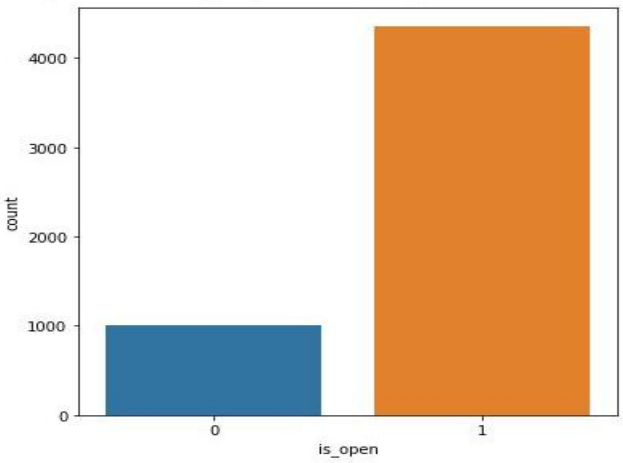

Figure 4 Among 5355 shopping businesses, 4355 businesses are open and 1006 businesses are closed

Data Pre-processing

The business dataset has 14 columns, out of which 6 columns (‘hours', ‘state', ‘latitude', ‘longitude', ‘postal_code', ‘stars') are dropped as they are not much use for the project objective. Business and tip datasets are merged concerning business_id of open businesses.

The text review tips need to be pre-processed as Yelp allows them to write text reviews in free form. The NLTK module helps with splitting up words, splitting sentences from paragraphs, highlighting the main subjects, recognizing the part of speech of those words and much more that part of Natural Language Processing (NLP) methodology.[5]

The first step would be likely doing a simple split on text reviews by regular expression to split by '.!?' Followed by tokenization and converting the texts to lowercases. One of the major forms of pre-processing is going to be filled out useless data. In natural language processing, useless words, are referred to as stop words that carry no meaning. It is necessary to pre-process the reviews to extract meaningful content from each of them. We would not want these words taking up space and processing time. NLTK helps with a bunch of words that they consider to stop words, which can access it via the NLTK corpus. It is followed by stemming which is a sort of normalizing method. Many variations of words carry the same meaning when tense is not involved. We use the stem to normalize sentences and shorten the lookup. Porter stemmer is one such algorithm that is most popular, which has been around since 1979.

Frequently occurring words that fall under the considered classes are selected from the word cloud. The presence of each class label containing the obtained words in the tip review text is checked and the outcome is stored in a new label column respectively.

Figure 5 Categories in shopping

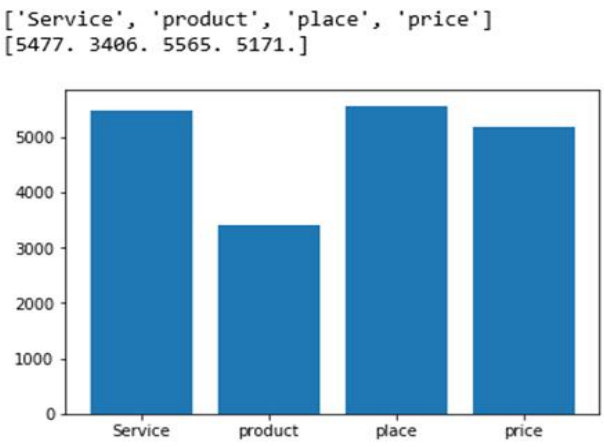

Figure 6 Labels identified in the review

4. Experimentation

Text classification is the task of allotting predefined categories to a free-text. The review written by Yelpers is classified into 4 different categories. The categories are the product, price, service, and place. It helps the new Yelpers to extract insight from already written tips and make the right decisions.

[2]The classifier takes text as input, it analyses the text contents and then automatically assigns the relevant categories that represent the text. Consider the text "Great Service, Great People". The text classifiers take the above text as input and since the text is written is on the service provided by the business it classifies it as text belonging to the service category. There are two ways to classify text. One approach is to classify text manually by interpreting the context of the text and classifying if accordingly. It provides quality result but it is time-consuming and expensive. The second approach applies machine learning, natural language processing. It classifies the text faster and in an effective way. The rule-based approach helps us to classify the text into relevant categories by using a set of handmade acceptable rules. These rules discipline the model to use semantically relevant elements of a text to classify the text into relevant categories by using a set of handmade acceptable rules. These rules discipline the model to use semantically relevant elements of a text to classify the text into appropriate categories. We want our model to classify the text into appropriate categories. We want our model to classify the text into four groups, namely, product, price, place, service. We have defined the list of words that represents each group (e.g. words related to the product are the item, brand, variety, samples, stuff, and things). When we want to classify the text, the model will count the number of product-related terms that appear in the text and do the same for the rest of the categories. If the number of product-related terms is greater than the rest, then the text is classified as a product and vice versa. Rule-based systems are humanly comprehensible. The disadvantage is that these systems are time-consuming and handcrafting the rules is a tedious process for the complex datasets and also datasets where domain knowledge is poor. Rule-based systems are difficult to maintain. Another approach i.e., Machine Learning Based Systems which classifies the text by learning different associations between the words in the text to that of a particular output. Feature extraction is the first step used to train the classifier. It is a process of dimensionality reduction in which the text is reduced to numerical representation in the form of vector. The machine learning algorithm I fed with the training the data that consists of vectors for each text example along with the category to which the text belongs to. The first machine learning algorithm that we used to build our text classifier is Naïve Bayes. We could train and test our algorithm on the same dataset but it leads to some serious bias issue, so we have split our dataset 70:30. 70 percent of training data and 30% testing data. Around 13264 texts belonging to all four different categories constitute our training data. And 5685 texts belong to testing data. It is a supervised machine learning algorithm because we are

training the model with the text and along with that, we are telling it to which category that text belongs. While testing the model we show the machine with some new data and ask it to predict the relevant category based on what we taught the model before. Naïve Bayes [3] is based on Bayes theorem, which helps us to compute the probability of an event occurring given the probability of another event that has already occurred. It means that the vector representing the text will have to contain the information on the probability of appearance of words of the text within the texts of a given category so that the algorithm can compute the likelihood of that text's belonging to the category.

The second model, we applied to our dataset is the support vector machine [4]. It is a linear model that can be applied for classification problems. This model is also applicable to the regression model. The algorithm creates a line or hyperplane that separates data into classes. Since there are four different categories in our dataset SVM finds a separation line between data of four classes.SVM algorithm takes data as input and it outputs a line that separates those classes. According to the SVM algorithm, it finds the points that are closest to the hyperplane from all four classes. These points are referred to as support vectors. Then the distance between the line and support vectors is calculated and the calculated distance is called margin. The hyperplane for which the calculated distance is maximum is the optimal hyperplane.SVM model represents data as a point in space and is mapped so that the data belonging to separate categories are divided by a clear gap in such a way that the distance between hyperplane and data point is as far as possible.

The third model applied to the dataset is Random forests are an ensemble learning method for classification, regression, and other tasks. It operates by constructing a large no. of decision
trees during training time. It creates a set of decision trees from a randomly selected subset of the training set. It outputs the class that is the mode of the classes for classification or mean prediction for regression of the individual trees. It sums the votes from all decision trees to decide the final class of the test object. It runs efficiently on large databases. In classification, it estimates of what variables are important. It handles thousands of input variables without removing variables.
Random forests do not overfit. You can run as many trees as you want. The algorithm is fast. About one-third of the cases are left out of the sample, when the training set for the current tree is drawn by sampling with replacement, Out of the sample, about one-third of cases are left when training set for the current tree is drawn by sampling. This out-of-bag data is used to get a running unbiased estimate of the classification error.

5. Result and analysis

Once the model is built, it is very important to check how good the model is. So, evaluating the model is the most important task. A table which is used to explain the performance of a classification model that is applied to a test dataset whose true values are known. This table is known as the confusion matrix. The observation that is correctly predicted belongs to true positive and true negative. Whereas the values which occur when actual class disputes with the predicted class belong to false positives and false negatives. Accuracy is the ratio of correctly predicted observation to the total observations and it is the most innate performance measure Accuracy = TP+TN/TP+FN+FP+TN. The weighted average of Precision and Recall is termed as an F1 score. Therefore, this measure considers both false positives and false negatives. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if we have an uneven class distribution. Accuracy works best if false positives and false negatives have a similar cost. It is preferable to look at precision and recall when the cost of false positives and false negatives are very different.

For Naïve Bayes classifier:

Classification Report:

	precision	recall	f1-score	support
Place	0.84	0.94	0.89	1881
Price	0.74	0.95	0.83	1673
Service	0.93	0.83	0.87	1299
Product	1.00	0.32	0.48	832
accuracy			0.83	5685
macro avg	0.88	0.76	0.77	5685
weighted avg	0.85	0.83	0.81	5685

The testing accuracy obtained is 82.78%.

SVM model:

Classification Report:

	precision	recall	f1-score	support
Place	0.97	0.98	0.97	1881
Price	1.00	0.95	0.97	1673
Service	0.96	0.99	0.97	1299
Product	0.95	0.97	0.96	832
accuracy			0.97	5685
macro avg	0.97	0.97	0.97	5685
weighted avg	0.97	0.97	0.97	5685

The testing Accuracy obtained is 97.19%.

Random Forest:

Classification Report:

	precision	recall	f1-score	support
Place	0.93	0.93	0.93	674
Price	0.91	0.71	0.80	432
Service	0.75	0.92	0.82	816
Product	0.95	0.86	0.91	921
accuracy			0.87	2843
macro avg	0.89	0.86	0.87	2843
weighted avg	0.88	0.87	0.87	2843

The accuracy obtained for testing data is 87.16%.

The accuracy obtained for training data is 92.034%

By looking at the accuracy we can conclude that SVM is the best model with an accuracy of 97.19%.

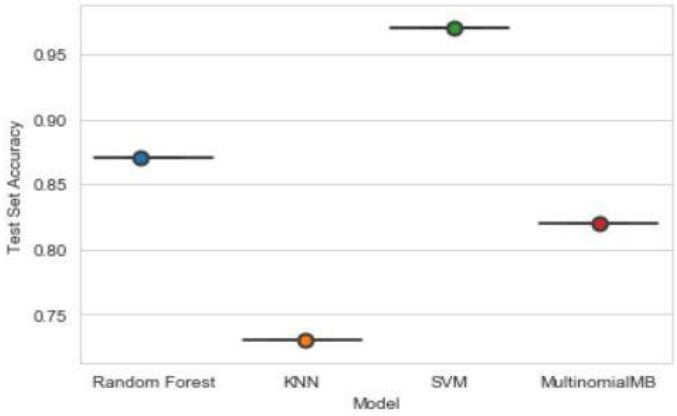

Figure 7 Model selection

Topic modeling

There are several examples of large texts like customer reviews of hotels, feeds from social media, news stories, etc. The basic application of natural language processing is to automate the extraction of the topics that are discussed in these large volumes of text. This knowledge is highly beneficial to administrators, businesses, political campaigns. This avoids the difficult manual process of reading through such large volumes. We are going to extract the volume and contribution of each topic in percentage. This gives a better idea of how important a topic is. For this to happen, we will be using Latent Dirichlet Allocation(LDA) that is imported from the Gensim package altogether with Mallet's implementation. This gives good topic segregation and runs faster.

LDA considers the document as mixtures of topics and each topic as a collection of words, both in certain probabilities. It creates proper topic-keywords distribution when we give

several topics to the algorithm. The rearrangement of keywords distribution inside the topics and topics distribution inside the documents occurs to complement the process.
The dominant keywords which are typical representations constitute the topic. We can identify what the topic is all about by the look of the keywords.

Bigram and trigram models
When two words frequently occur together in the document they are termed as bigrams.
Similarly, three words occurring together as trigrams. We can build bigrams, trigrams, quadgrams and further and also implement using Gensim's phrases. Min_count and threshold are the two important arguments to Phrases. Lower the values of these params, the easier it is to combine the words into bigrams.

Dictionary and Corpus
We create the dictionary and the corpus to give it as input to the LDA. The dictionary is a mapping between the words and their ids. Whereas the corpus is a mapping of word ids of the words and their word frequencies. To check the corresponding id of a given word, we can simply give the id as the key to the dictionary.Once the LDA model is built, lda_model, print_topics() is used to view the keywords for each topic and its importance. The weight depicts the importance of a particular keyword.

Model Perplexity and Coherence Score
The model perplexity measures how probable it is that the model that was earlier gets some new unseen data. The normalized log-likelihood of held-out data gives this measure.Topic coherence measures the degree of semantic alikeness between high scoring words in the topic. Based on this, it scores a particular topic.

After the LDA model is built, the next step is to test the generated topics and the associated keywords. A better tool to work well with Jupiter notebooks is the pyLDAvis package's interactive chart. In pyLDAvis's output, each bubble on the left-hand side plot represents a topic. The larger bubble size, the topic is said to be more prevalent. The good topic model will have large,non-overlapping bubbles scattered throughout the chart instead of being clustered in one quadrant. The models which have may topics will have overlaps with small-sized bubbles clustered in a region of a chart. The words and bars which are on the right-hand side will be updated if we move the cursor on one of the bubbles. These words are salient keywords. The next step is to improve this model by using Mallet's version of the LDA algorithm. Gensim contains a wrapper that implements Mallet's LDA. By changing just, the LDA algorithm the coherence score can increase. By building many LDA models with different values of several topics and choosing the model which gives the highest coherence value is the approach to find the optimal number of topics. Choosing the values of the number of topics that make the end of a rapid increase in topic coherence is usually a

meaningful and interpretable topic. It sometimes provides smaller sub-topics by picking even higher value. The values of the number of topics are too large when you see the same keywords are being repeated in multiple topics. The function compute_coherence_values () trains many LDA models and it gives the models and their corresponding coherence scores. It is better to pick a model that gave the highest coherence value before the plot is flattened. To find what topic a given document is about, we examine the topic number that has a maximum percentage contribution in that document.The function format_topics_sentences () aggregates the information in a table. The topic keywords are not just enough to make sense of what a topic is about. So, to understand the topic we can find the documents that the given topic has contributed much and examine the topic by reading it. In the tabular output, the perc_contribution column describes the percentage contribution of the topic in the document. The tabular output has 20 rows, one row representing one topic. It consists of a topic number, keywords, and many representative document columns. The table topic volume distribution contains information about the volume and distribution of topics to judge how widely it was discussed.

Models applied after topic modelling

Multinomial Naive bayes:

Classification report

	precision	recall	f1-score	support
0	0.73	0.85	0.79	1048
1	0.78	0.83	0.80	986
2	0.81	0.71	0.76	883
3	0.83	0.69	0.75	772
accuracy			0.78	3689
macro avg	0.79	0.77	0.78	3689
weighted avg	0.78	0.78	0.78	3689

Figure 8 Result of Multinomial Naive Bayes

Random forest:

Classification report

	precision	recall	f1-score	support
0	0.76	0.60	0.67	1105
1	0.75	0.51	0.61	883
2	0.43	0.81	0.57	849
3	0.69	0.50	0.58	785
accuracy			0.60	3622
macro avg	0.66	0.60	0.60	3622
weighted avg	0.66	0.60	0.61	3622

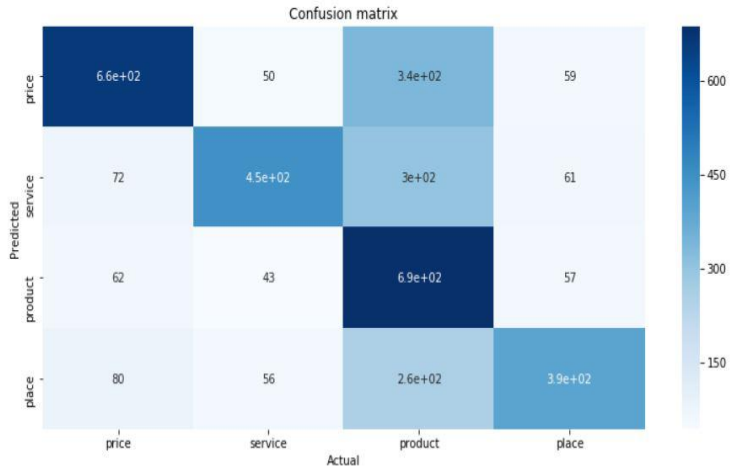

Figure 9 Result of Random forest, confusion matrix

KNN:

Classification report

	precision	recall	f1-score	support
0	0.55	0.57	0.56	1105
1	0.52	0.41	0.46	883
2	0.45	0.47	0.46	849
3	0.44	0.51	0.47	785
accuracy			0.49	3622
macro avg	0.49	0.49	0.49	3622
weighted avg	0.50	0.49	0.49	3622

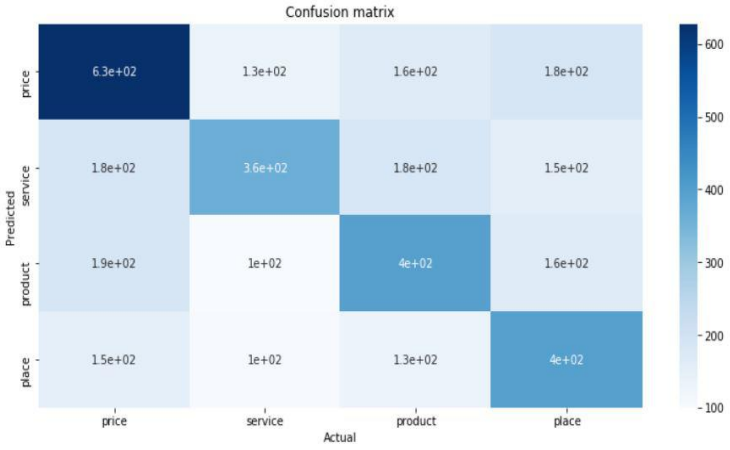

Figure 10 Result of KNN, confusion matrix

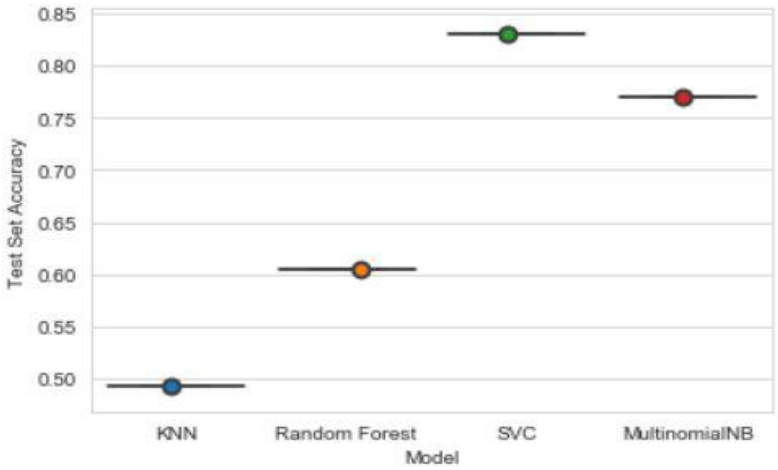

Figure 11 Model selection

By looking at the accuracy we can conclude that SVM is the best model with an accuracy of 84.38%.

Sentimental Analysis

Sentimental analysis is a process of computationally categorizing and identifying opinions from a piece of text, and determine whether the writer's attitude towards a particular topic or the product is positive, negative or neutral. Sentimental analysis is done to know whether a particular topic is good or not. The first step of sentimental analysis is tokenization. Tokenization is nothing but dividing a paragraph into a different set of statements or dividing a statement into a different set of words. So, once the process of tokenization is done, the next step is cleaning the data. Cleaning the data means to remove all the special characters i.e. punctuation. The next step is removing the stop words, which do not add any value to the analysis. The next step is classification, to classify whether a word is positive or negative or

neutral. For positive words, we give sentiment score as +1 and for the negative word we give sentiment score as -1 and for neural we give 0. Next step is modeling, where we can model our data with bag of words or we can use lexicons which is dictionary of pre-classified set of words and once the model is trained, we can perform the test on the analysis statement , more the accuracy score better will be the classification Now, next step is to calculate the final sentiment score of the sentence, by adding the sentiment score of each word . From the final score, we classify the statement is positive or negative or neutral.

6. Conclusion

In conclusion, based on our classification report and experimentation with different models such as multinomial naive bayes, random forest, KNN, and SVM, we found that SVM works best with testing accuracy of 84.38% after performing topic modelling. When rule based method was performed models gave more accuracy but it required human analysis to determine which topic was relevant unlike topic modelling which works good even for external dataset. Sentiment is determined for each review given by user for each business in Las Vegas which makes it easier for a new user and to the business managers to know which sector is satisfactory for users and on which sector they have to improve.

7. References

[1] https://www.yelp.com/dataset/challenge

[2]https://monkeylearn.com/text-classification/

[3]https://youtu.be/rISOsUaTrO4?list=PLQVvvaa0QuDf2JswnfiGkliBInZnIC4HL

[4]https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989

[5] https://pythonprogramming.net

[6]https://towardsdatascience.com/text-classification-in-python-dd95d264c802