**Predicting future Business Attention using Feature Extraction, Clustering and Sentiment Analysis.**

**Kriti Shaw**

***Abstract—Yelp started off in 2005 as a way for users to be able to provide rating and also review businesses. Many forms of data like user rating, user reviews, addresses are collected about the business. It would be of great value to businesses to be able to visualize or catch a glimpse of the reviews they will be receiving in future in Yelp - how much attention will they get in future as far reviews are concerned? In this paper, we have described different methods for predicting the amount of attention a business will receive using feature selection and sentiment analysis.***

***Index Terms*— Feature Extraction, Business Future Prediction, Feature Selection, Sentiment Analysis.**

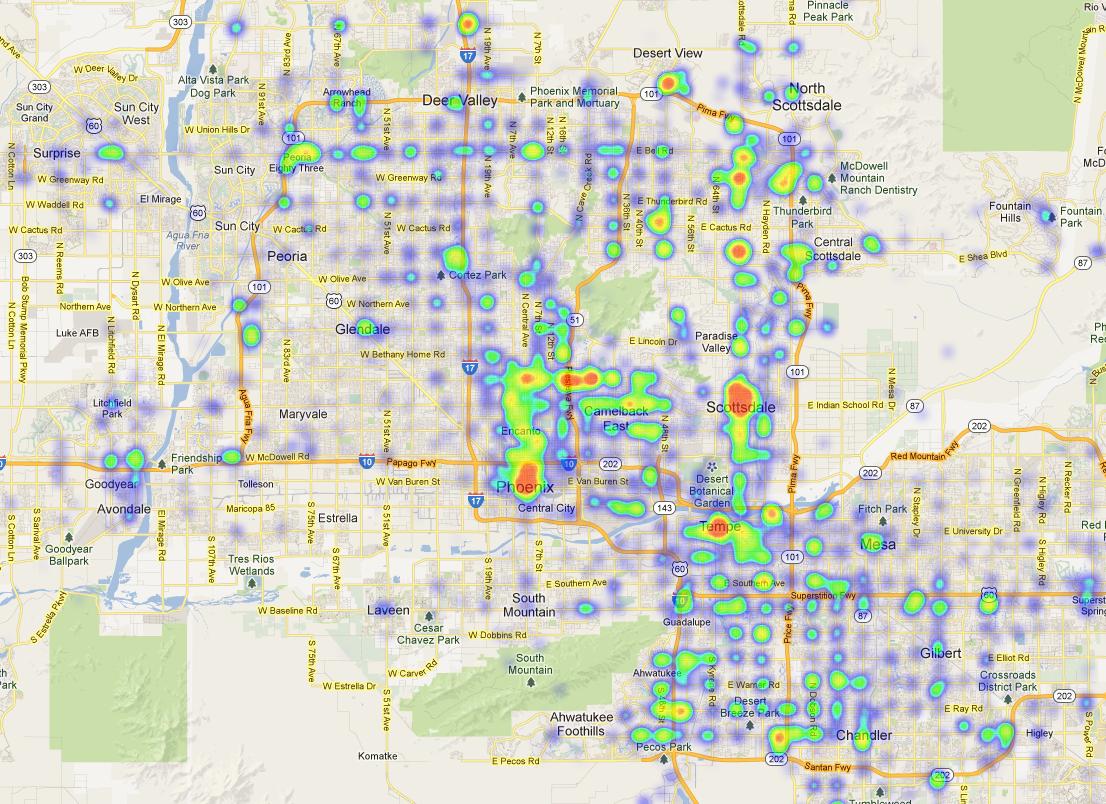
1. **INTRODUCTION**

The different types of Businesses have their listings organized while the users rate business giving them rating between 1 to 5 stars and then log reviews on Yelp. Inside this review system, there is an internal meta review system in which the users can again vote to depict whether the reviews are funny or useful. In its operation of almost 11 years, Yelp has gathered a huge amount of data for different businesses. Although the businesses can see their reviews and also view the ratings they have currently, there is nothing that gives them future predictions about their business

In this project, we will process the raw information that is available and also determine the attention that the business will receive. We will aim to predict the attention a particular business receives at different times in future. Specifically, we will predict the amount of attention a business is currently getting time which basically tells us that the amount of attention a business is currently having is in line with the expectation of the Yelp market. We will also predict how much attention the business should get in the next six months which will explain the fashion or direction that their business is following - will their attention increase or decrease in the near future? We will mine a Yelp-provided data set to try and learn features with respect to different kinds of business. This data is limited to businesses, users, and review text in Phoenix, Arizona. Our work involves extracting sets of features to use in our model. We describe two methods: simple manipulation of the given data and natural language processing on reviews provided by the users. We will then use various feature selection and analysis methods to select the features that will help us best predict.

1. **DATA DESCRIPTION**
   1. **11008 Business Record File**

* Type of Business
* ID of Business
* Name of business
* Latitude of business
* City
* Longitude
* State
* Address
* Number of stars
* Review Count
* Categories
* Whether open or not



# Fig 1. HeatMap showing review count in Phoenix

**2.2 187928 Reviews**

* Review Type
* Business ID
* Review Text
* Date of Review
* Number of votes and type of vote
* ID of User
* Number of stars

**2.3 43672 User Records**

* User Type
* User ID
* User Name
* Review Count
* Mean of number of stars
* Vote count and Type of Vote

1. **DATASET GENERATION**

In this section, we ran the data generation code on the 4 json files for Business, User, Review, Tip and generated a single dataset by doing a join on the Business and Review table on business\_id, Business and Review and User table on user\_id, Business and Review and User and Tip table on user\_id and business\_id. After getting the data set we filtered it on the the basis of city = “Phoenix” and for a single category of business, the category being “Restaurants” so that we get a workable size of the dataset.

1. **FEATURE EXTRACTION**

We have generated additional features via 2 methods. We have done simple manipulation of the data to create features that are time dependent. Then we have done sentiment analysis on the raw review text and then used the results as a usable feature vector.

4.1 Extracting features

I have generated additional features which would be helpful in predicting the future of business. We have basically generated 2 types of features.

1)Features containing metadata about the business.

* Latitude
* Longitude
* Businesses within 1 km –we have done geographical clustering using Geographical\_Cluster.R of businesses using Latitude and Longitude and then found out the number of businesses a particular business has around it by finding out the number of business in one cluster.

2)The second category contains features which gives us information about reviews for a particular business. We have used the reviews data with respect to the businesses and have come up with below features.

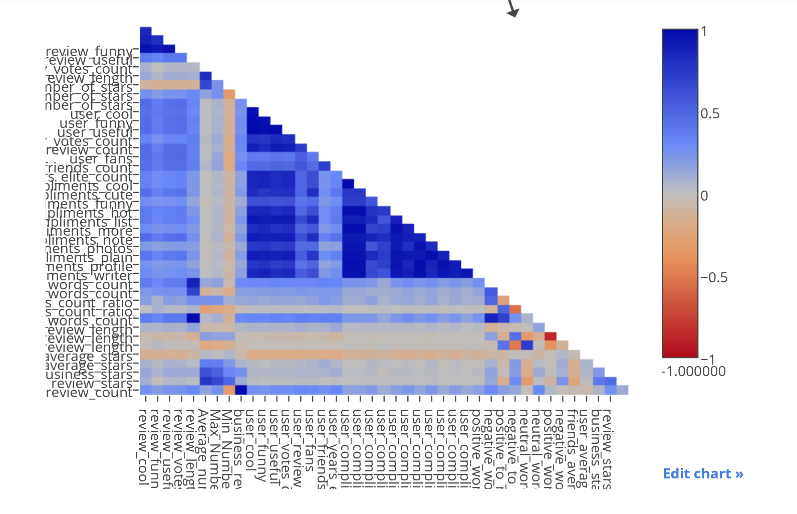
* Review count in the set
* Star Average Ratings over reviews
* Max Star Count
* Min Star Count
* ‘Cool’ vote count for reviews
* ’Funny’ vote count for reviews
* ‘Useful’ vote count for reviews
* User Count posting the reviews
* Days count since first review posted.
* Days count since last review posted.
* The duration of attention (last review date – first review date)

The problem of coming up with feature vectors then boils down to generating feature subsets which could be of help in predicting the amount of attention that a business will get and then generating the above feature vector for each subset.

One subset of features is the collection of the reviews the business have received until now. These features can be very useful in answering the crucial question: We can predict future attention for a business by using the information which tells us the amount of attention the business has already received so far.

We are also expecting that review text of businesses sharing different with our business can help predict the amount of attention the business will get in future. In particular, we believe this information will be helpful. We are also expecting features denoting location of business to be helpful for predicting attention. Therefore, we come up with a subset that contains the list of business that are present within a km of the target business.

Finding Correlation between features – After generating all feature vectors we have also found out the correlation between all the features which gives us a matrix with the correlation values(matrix available in YelpFeatureExtraction\_Final.ipnyb) which enables us to basically tell at a glance which features would be more useful in predicting the future business.



# Fig 2. Correlation between different features

4.2 Generating Features WRT Reviews

Raw review text can be mined and can be useful in numerable ways for a prediction model - sentiment analysis, keyword associations and ngram analysis. Our motive was to come up with the most frequent keywords within all reviews of restaurants and then divide them into positive words and negatives words and then use the count of these positive and negative words as feature vectors.

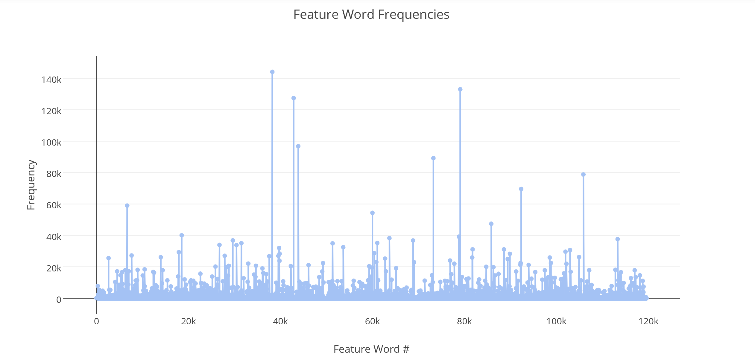
With this goal, we have used NLTK to come up with a feature vector containing the counts of the number of positive and negative word counts. This is done in two steps

- First, we compute the top keywords among all the reviews.

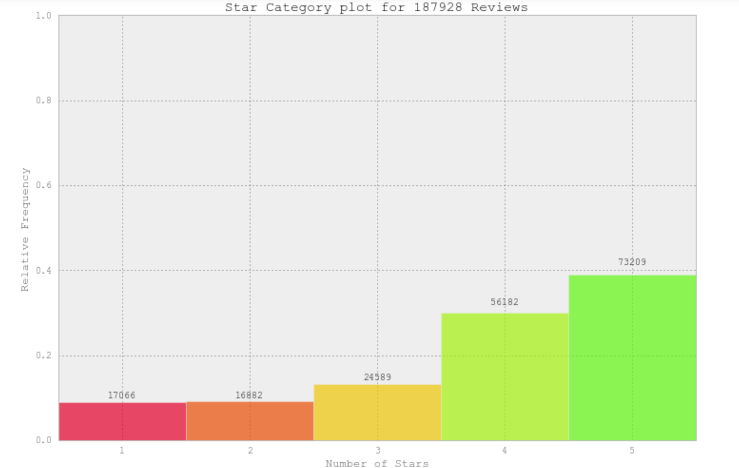
- Second we count positive and negative words

4.2.1 Review Text Feature Generation - Feature Generation from the review text starts with first preprocessing and removing unwanted things from the text(removing stopwords, checking spellings, removing unnecessary punctuations). We them perform tokenization. Each review is tokenized into sentences. Post that each sentence is tokenized into different words. We have used Python Natural Language Toolkit (NLTK) for this process.

4.2.2 Positive and Negative word count extraction - After finding the total number of words as feature vectors we compare them to the positive word document and the negative word document to find the positive words count, negative words count and the neutral words count. We use this information as feature vectors.



# Fig 3. Plot showing feature words with frequencies



# Fig 4. Plot showing number of stars with relative frequencies

4.3 User Clustering

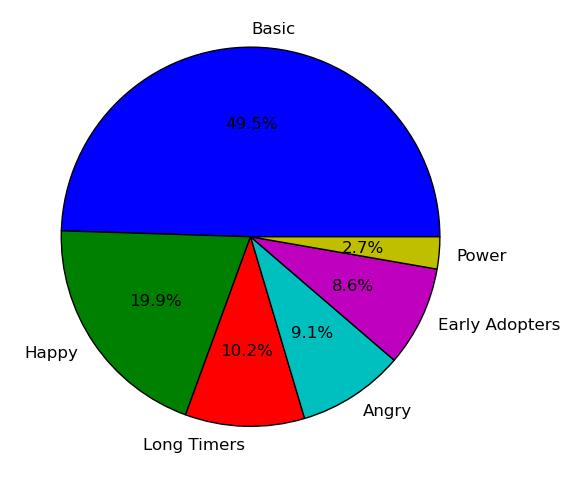
In order to predict the attention a business has along with the rating, having details about users is also very necessary. The user count is extremely huge, so a better approach way to handle with users would be clustering. We have used the below list of user features below, we have perform clustering of users using k-means.

|  |  |
| --- | --- |
|  | * Review Count |
|  | * Average number of stars |
|  | * “Funny” votes count * “Useful” votes count |
|  | * “Cool” votes count |
|  | * First review posted date |
|  | * Last review posted date |
|  | * Numbers of days in between first and last review |

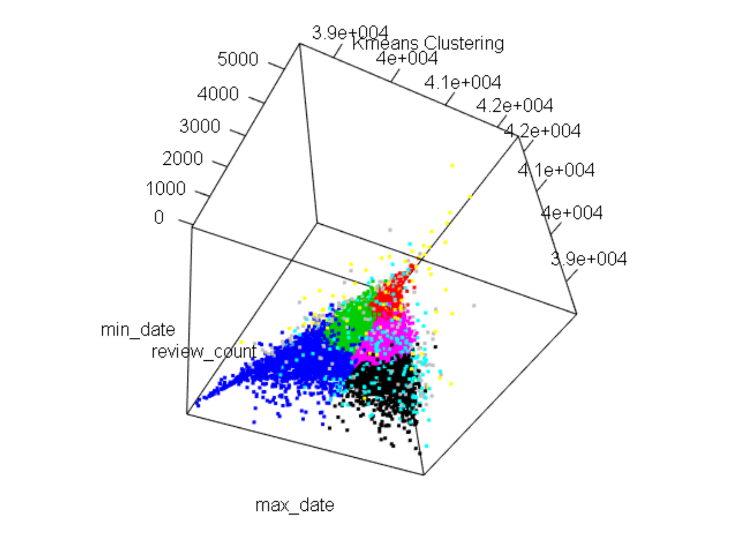
We basically labeled the clusters by clustering them using K Means clustering algorithm as below –

The labelings were basic users, power users, happy users, long time users, angry users, early adopters.

* Basic users - 30 reviews on average and were not active for a long period of time.
* Angry and happy users had written few reviews each, and gave a very high or low rating to a business and never bothered to write reviews again.
* Long time had written over 50 reviews on an average, and were there for 3 years.
* Early adopters wrote 50 reviews, had written the first one just after the launch of yelp.
* Power users wrote more than 800 reviews.



# Fig 5. Pie Chart showing user clusters



# Fig 6. User Clustering via K Means

1. **FEATURE SELECTION**

5.1 PCA

Principal component analysis (PCA) is used for reduction of dimensionality of a feature space. It is useful when the feature count is too large for a regression model or classification model to be computationally feasible or when large amounts of data is contained within noise. So, for reduction of dimensionality, PCA selects orthogonal vectors in the feature space that maximize variance. This way, the features are compressed and a few principal component vectors can describe a large portion of the feature space. The disadvantage to using PCA is there is information loss in the compression and also there is no guarantee that the principal components will be separable.

5.2 Feature Selection

Quite often extracted and generated features contain irrelevant and also redundant information. Having a lot of these poor features can compromise the performance of a learning algorithm because there is a lot of noise in the data. The best features are those that make data predictable.

We have employed the below techniques for feature selection –

* Naive approach to selecting features is to search exhaustively over all possible sets that we can generate from features and to select the set with lowest error. For huge datasets with a lot many features, it is not a very feasible. But we have used domain knowledge and have come up with subsets of features to predict the attention that the business will receive in the future. Instead of coming up with random subsets we have come up with logical subsets.
* Univariate Feature Analysis- A very efficient approach to selecting features is to look at the prediction quality of each feature separately. The best features are then used for the prediction model. The major advantages of this method are that it is fast as well as simple. However, a major disadvantage is that combinations of bad individual features can often combine to create a good prediction model.
* Combination of PCA and Univariate Feature Analysis. We have combined both the PCA to reduce the dimensionality and Univariate Selection considering the case that some original features were good and have come up with combined features and then pipelined this function for different values of PCA components and the value of k in univariate feature selection.
* RandomForestRegressor - I have also used the RandomForestRegressor for the feature importance measure exposed in sklearn’s Random Forest implementation which is a univariate selection method.

1. **PREDICTION**

We chose the below models for classification and regression models. We can imagine that an implementation of our algorithm will show up as a statistic for every business.

SVM - Because SVMs provide very fast prediction performance, this will likely hold up on running numerous predictions once the data sets become bigger. It aims to find a function f(x) that does not deviate more than some margin away from the training data. By having the epsilon margin, the training process can determine a subset of training points as having the most influence on the model parameters. The points with the most influence end up defining the margin and are called support vectors. Thus, the final model that is produced depends only on a subset of the training data. Predictions then run only with these smaller number of support vector.

Random Forest – As part of their construction, RF predictors naturally lead to a dissimilarity measure between the observations. One can also define an RF dissimilarity measure between unlabeled data: the idea is to construct an RF predictor that distinguishes the “observed” data from suitably generated synthetic data. The observed data are the original unlabeled data and the synthetic data are drawn from a reference distribution. An RF dissimilarity can be attractive because it handles mixed variable types well, is invariant to monotonic transformations of the input variables, and is robust to outlying observations.

1. **RESULTS**

7.1 TEMPORAL PREDICTION

In this section we describe results for a classifier which is basically predicting the attention the business is going to receive in the near future by training it with the amount of attention the business has already received. Here we consider the businesses which have listed ’Restaurants’ as one of their categories. This gives us a set of 3002 businesses. We have generated another feature vector called Business\_Attention which is directly in line with the number of reviews the business has already received. The figure shows the review counts received with respect to month for the business which have restaurants listed as one of their categories. For every business we have assigned values from 1 – 5 which denotes the attention the business has received 5 being the highest.

For every experiment reported we have tried to come up with a model which can be used for the following: Given all the data about a particular business and all reviews in the Yelp database logged before a particular date, we are predicting the amount of attention that the business will receive during a year starting from the target data.

We do three kinds of predictions and post results

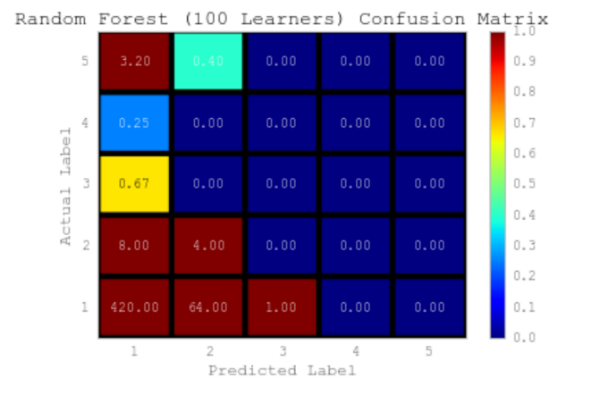
* Prediction using review text features
* Prediction using metadata of business
* Prediction using original features.

7.1.1 Prediction using original features.

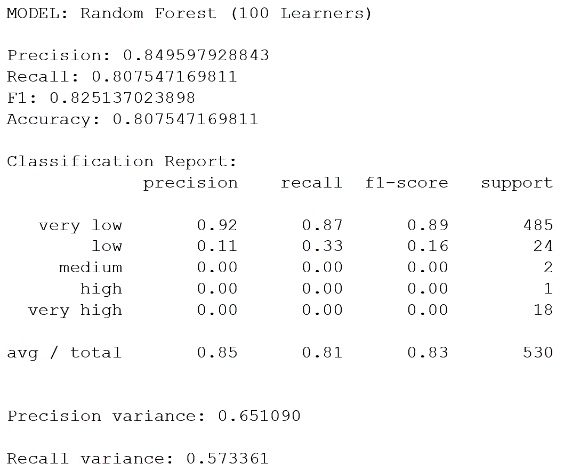
Here we are using one single date and we will be using this as the target date. Basically, we come up with one particular date and use all reviews till the particular date and try and train the prediction model. The date we are selecting is 2014-05-06. This time is in almost the midpoint of dataset and has a huge review count that were entered before the selected date, while also having a huge review count entered in the 1 year after that date.

In the next step we have built features for each of the 3002 restaurants which have received reviews before the date that we have selected. Initially we have built a model using the features originally present in the dataset Using PCA and Univariate Feature Analysis we try and select best features and build our training model. We have selected the best twenty five percent of the features. The below list denotes features which were chosen by our algorithm.

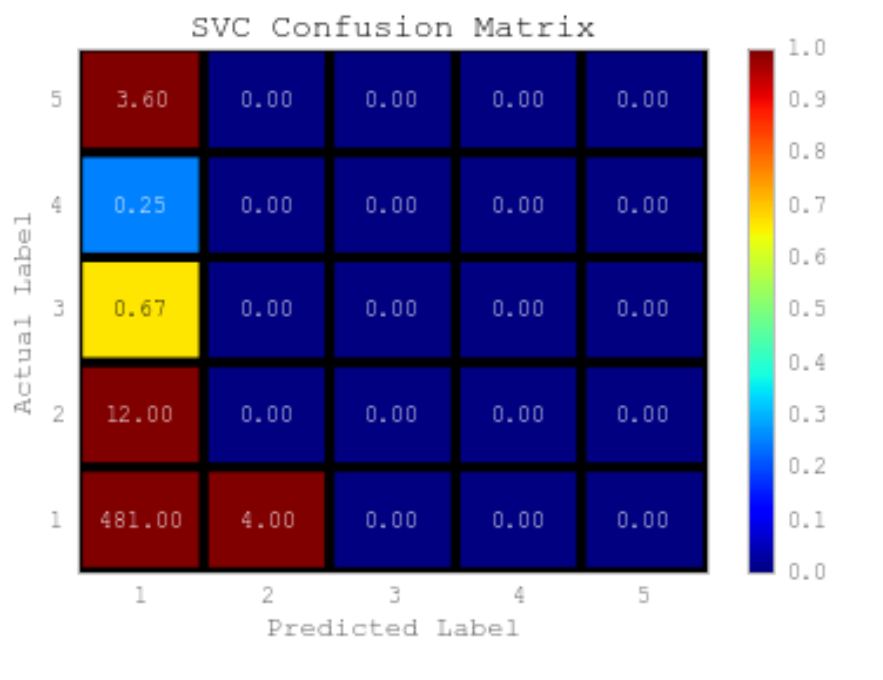
* Review count for the business before the target date
* Review Count tagged as ’cool’
* Review Count tagged as ’funny’
* Review Count tagged as ’useful’
* User count entering reviews for the business.
* Number of review votes



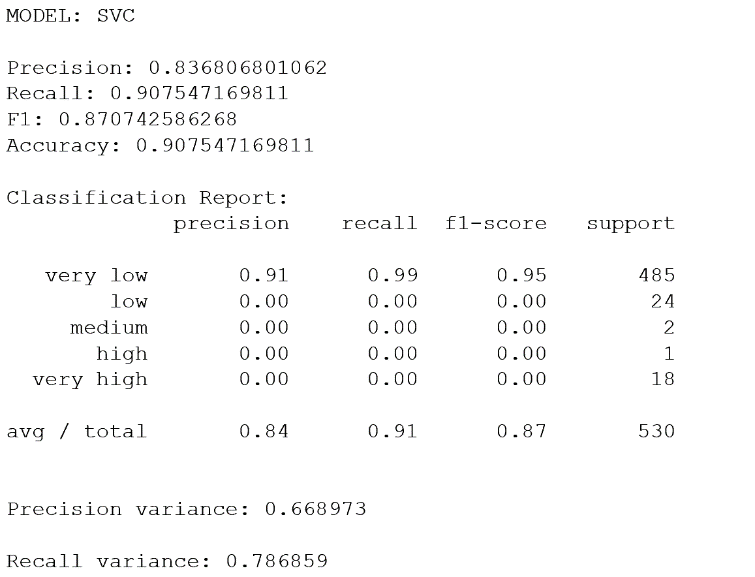
# Fig 7. Confusion Matrix for Random Forest Model



# Fig 8. Classification Report for Random Forest Model



# Fig 9. Confusion Matrix for SVC Model



# Fig 10. Classification Report for SVM Model

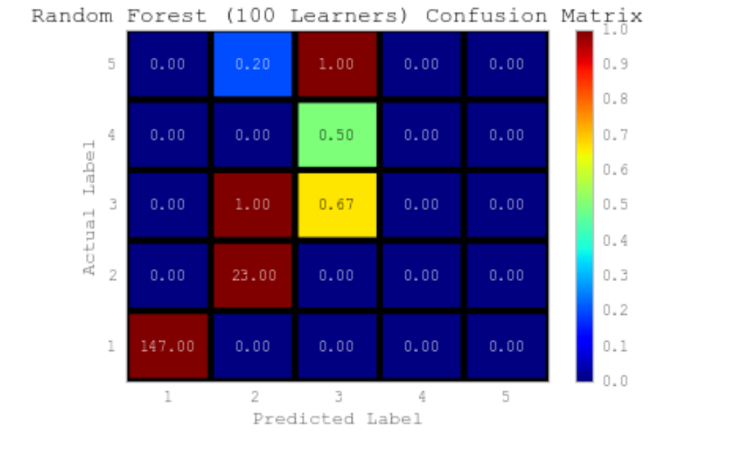
7.1.2 Prediction using Metadata about the business.

Here we are using one single date and we will be using this as the target date Basically, we come up with one particular date and use all reviews till the particular date and try and train the prediction model. The date we are selecting is 2014-05-06. This time is in almost the midpoint of dataset and has a huge review count that were entered before the selected date, while also having a huge review count entered in the 1 year after that date. Here initially we generate a vector of 43 features using methods described in the Feature Generation section highlighting metadata about each business.

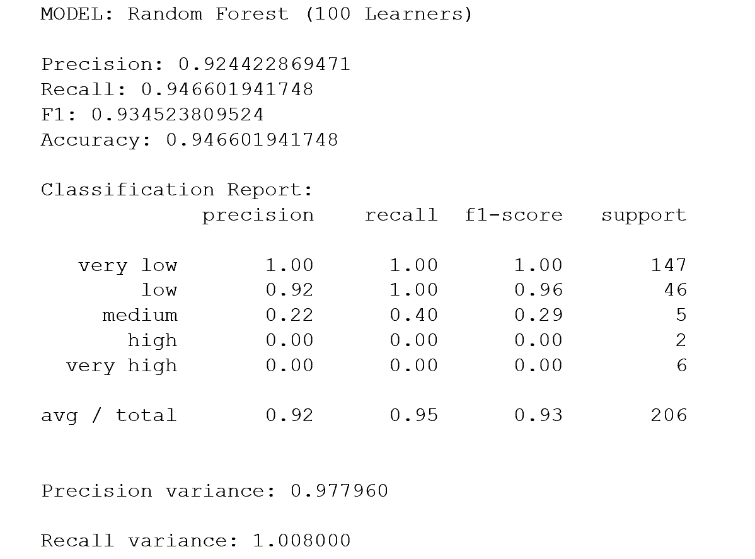
In the next step we have built features for each of the 3002 restaurants which have received reviews before the date that we have selected. Initially we have built a model using all 43 features. Using PCA and Univariate Feature Analysis we try and select best features and build our training model. We have selected the best twentyfive percent of the features. The below list denotes features which were chosen by our algorithm.

* Review Count for the business
* Max stars for review entered for the business before date that is selected.
* Review Count logged for the business marked as ’cool’
* Review Count logged for the business marked as ’funny’
* Review Count logged for the business entered as ’useful’
* Number of users entering reviews for the business
* Days count since review that was last entered.
* Days count in between reviews that were first and last entered.
* Max review count for business entered in a single day before the target date
* Business count within one km

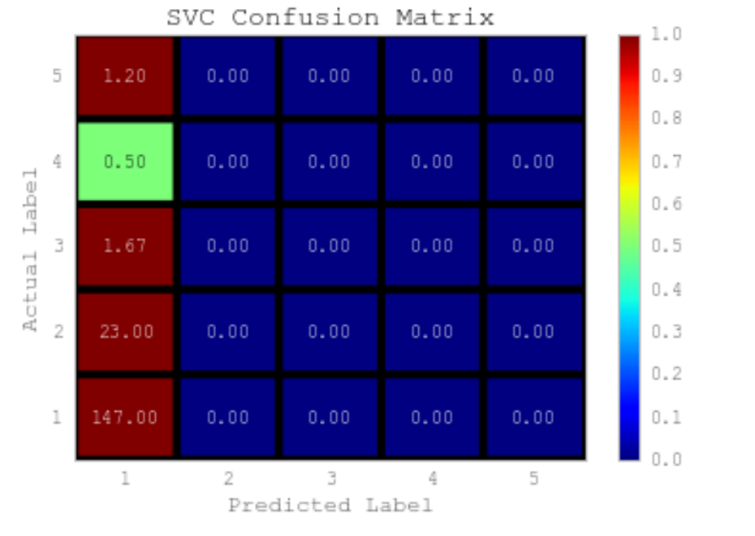
Results after Feature Vector Reduction –



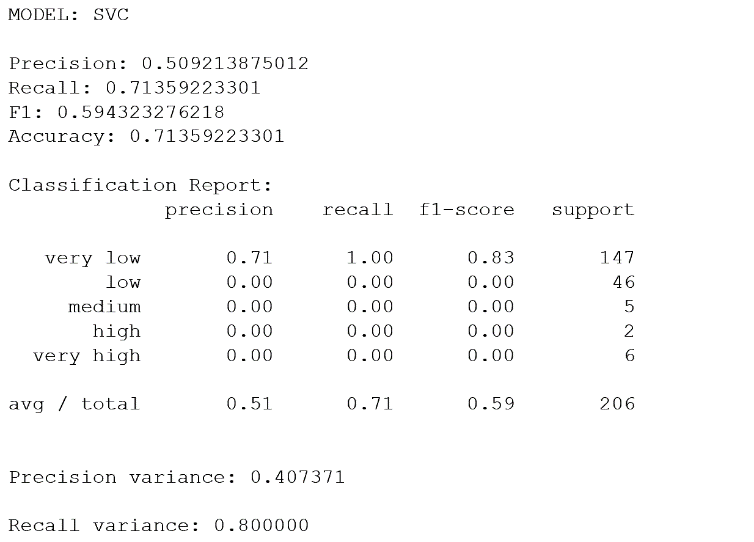
# Fig 11. Confusion Matrix for Random Forest Model



# Fig 12. Classification Report for Random Forest Model



# Fig 13. Confusion Matrix for Random Forest Model



# Fig 14. Classification Report for SVC Model

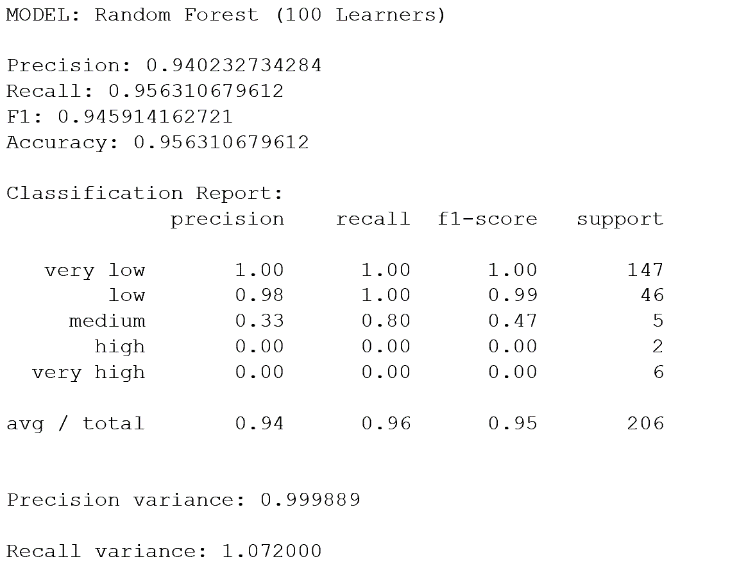
7.1.3 Prediction using review text features.

Here we are using one single date and we will be using this as the target date. Basically, we come up with one particular date and use all reviews till the particular date and try and train the prediction model. The date we are selecting is 2014-05-06. This time is in almost the midpoint of dataset and has a huge review count that were entered before the selected date, while also having a huge review count entered in the 1 year after that date. We then generated a number of feature vectors that we got by doing NLTK processing and sentiment analysis over the review text.

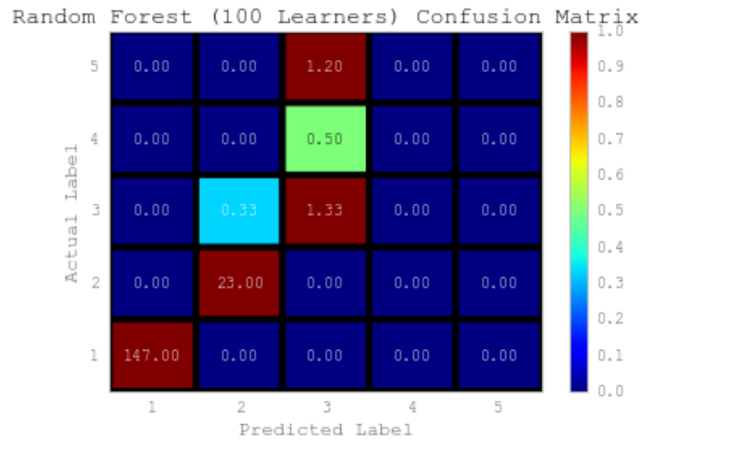
* Review Count for the business
* Number of positive word count in the review texts for the business
* Review Count logged for the business marked as ’cool’
* Review Count logged for the business marked as ’funny’
* Review Count logged for the business marked as ’useful’
* Number of negative word count in the review texts for the business
* Number of neutral word count in the review texts for the business
* Number of user votes entered as cool, funny, useful.
* Number of stars for the business
* We took into consideration the user compliments and generated feature vectors out of the complements.
* Max reviews in a day
* Min reviews in a day

Using PCA and Univariate Feature Analysis we try and select best features and build our training model. We have selected the best twenty five percent of the features. The below list denotes features which were chosen by our algorithm.

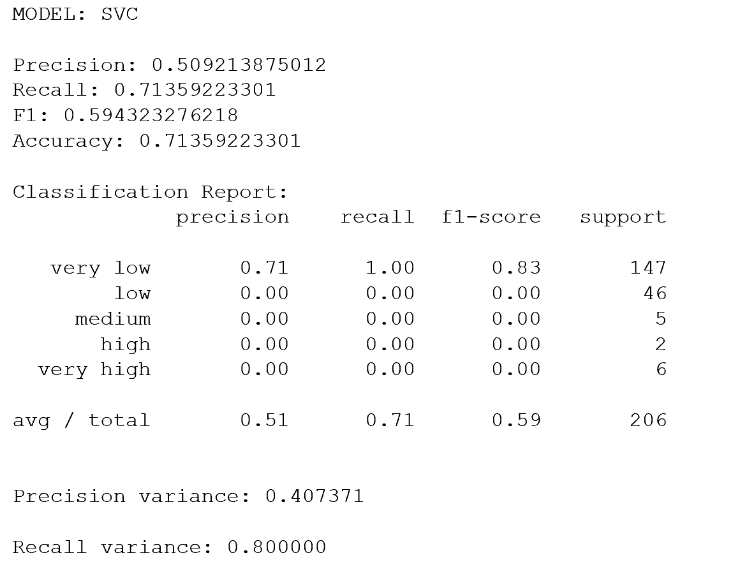
* Review Count for the business
* Number of positive word count in the review texts for the business
* Review Count logged for the business marked as ’cool’
* Review Count logged for the business marked as ’funny’
* Review Count logged for the business marked as ’useful’
* Number of negative word count in the review texts for the business
* Maximum review count for a business in a day



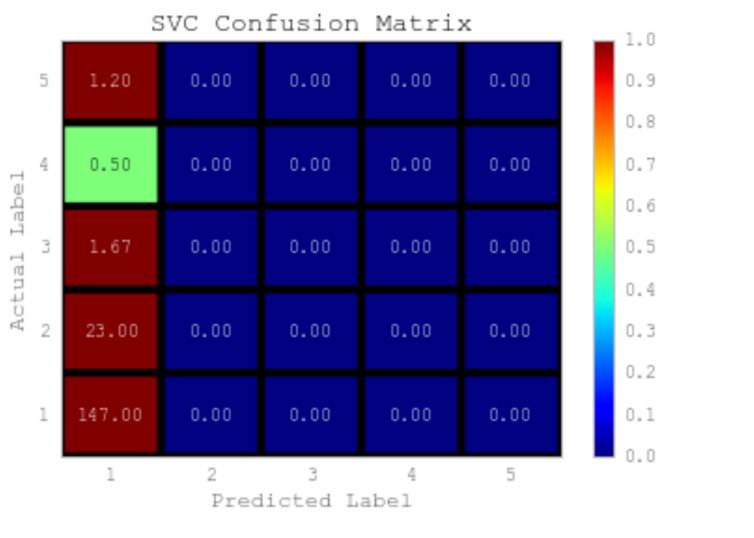
# Fig 15. Classification Report for Random Forest Model



# Fig 16. Confusion Matrix for Random Forest Model



# Fig 17. Classification Report for SVC Model

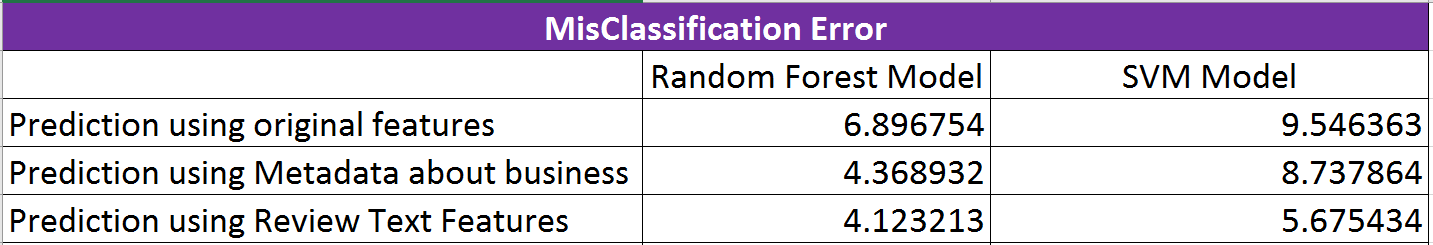


# Fig 18. Confusion Matrix for SVC Model

7.2 PERFORMANCE COMPARISON

We compare the performance of the 3 types of predictions that we have done above and have come up with the below results

* The Random Forest Model gives a better performance or prediction results as compared to the SVM model This could be the case as we have not scaled the data and this step could be taken in future.
* The Review features and the Metadata about the business feature vectors give fairly better results as compared to the original set of features used.



# Fig 19. Mis Classification Error Comparison

# 

# Fig 20. Prediction Accuracy Comparison of 4 Businesses

* The prediction models perform better after dimensionality reduction by PCA and Univariate Feature Analysis in comparison to when we use all features except for the review text feature model.
* The High success rate can be attributed to the fact as we are trying to predict the amount of attention the business is going to receive only on a scale of 1 to 10.
* However, we can employ the same approach followed in this paper to predict the number of reviews a business will receive in future say a period of n months. The prediction models would work but with a higher error rate as the number of values to be predicted would be high.

1. **CONCLUSION**

In this report, we have explained a series of steps that is used to predict the amount of attention a business will receive in the future using various subsets of features. These features were generated from text of reviews and statistics of business. In our prediction models, we analyze different subsets of features to see their result, and then try to predict the amount of attention a business are going to get in the next few months. We find that of all features, the metadata about the business features provided the best results. The features we generate are very interesting and will be very useful for predicting various things and coming up with more inferences. In future work, we will predict other things with same type of features, such as to predict the number of stars a user will give to a business depending on the text of the review. We are also motivated to generate additional and more interesting features to improve our results.

1. **REFERENCES**

[1]https://pdfs.semanticscholar.org/64b8/553bd62997d332910078bcd5ee74a43f9350.pdf

[2]https://www.yelp.com/html/pdf/YelpDatasetChallengeWinner\_InferringFuture.pdf

[3] https://en.wikipedia.org/wiki/Support\_vector\_machine

[4] https://en.wikipedia.org/wiki/Random\_forest