Multinomial Success Classification of New Restaurants

Denim Dutta, Harsha Illuri, Keshav Rajput, Rohith Vajrala, Tarun Kolla, Tej Tharang Dandala

Arizona State University

[***datta4@asu.edu***](mailto:datta4@asu.edu)***,*** [***hilluri@asu.edu***](mailto:hilluri@asu.edu)***,*** [***ksrajput@asu.edu***](mailto:ksrajput@asu.edu)***,*** [***rvajrala@asu.edu***](mailto:rvajrala@asu.edu)***,*** [***tkolla@asu.edu***](mailto:tkolla@asu.edu)***,*** [***tdandala@asu.edu***](mailto:tdandala@asu.edu)

**Abstract**

This project aims at conducting research and analysis on Yelp challenge dataset to predict the success rate of a new restaurant given its location. We start with extracting the dataset into appropriate format followed by data cleaning and applying various binary and multinomial machine learning classification algorithms with accuracy around 65%. Applying principal component analysis (PCA) helped us in reducing the feature set from 775 to 22 **and further applying sentimental analysis on reviews helped us improve the accuracy to 85%.** Finally, were also able to improve our accuracy by formulating a new formula as success parameter from ***X.00 to Y.00.*** The different classification algorithms that we used are Naïve Bayes, Scalar Vector Machines, Random Forest, Logistic Regression and Neural Networks. We end up getting the best accuracies for ***Random Forest and Neural Networks***

**1 Introduction: a summary of the problem, previous work, methods, and results**

**1.1 Summary**

With an already existing competition in the restaurant business (which is evident by the fact that 59% of restaurants fail in its first year while 89% in first five), it is important to have proper research done before putting a step forward. Our project provides a restaurant entrepreneur with a guide on his success or failure chances based upon the location, cuisine and various other parameters. We focus on finding the most important parameters responsible for a restaurant’s success and defining a model to predict the success rate by applying various machine learning methodologies.

**1.2 Problem Description**

The primary goal of our project is to provide success rate prediction for new restaurants. We achieve this by dividing our main goal into set of goals as:

1. Pre-processing of the dataset after retrieving it and converting it into appropriate format.

2. Reduce the dimensionality of the data applying principal component analysis.

3. Predict restaurant success using binary classification based upon the success criteria of reviews greater than 20 and average star rating over 3.5.

4. Predict restaurant success using multinomial classification showcasing success rate by rating ranging from 1 to 5.

5. Formulating a new success parameter to boost the accuracy of the predictive model.

**1.3 Previous Work**

There has been a lot of research work done using the Yelp challenge dataset and there are a few papers out there predicting the success rate of restaurant as well.

A paper by Vasa, Vaidya, Kamani, Upadhyay and Thomas also utilized the Yelp dataset to predict restaurant success using binary classification [2]. They investigated only restaurants in the Phoenix area and relied heavily on the type of food that restaurants served to make their predictions, attributing the importance of the food to the demographic layout of the city. They achieved final accuracy of 51%.

A paper by Aileen Wang, William Zeng, Jessica Zhang from Stanford University offers better accuracy (85%) by utilizing Chi-square to narrow down the number of features and later applying sentimental analysis on the restaurant reviews. Their paper is based on predicting success for restaurants in different cities across the country using binary classification as well as multi-class classification. The success parameter that they utilize is average rating greater than 3.5 and reviews more than 20. It is evident that this parameter is solely dependent on the reviews and does not consider other factors.

We are primarily focusing on formulating a new success parameter by utilizing the user reviews and user ratings provided based upon the feedback their reviews recieve, along with the average star rating. We apply Principal Component Analysis for dimensionality reduction and identifying most important features useful for the prediction.

**2 Methodology: A detailed description of methods used**

**2.1 Data Preprocessing**

In this project we have used the yelp dataset provided in the https://www.yelp.com/dataset/download. The reason for selecting Yelp is because of its popularity and large user base in America. The dataset shared to us was a huge 6.52 gigabytes dataset with business.json, user.json, reviews.json and tips.json files in it. We had to do considerable amount of preprocessing before we could apply any feature extraction techniques or machine learning algorithms. Initially we have to convert all the json files to csv and convert all the nested dictionaries to columns in csv. The business.json served as the main file with all the unique business id in it. There were a total of 174k businesses and 58k restaurant’s in which where extracted using the keywords such as ‘restaurant’, ‘food’.

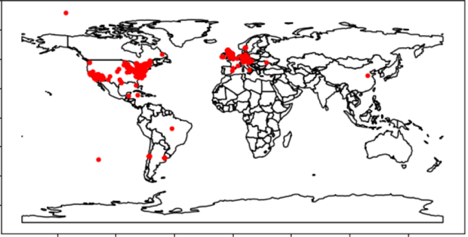
****

Fig 1 : Restaurant Population Density

From the review.json we extracted review\_id, user\_id, stars, useful, fun, cool and text.

We are performing semantic analysis on the text feature which is the review of the user of predict if the review is happy, neutral or angry. After performing the semantic analysis, we see that the results of the analysis is correlated with the text length. After this finding we removed the text feature from the review.json dataset and appended the text length to it.

For the user.json we have extracted user\_id, review\_count, useful, funny, cool, fans, average\_stars, compliment\_hot, compliment\_more,compliment\_profile, compliment\_cute, compliment\_list, compliment\_note, compliment\_plain, compliment\_funny, compliment\_writer and compliment\_photos. These features are used to add weights to the user review while calculating the formulated output parameter.

**2.2 Feature Conversion in business.json :**

The file “business.json” contains the details of all the businesses in the yelp dataset. The dataset contains generic features like the Business ID, Name and Address as well as quantitative features like latitude-longitude coordinates and rating and also binary value for attributes like whether or not the restaurants accept credit cards. The data is represented in various data structures like nested, double nested binary data of varying lengths, arrays of varying lengths containing strings, numerals, binary and empty values. Cleaning this data consists of two steps – Extracting the nested data into separate vectors and Replacing the string data with numerals and binary values.

**2.2.1 Extracting nested data**:

The data contains nested data of three formats

Nested attributes : { BusinessAcceptsCreditCards : TRUE }

Double Nested attributes : { Ambience : { classy: TRUE, hipster: TRUE, divey : FALSE }}

List attributes : { categories – [‘Mediterranean’,’Indian’,’Burgers’]}

Python scripts are run to extract each of these nested data into vectors that can appended to a csv file. The column vectors that are produced after this step are of the following form

Nested BusinessAcceptsCreditCards [TRUE, FALSE, TRUE, TRUE …. (one for each business)]

Double Nested Ambience\_classy [TRUE, FALSE, TRUE, TRUE ….. ( one for each business)]

Ambience\_hipster [TRUE,TRUE, FALSE, FALSE ….. ( one for each business)]

Ambience\_divey [TRUE, FALSE, TRUE, FALSE …. ( one for each business)]

List Categories\_Mediterranean [TRUE (if present in list), FALSE ( if absent in list )]

Categories\_Indian [TRUE, FALSE ……………… (one for each business )

Categories\_Burgers [TRUE, FALSE ……………….. (one for each business )]

**2.2.2 Cleaning the extracted data:**

The data extracted from the previous step contains the unrolled data from the nested data, String, numerical and binary data. A lot of data from multiple columns is empty. To fill in these gaps and make the entire data set readable by machine learning algorithms, these empty columns are filled with 0s. All the data representing TRUE and FALSE either in string format or binary are replaced with 2s and 1s respectively. The values that represent more than two classes or strings are replaced with increasing numerical values. For example, one of the attributes ‘RestaurantsAttire’ contains values like ‘casual’, ’dressy’ and ‘formal’. These values are replaced with 1, 2 and 3. In a similar fashion, all the values of the dataset are modified.

In a particular attribute; ‘categories’, each category is represented as a column vector and each business id in the dataset corresponds to every category of the data i.e. if a business falls under a category, then the corresponding value is 1 and in any other case, value is 0.

After this preprocessing and cleaning, entire dataset contains 770 features and contains nearly 57K records.

**2.3 Reducing dimensionality using PCA**

When the data has large number of features, sometimes many features measures related features, so considering everything gets redundant. So instead of whole feature set we can work with a reduced feature set. But, if we blindly select some of the features and discard rest, we might discard some important features, and thus the trained model won’t have the information about these features.

So one important aspect of reducing feature size is to take into account all the feature information. Dimension Reduction process ensures that the reduced feature set conveys similar information.

PCA projects the data into a lower dimensional feature space. For a constant desired dimension size, PCA projects the data points in such a way that minimizes the reconstruction error.

In our dataset, we have 775 features. We minimize the dimension using PCA and select the lower dimension size as 22. We have used PCA function from sklearn.decomposition module for Dimensionality Reduction.

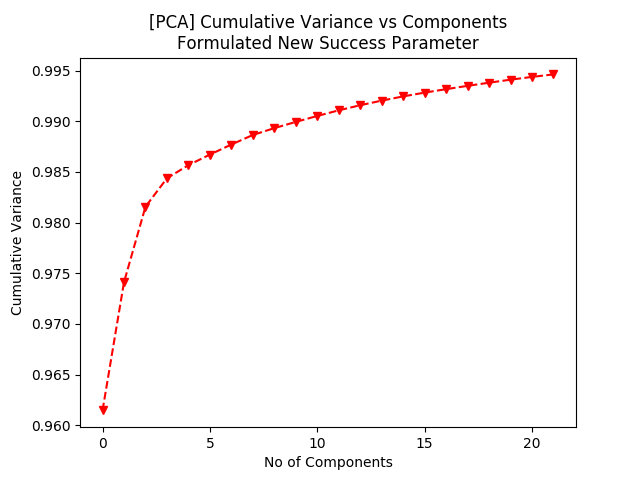
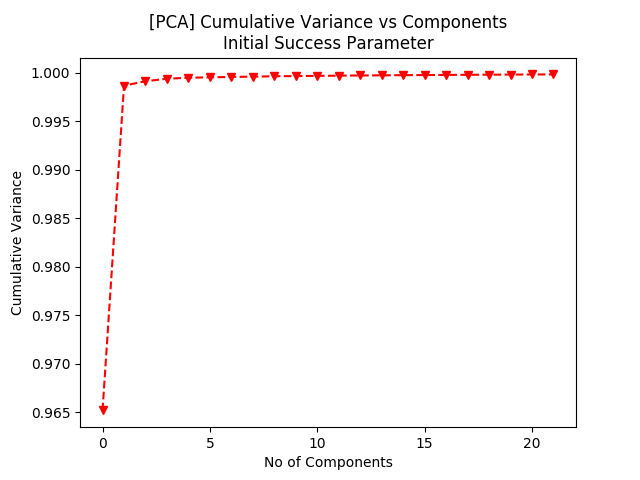


Fig 2 : PCA for initial success parameter Fig 3: PCA for New Success Parameter

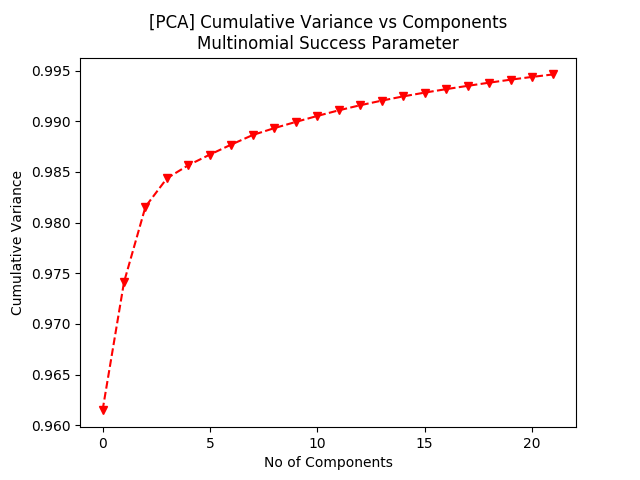


Fig 4 : PCA for Multinomial success parameter

By selecting the reduced dimension size 22, more than 99% of the variance is retained.

**Formulating new success parameter**

Success parameter predicting success with average star rating greater than 3.5 along with more than 20 reviews gave us good results. As it is evident we are solely relying solely on the star rating which is a very good parameter for success but sometimes star ratings can be vague depending upon the user. some user are very lenient while rating whereas some can be quite strict, some can have one bad experiences whereas some of them can have a single very good experience and hence it was important to consider the user rating while formulating the success parameter. Yelp data set provide us data where they rate each user who has reviewed the restaurants, number of reviews posted, feedback on their reviews etc. We utilize all these features to calculate our new success parameter described below.

1. For each user in user.csv
2. Calculate complement ratio = review count / complement flag count. (Gives and insight on the ratio of number of complements received for a particular review).
3. Calculate the sum of useful ratio and complement ratio. (This will give you a value from 0-2). Now scale this score from range 0.0 - 2.0 to a new range 1.0 - 2.0 for each user using the below mentioned formula.

For scaling values from [min, max] to [a.b]:

new\_value (x) = (((b-a) (x-min)/(max - min)) + a)

2. Join these values of user rating with the review table such that every entry in review table now shows busines\_id, review\_id, user\_id, stars, user\_rating.

3. Now, for each review\_id:

1. Calculate review\_rating as stars\*user\_rating (As minimum rating is; multiplying star rating with 1 would not add any weight to the rating but as the user rating increases, the star rating for the review will be weighted by a factor between 1 to 2).
2. Group the business\_id together and calculate the average review\_rating for each business\_id. This gives us new ratings for each restaurants.

4. Merge these ratings with the business.csv to classify the data based upon different models.

**3. Classification Techniques**

**3.1 Neural Network**

Neural network is a computational model that is inspired by biological neural networks used to predict or classify accordingly. Neural networks are organized in layers, specifically three kinds of layers – Input Layer, Hidden Layer and Output layer. The depth of the Neural network is typically determined by the number of hidden layers between the input and hidden layers.

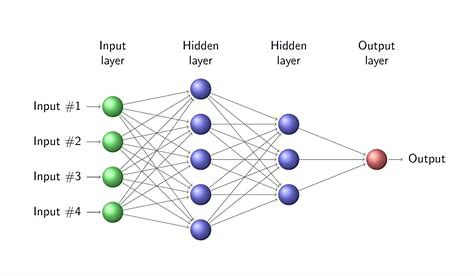


Fig 5 : Sample Neural Network

**3.1.2 Why Neural Networks**

Neural networks are widely popular for handling high dimensional data. In our case, the dimensionality is in the order of 770. Neural networks attempt to learn complex features, representations and patterns based on the number of hidden layers in the network and the number of neurons in each layer. In this project we attempted to make the neural networks recognize patterns of successful and unsuccessful restaurants and predict whether a restaurant will be successful or not.

The input dataset has been divided with a split of 4:1 into training and testing data. The first column of the data represents the ID of each business and the last column represents the label of the data. To build our neural network, Keras framework with tensorflow backend was used. We have used two neural networks, one with one hidden layer and the other one with two hidden layers. We used 770 nodes in the input layer, 300 neurons in the hidden layer and depending on whether the Neural network is a binary or multiclass classifier, there are 2 or 5 output nodes.

**3.1.3 Using initial success parameter**

We have trained a neural network to be a binary classifier with the current dataset. These results are in turn used for predicting both non-PCA data and PCA data.

For the model trained with non-PCA data, testing accuracy of 86.84% was observed. Though the training accuracy observes a steady increase, the validation accuracy observes a much more erratic increase showing that the data was going to overfit had the validation set had not been introduced. In general, the training and validation accuracy reach ~86%

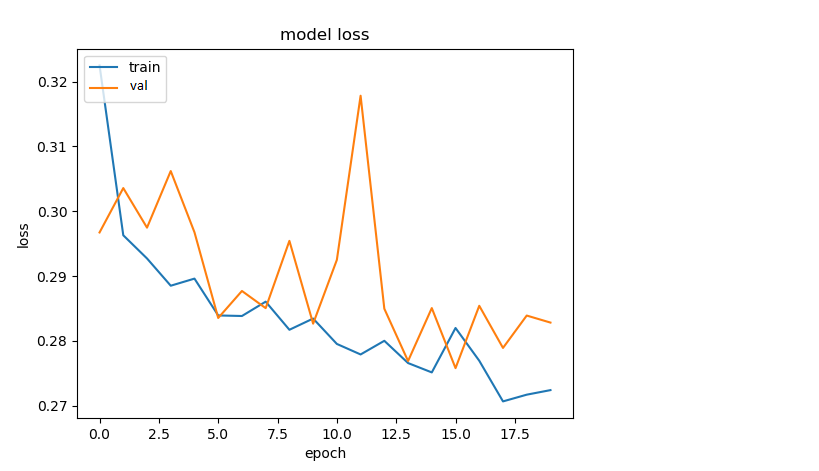
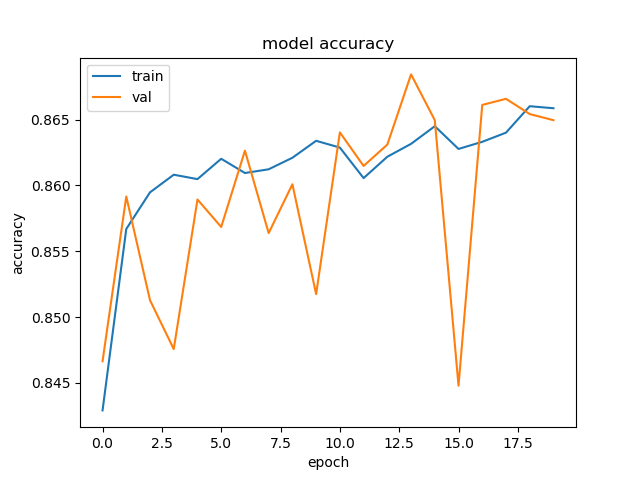


Fig 6 : Accuracy for non-PCA with initial parameter Fig 7 : Model loss for non-PCA with initial parameter

Principal Component Analysis was applied on the previous dataset and a new dataset with 22 components was obtained. The test accuracy applied on the new dataset is 87.12 % which is nearly the same as the original data set. But as we can see in the graphs below, the increase in the accuracy follows a much more linear increase indicating smaller overfitting margin.

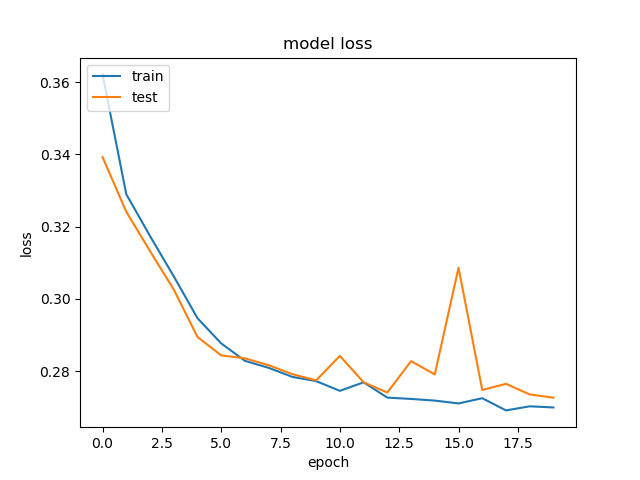
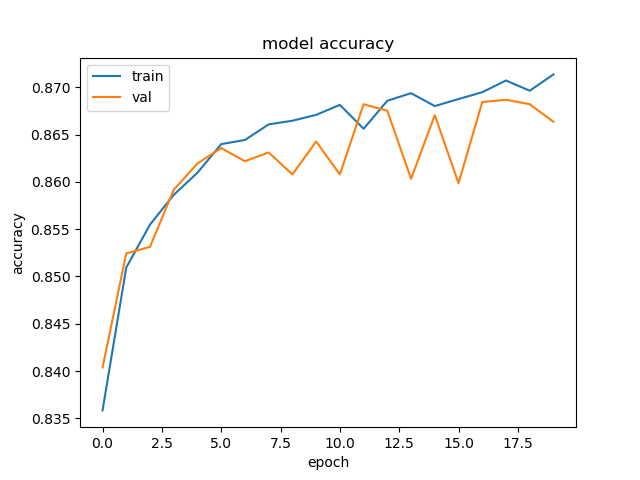


Fig 8 : Accuracy for PCA data Fig 9 : Model loss for PCA data

with initial parameter with initial parameter

**3.1.4 Reformulated success parameter :**

When only one hidden layer to train the neural network, the accuracy obtained is 51.08 % which is quite low. We can clearly observe from the above graphs that the training and validation accuracy are capped at 52% and the model loss observed is also quite high at 0.68-0.7. In an attempt to improve the model accuracy, we added one hidden layer with 100 neurons. Given below are the results obtained on that data.

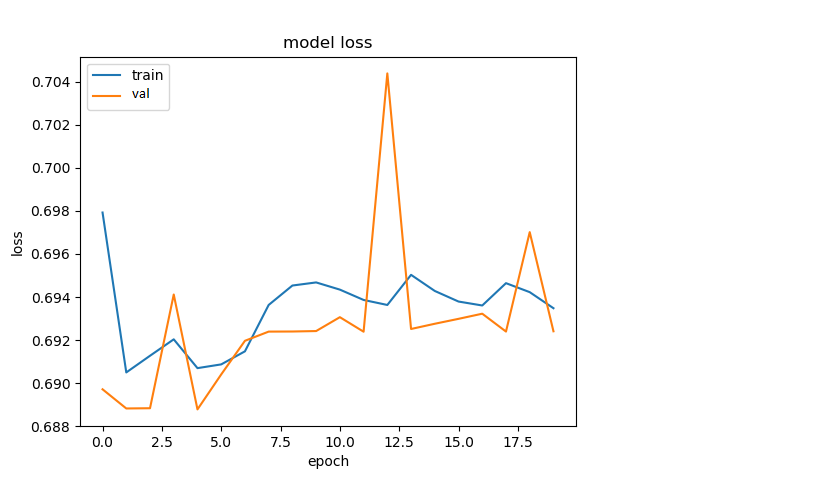
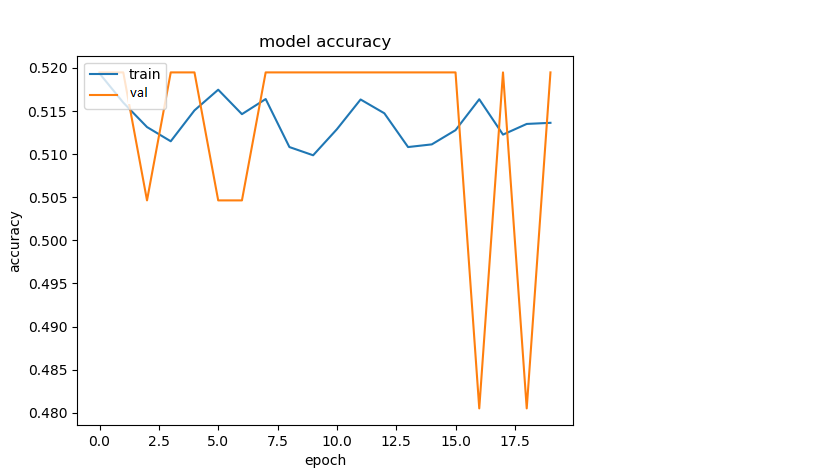


Fig 10 : Accuracy for non PCA data with new Fig 11: Model loss for non PCA data with new success

success parameter and one hidden layer parameter and one hidden layer

We observed a steep increase in the accuracy with the additional hidden layer. The maximum accuracy obtained in our trials is 95.02 %

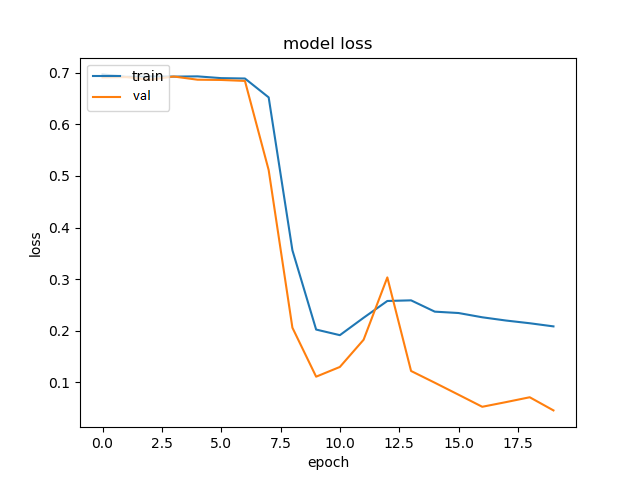
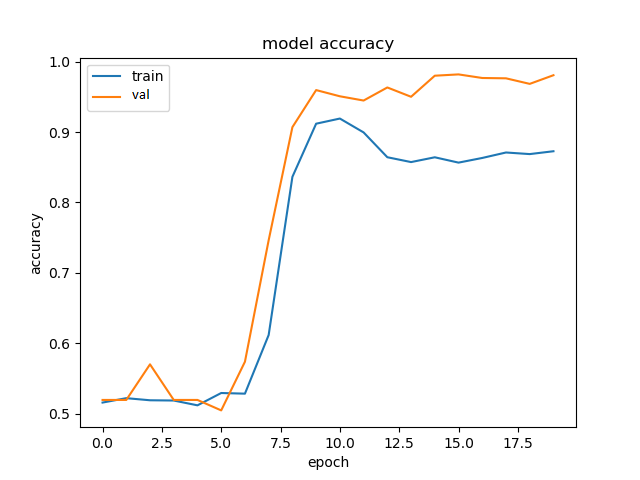


Fig 12: Accuracy for PCA data with new success Fig 13 : Model loss for PCA data with new success

parameter and two hidden layers parameter and two hidden layers

Experimentally, We performed PCA on this data set and used the network with 2 hidden layers to predict the class labels. An improved accuracy of 97.93% was obtained on the test data.

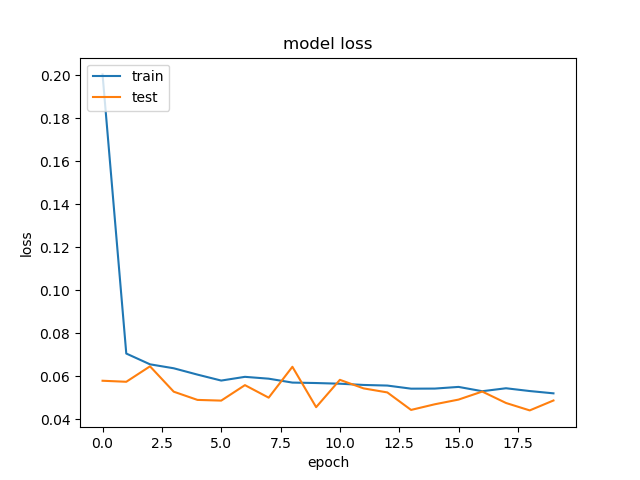
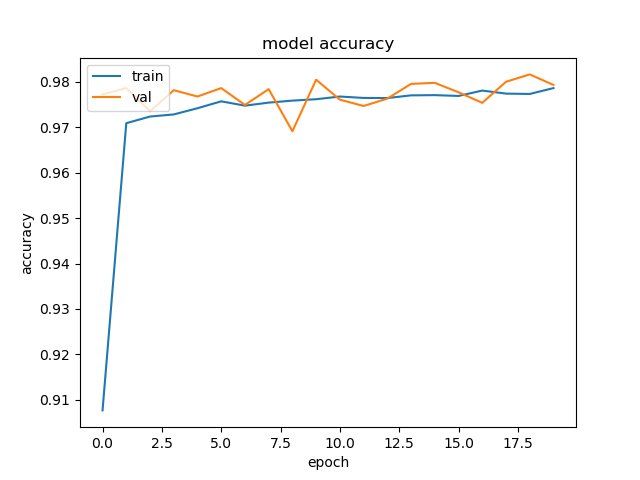


Fig 14 : Accuracy for PCA data with 2 hidden layers Fig 15: Model loss for PCA data with new success

parameter and two hidden layers parameter and 2 hidden layer

**3.1.5 Multinomial data**

We observed a testing accuracy of 94.41% on this model when multinomial data was used without applying PCA

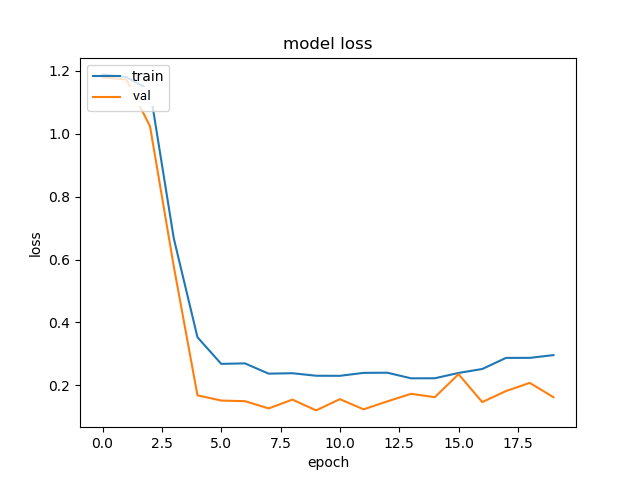
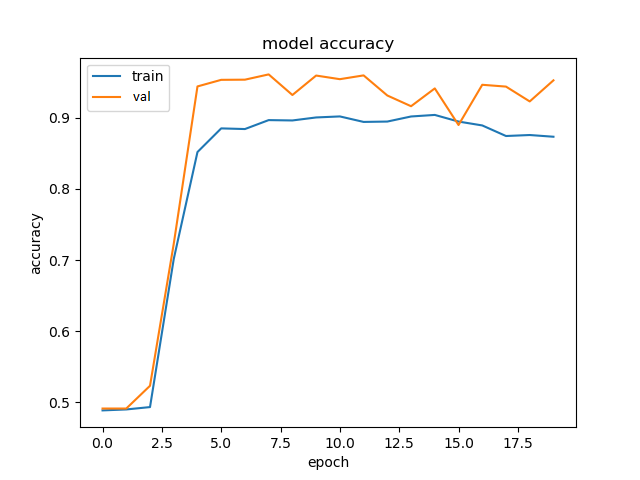


Fig 16 : Accuracy for non PCA data with new success Fig 17: Model loss for non PCA data with new

parameter for multinomial data success parameter for multinomial data

The accuracy in this case, remains almost the same, reaching 94.67 %. This indicates that performing PCA on the multinomial data is not that productive as we are consolidating the features, not allowing the neural network to capture the subtleties.

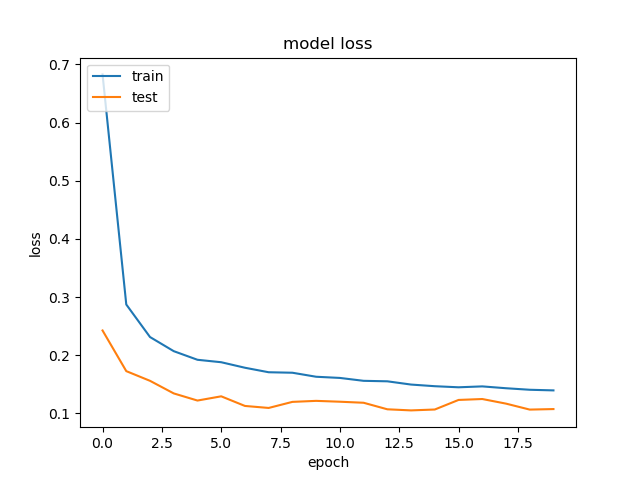
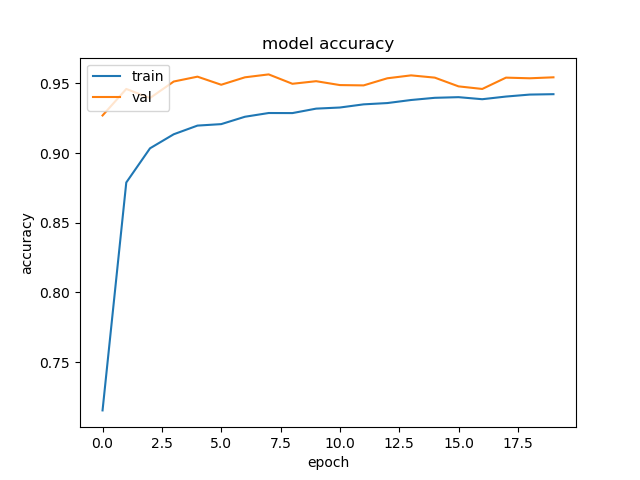


Fig 18 : Accuracy for PCA data with new success Fig 19 : Model loss for PCA data with new success

parameter for multinomial data parameter for multinomial data

Based on the results obtained in each of the above cases, Neural networks proved to be good predictors of restaurant success achieving an average accuracy of nearly 95% when an ideal success parameter is formulated.

**3.2 Support Vector Machine (SVM)**

Support Vector Machine uses kernel tricks and transforms the data, and based on the transformation, it calculates an optimal boundary between class labels. SVM creates hyperplane to separate the classes by creating largest margin between closest points and decision boundary.

We are using SVC with Radial Basis Function from svm class of sklearn package. For SVM, we used the PCA data, because of time constraint with full feature data. We are using the gamma parameter value as 1/feature\_size = 1/22.

We receive the highest accuracy when we do not restrict the SVC to a constant iteration, and we get 90.02% accuracy.

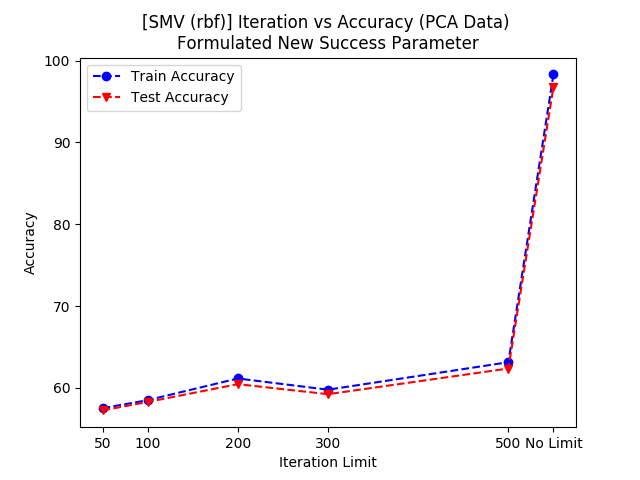
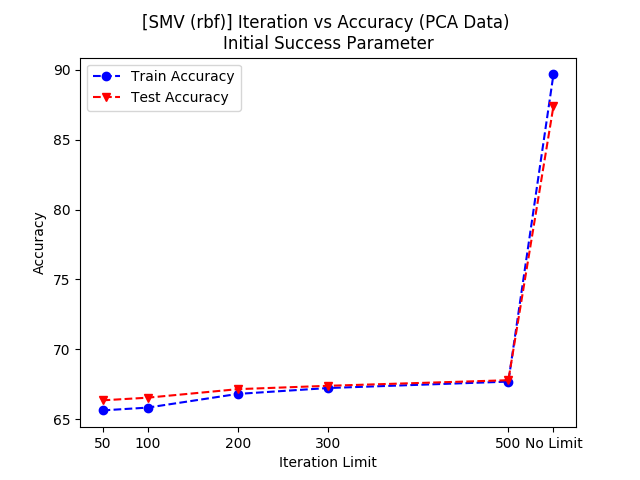


Fig 20 : Accuracy for Initial Success Parameter. Fig 21 : Accuracy for Formulated Success Parameter.

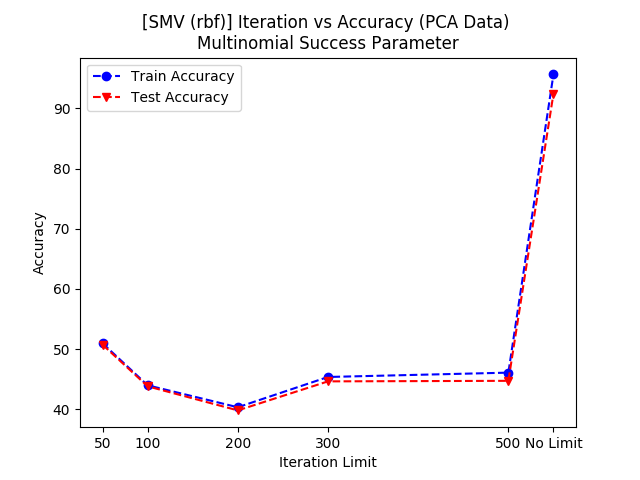


Fig 22: Accuracy for Multinomial Success Parameter.

**3.3 Logistic Regression**

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a [logistic function](https://en.wikipedia.org/wiki/Logistic_function), which is the cumulative [logistic distribution](https://en.wikipedia.org/wiki/Logistic_distribution). Thus, it treats the same set of problems as [probit regression](https://en.wikipedia.org/wiki/Probit_regression) using similar techniques, with the latter using a cumulative normal distribution curve instead.

We used logistic regression method to predict the probability of success with a Simple and efficient tool Scikit-learn. With logistic regression the dependent variable was classified based upon 775 features in the original data set and was compared with the outcomes of pca data set after dimensionality reduction to 22 features. Regularization parameter of C was set to 1.0 . We used it to classify the success of new restaurants with the predefined constraints. The data set was split in the ratios of [0.55,0.65,0.75,0.85,0.95] between training and validation data set and we observed different accuracies that have been listed below.

**3.3.1 Initial Parameter**

The graphs below represent binary classifier data with success parameters of rating > 3.5 stars and restaurants having more than 20 reviews. the fig below has an average of 82.3% for data after dimensionality reduction compared to 83.9% for original data.

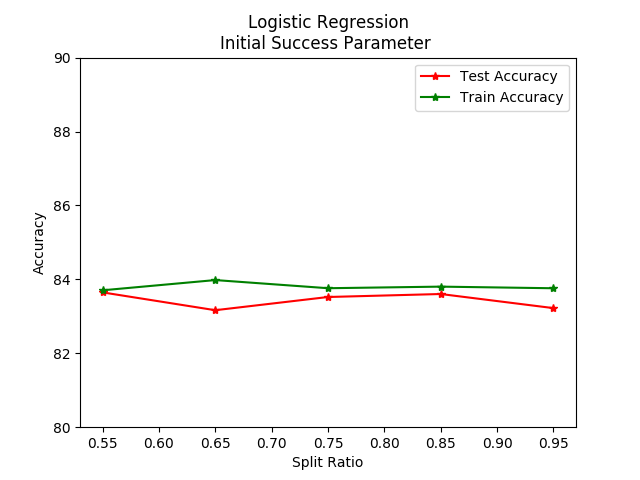
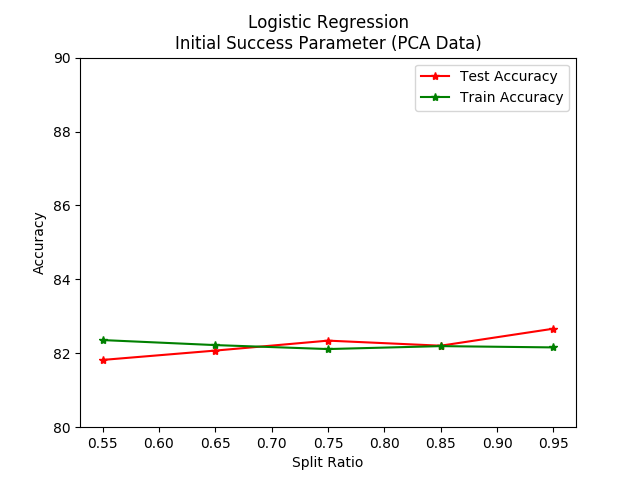


Fig 23 : Accuracy with complete feature data Fig 24 : Accuracy with PCA feature data

**3.3.2 Reformulated Success Parameter**

In the following binary classification the formulated parameters used for success classification is the restaurant rating greater than 3.5 stars and the following to the left is data after PCA and to the right is data without PCA. We clearly have the accuracy at 98% on average for different split ratios after dimensionality reduction compared to data before dimensionality reduction. These having high accuracy may miss classify data as the classification parameters are not as appropriate for performing logistic regression.

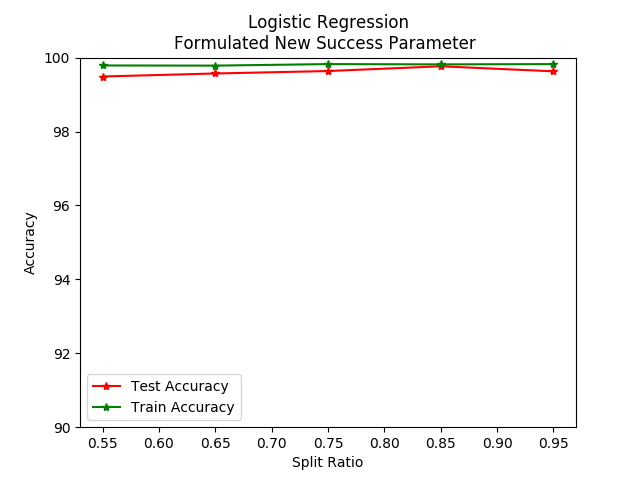
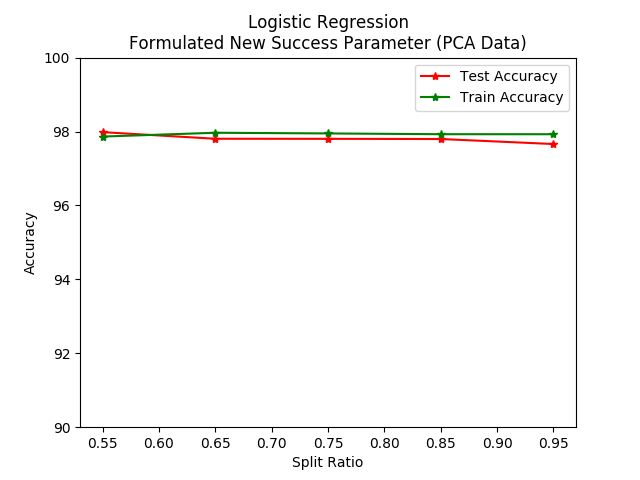


Fig 25 : Accuracy with complete feature data Fig 26 : Accuracy with PCA feature data

With multinomial classification we have classified the success of a restaurant as highly successful, successful, average, poor and very poor. This would give us a better result of the having success. Figure below represents an average of 85% for data after dimensionality reduction compared to data without dimensionality reduction that has an average of 87.5% accuracy for different split ratios.

We could conclude that data with multinomial classification has a better chance of predicting the success.

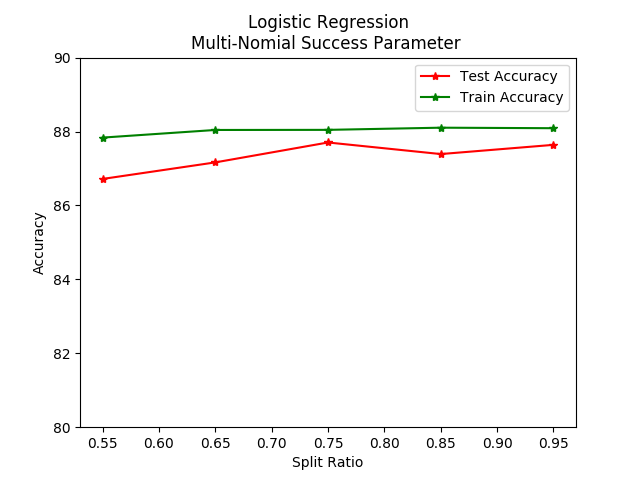
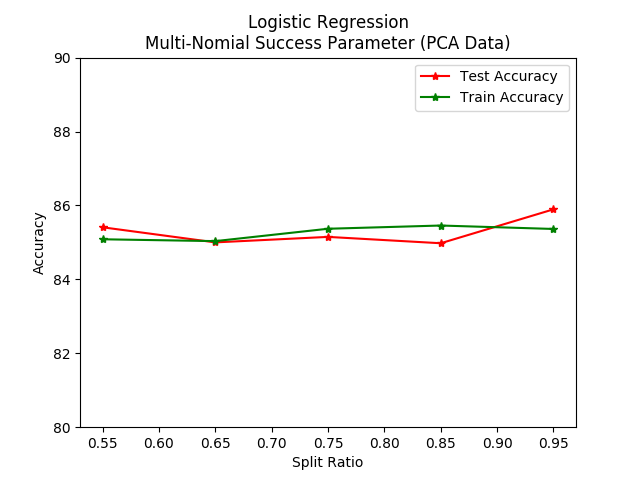


Fig 27 : Accuracy with complete feature data Fig 28 : Accuracy with PCA feature data

As the logistic regression above is based on liblinear solver we have also classified this data sets for other solvers such as 'newton-cg', 'lbfgs', 'sag', 'saga' and compared it with our default solver of liblinear. The split ratio of train vs test ratio was taken as 0.67 and 0.33 respectively.

The figures below represent the accuracy for different solvers for data without PCA to the left and with PCA to the right.

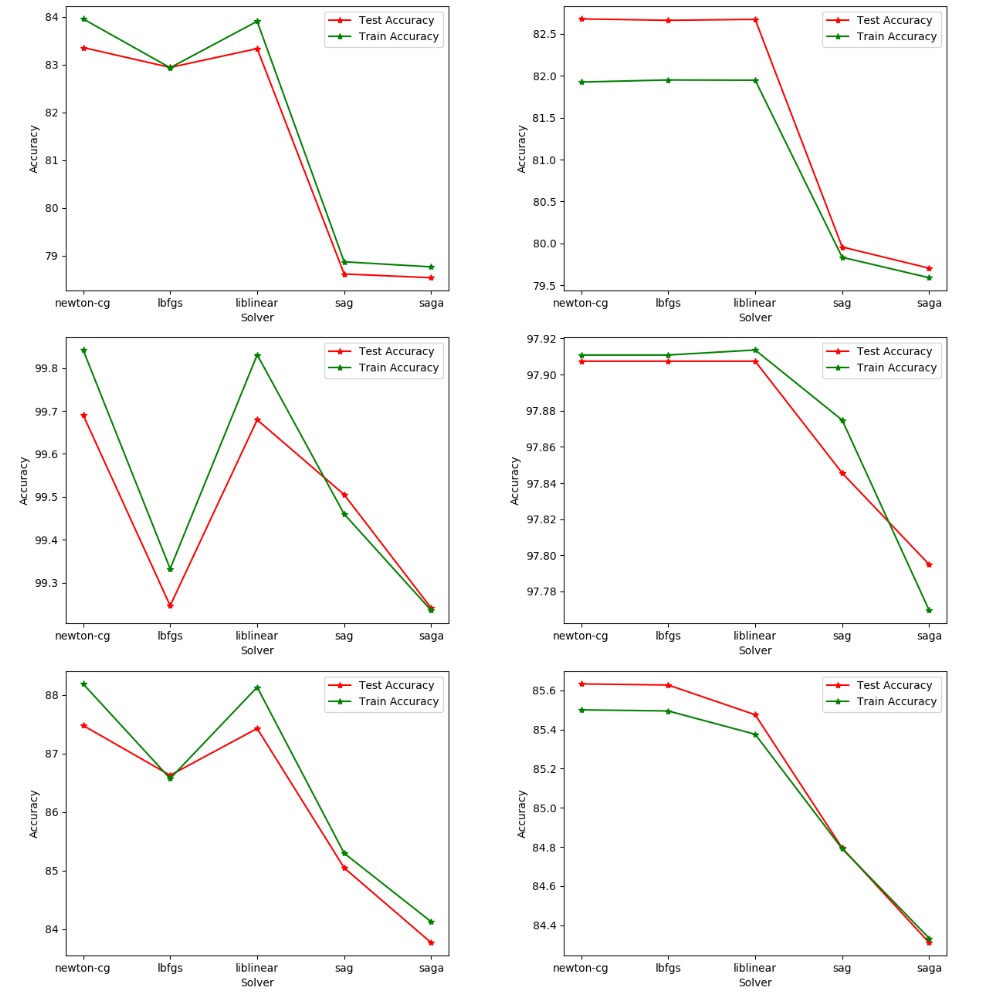


Fig 29 : Without PCA to the left and with PCA to the right on Binary Classification, Formulated

Parameter, Multinomial Classification from top to bottom respectively

**3.4 Random Forest**

Random Forest is an extension of Decision Tree. In Random Forest, during training time we create multiple decision trees during training time, and predict the final class label as the mode of predicted classes given by different trees of the forest.

We create the model using different tree depth level, and achieved maximum accuracy with maximum possible depth.

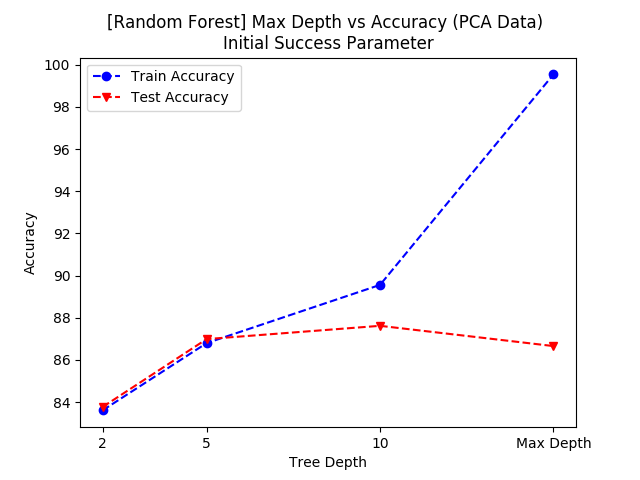
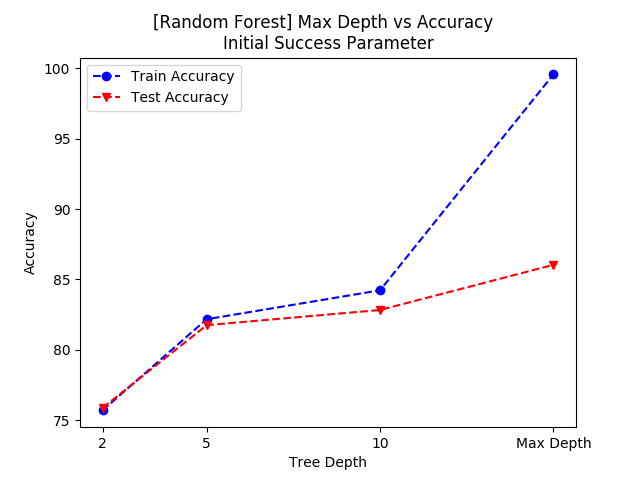


Fig 30 : Accuracy with complete feature data Fig 31 : Accuracy with PCA feature data

**3.4.1 Initial Parameter**

1. **Binary Classification**

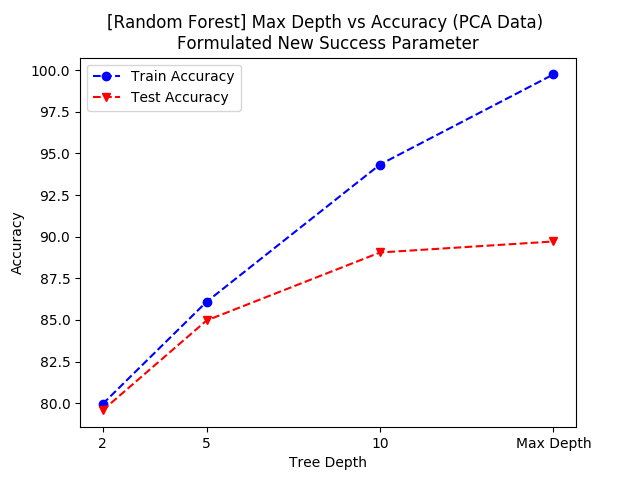
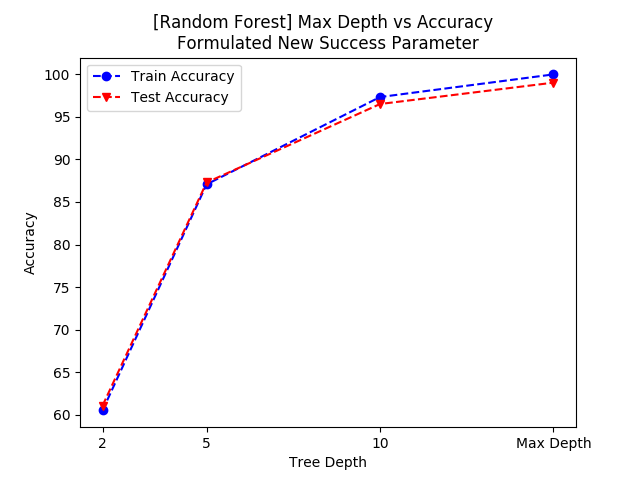


Fig 32 : Accuracy with complete feature data Fig 33 : Accuracy with PCA feature data

1. **Multinomial classification**

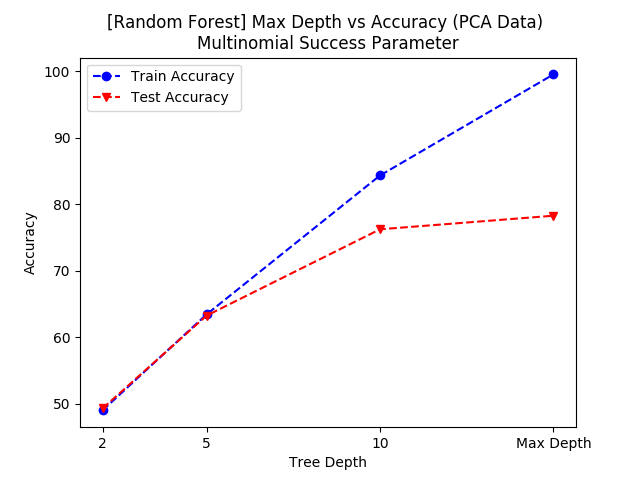
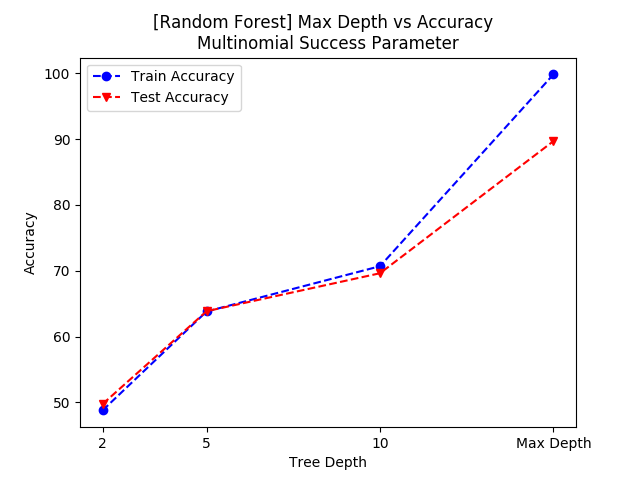


Fig 34 : Accuracy with complete feature data Fig 35 : Accuracy with PCA feature data

**3.5 Naïve Bayes**

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of independence between every pair of features. Given a class variable y and a dependent feature vector x_1through x_n, Bayes’ theorem states a relationship between then.

Naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. Thus using this classifier for our data would help in success prediction.

**3.5.1 Initial Parameter**

For binary data classification in Naive Bayes we have implemented it on two different initial data sets with to the left and without PCA to the right. The data with PCA had better results as naive bayes works well for smaller feature set with the assumption of being independent.

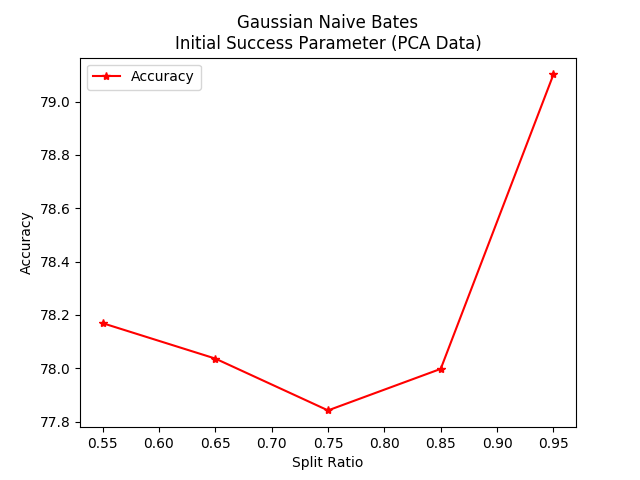
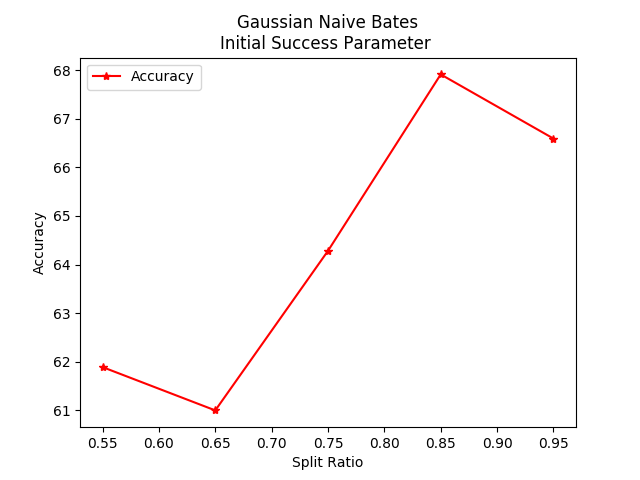


Fig 36 : Accuracy with complete feature data Fig 37 : Accuracy with PCA feature data

With a change in the classification parameter of success for restaurants with a rating of more than 3.5 star rating the values have a gradual increase on smaller data to the right that has its features reduced. with an average of 91% this classification using naive bayes has the highest prediction for the given success parameters in Naive Bayes.

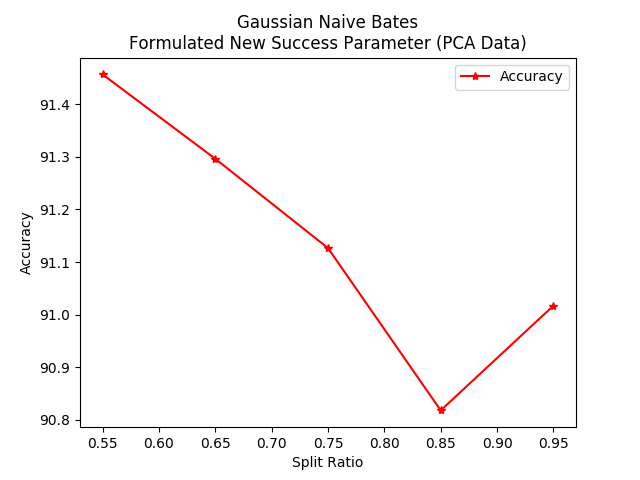
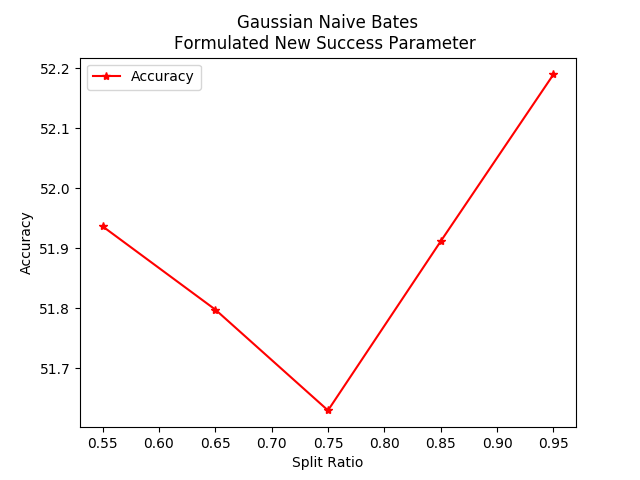


Fig 38 : Accuracy with complete feature data Fig 39 : Accuracy with PCA feature data

1. Multinomial classification

Here we could clearly look at the multinomial classification of naive bayes as the data set has increased gradually.

On the flip side, although naive Bayes is known as a decent classifier, it is known to be a bad estimator, so the probability outputs from the classifier are not to be taken too seriously. But the accuracy on average is around 68.35% to the right on data with PCA.

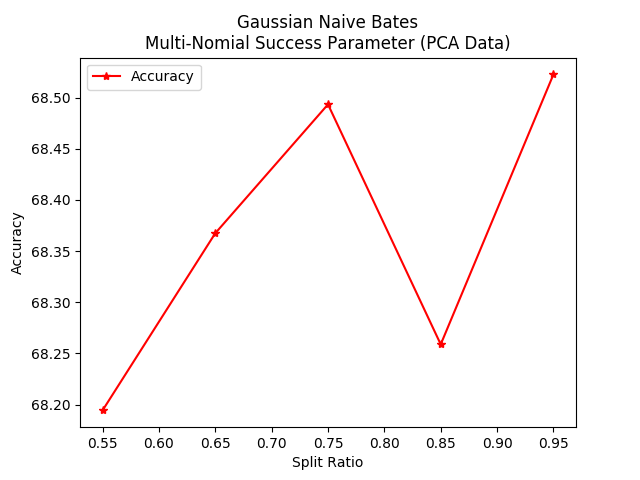
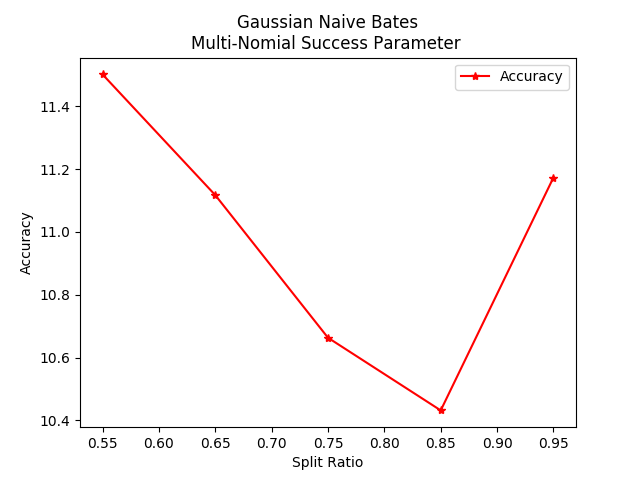


Fig 40 : Accuracy with complete feature data Fig 41 : Accuracy with PCA feature data

**4. Results**

The initial accuracy that we achieve without applying PCA and sentimental analysis is 65% using **Neural Networks.** We are able to reduce the number of parameters from 775 to 22 after applying PCA and applying different classifiers gives us accuracy of around **85%.** Formulating a different success parameter by utilizing user ratings and reviews helped us boost our accuracy to **95%**

**5. Conclusions**

Contrary to the other research papers we are leveraging all the features from the dataset in order to capture the complex relationships between them. This, along with PCA, helped us in getting higher initial accuracy of 85%. We were able to boost the accuracy to **95%** by formulating a new success parameter.

While implementing different machine learning algorithms, we figured out following conclusions:

1. It is computationally expensive for SVM to process 770 features. Hence we had to apply PCA to reduce the number of features that helped us processing SVM smoothly.
2. Naive Bayes ended up having the least accuracy among all the classification models that we worked with. Naive Bayes is not able to capture the complex relationship between the features as it considers each feature as an independent and identically distributed variable.
3. Neural Network ended up with the highest accuracy of **97.93 percent.**

**6. Future work**

**GeoSpatial Data Analysis**

The dataset from yelp has the location attributes of the restaurants, GPS co-ordinates, Address, City and Zip. We performed an exploratory analysis using modified K nearest neighbors algorithm to see how the success rate of surrounding restaurants affect the success of the new restaurant. This can also be used as a metric to analyze the success rate of a location.

KNN with k value of 20 has been modified to include a distance constraint that it should not exceed 1 mile. A Python library geopandas has been utilized for achieving this.

Fig 1 , shows the distribution of restaurants from dataset around the world

Once we had the GPS coordinates of all the restaurants in a data frame, we performed a group by zip on the data-frame to optimize the nearest point calculation time. We got a ‘geo\_score’ by dividing the count of nearest successful restaurants from the count of total nearest restaurants. To convert ‘geo\_score’ in to a binary class , we put 1 for geo\_score > 0.5, else 0

Result, We have found that if 50% of the surrounding restaurants are successful then 78% of the new restaurants will be successful. We haven't used any other attributes other than location, Future work could include seeing how ‘cuisine’ and other attributes affects the success rate.

**Business Recommendation System**

We plan to build a recommendation system which takes location and cuisine as input and provides a success prediction based on the existing data supported by GeoSpatial analysis.

References :

<https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods/>

<https://towardsdatascience.com/dimensionality-reduction-does-pca-really-improve-classification-outcome-6e9ba21f0a32>

<http://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics>

#To check : Related Work