# Yelp Dataset Challenge - Review Rating Prediction

Saksham Gupta
Virginia Tech.
saksham@vt.edu

Pronnoy Goswami
Virginia Tech.
pronnoygoswami@vt.edu

## ABSTRACT

In today's data-driven world reviews play a significant role in impacting various business, products and services. Our preferences while buying products online or at a store are involuntarily affected by the reviews of that product. The business and revenue of all major e-commerce, travel, and restaurant business are largely impacted by the reviews their products or services receive. Yelp is also a platform that allows users to post reviews about various restaurants where they had an opportunity to visit. The review is a free-form text whereas the star rating is out of 5 stars. In this project, we present a systematic methodology to predict the rating given a review. We treat the the rating prediction task as a multi-class classification problem. In this project we explore the Yelp Dataset, different feature selection techniques and possible machine learning methodologies which can be used to tackle this multi-class classification problem. Finally, we provide a detailed comparison of the between the combinations of feature extraction and prediction methodology. We also provide an online tool for testing our models.

## KEYWORDS

Data Mining, Data pre-processing, Machine Learning, Model Evaluation

## 1 INTRODUCTION

User reviews act as 'word-of-mouth' [1] for users to choose between products and services. A lot of websites like Amazon, Walmart, Yelp, TripAdvisor are a major repository of such reviews where user can post their opinions about the businesses and their products. Reviews consists of free-form text and a numeric star rating, usually between 1 to 5. A lot of studies [2] indicate that the reviews have a significant impact on in altering our purchase decisions as well as on the sale of the specific product.

Yelp [3] is an online public sourced review forum for local businesses. Founded in 2004, Yelp currently has more than 4.5 million crowd-sourced reviews. In the growing world of connectivity Yelp acts as platform to help new users by providing peer information about businesses and local markets. The huge number of reviews makes it impossible for users to read every review and decide whether or not to visit the specific restaurant. So, generally users look at the star rating of the particular restaurant and ignore the text. Although it seems that the relationship between review text and the ratings is trivial but it is not. It is very difficult to comprehend based on what factors did a particular user give a particular review - were the food and ambience more important to him/her than the service and average waiting time. Hence, predicting the rating of a restaurant based on the user review is a difficult problem to solve.

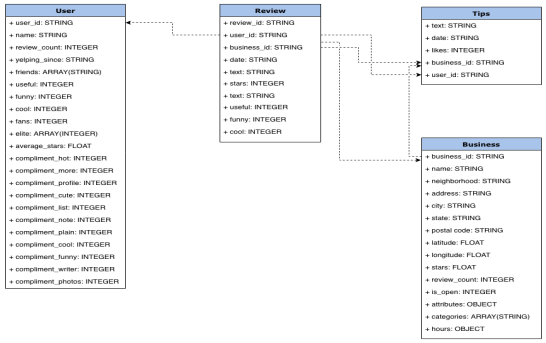



**Figure 1: Yelp Dataset Schema: Depicts the attributes provided and the relationship between different tables**

Yelp annually holds a challenge where in it provides its text review and photo dataset with the aim of getting some meaningful insights and models from the research community. One of the use-case of learning is Sentiment Analysis, wherein given some review text the learned model tries to predict whether the user finds the business good or bad.

## 2 DATA DESCRIPTION

Yelp Dataset [3] consists of 5,996,996 text reviews about 188,593 businesses containing 280,992 pictures belonging to 10 metropolitan areas. For our project we use the reviews, business and user information to solve sentiment analysis and rating prediction problems. In reviews we kept the unique id of each review, user, business, text and stars, while for user we kept user ID, Name and review count. For Business we kept business ID, Name and address. Fig:1 shows a detailed relational view of the dataset, describing the type of attributes provided and their relations. For sentimental analysis we convert the star ratings from 1-2.5 stars as negative and 2.6-5 stars as positive reviews. For multi-class rating prediction we take round the ratings to form 5 classes ranging from 1 to 5 stars discretely.

## 3 DATA PRE-PROCESSING

For our problem we focused on businesses in United States only, so we removed all the non-US businesses from the dataset. Due to this we had to remove reviews belonging to removed businesses from the review dataset. We also cleaned out those reviews which did not relate to any business in the dataset. We had some concerns regarding the fake reviews present in the dataset, but for now we didn't clean them because that would require to resorting to unofficial dataset sources leading to false results.

**Tokenization:** Tokenization describes the general process of breaking down a text corpus into individual elements that serve as input for various natural language processing algorithms. A token is an instance of a sequence of characters which are grouped

together as a useful semantic unit for further processing. Figure 2 illustrates the process of tokenization on a specific sentence.

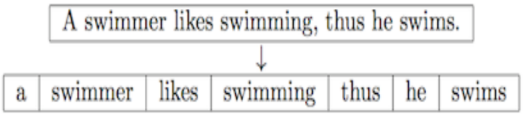



**Figure 2: Tokenization**

**Stop Words:** Stop words are words that are particularly common in a text corpus and thus considered as rather un-informative (e.g., words such as so, and, or, the). One approach to stop word removal is to search against a language-specific stop word dictionary. An alternative approach is to create a stop list by sorting all words in the entire text corpus by frequency. The stop list after conversion into a set of non-redundant words is then used to remove all those words from the input documents that are ranked among the top $n$ words in this stop list. Using a stop list significantly reduces the number of postings that a system has to store.

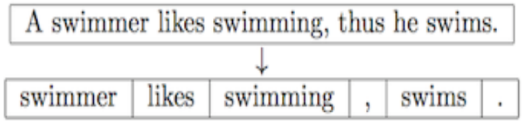



**Figure 3: Stop Words Removal**

**Stemming and Lemmatization** Stemming[4] describes the process of transforming a word into its root form. The original stemming algorithm was developed my Martin F. Porter in 1979 and is hence known as Porter Stemmer.

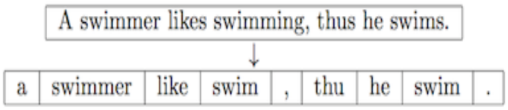



**Figure 4: Stemming example**

Sometimes stemming can create non-real words because it is a crude heuristic process that chops off end of words to obtain the word root. In contrast to stemming, lemmatization aims to obtain the canonical (grammatically correct) forms of the words with the use of a vocabulary and morphological analysis of words, the so-called *lemmas*. Lemmatization is computationally more difficult and expensive than stemming.

## 4 DATA EXPLORATION

Fig.8 depicts the ratings aggregated on different US States. From this we can observe that the distribution of ratings is similar to the overall distribution of ratings in US depicted by Fig.7.

We also observed that the highest variance restaurants categories(Fig.9) are the fast-food, pizza, sandwiches, burgers, and hot dogs restaurants. On the other hand, the lowest variance restaurant categories(Fig.10) are buffet, breakfast and brunch, steakhouses and Thai restaurants.

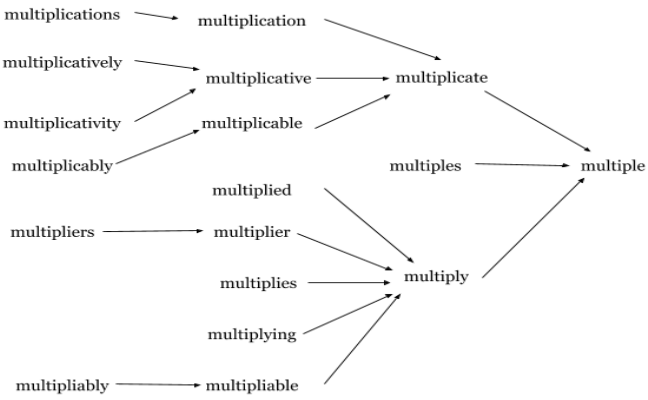



**Figure 5: Lemmatization**

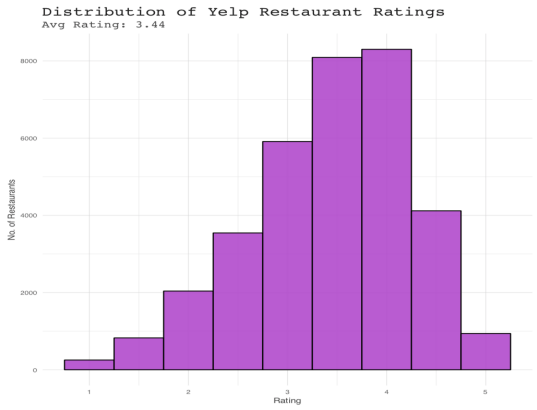



**Figure 6: Overall Distribution of Yelp Restaurant Ratings**


**Figure 7: A collection of the highest frequency words occurring in the reviews**

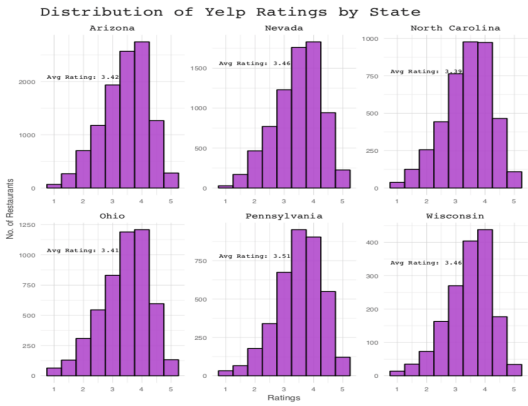



**Figure 8: State-wise Distribution of Yelp Restaurant Ratings**

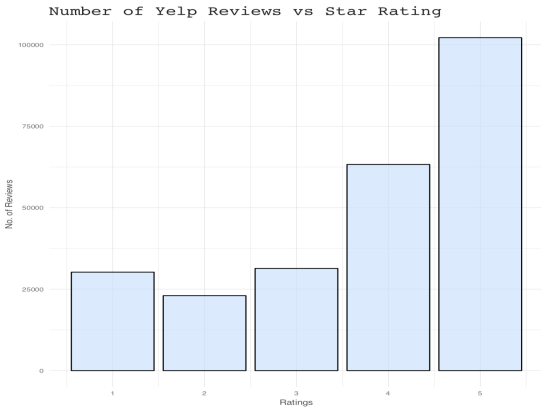



**Figure 11: Number of Yelp Reviews vs Star Ratings**

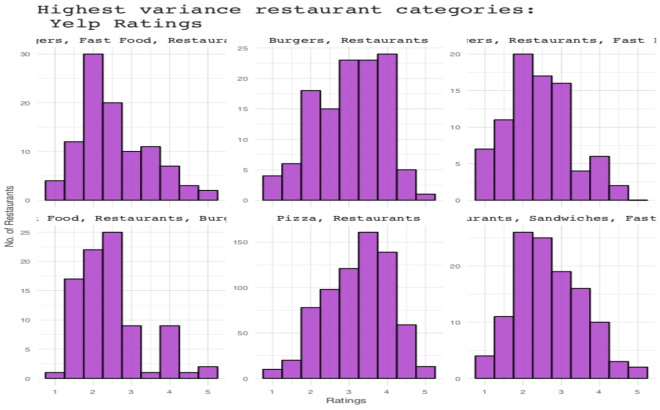



**Figure 9: Distribution of Yelp Ratings for highest Variance restaurant categories**

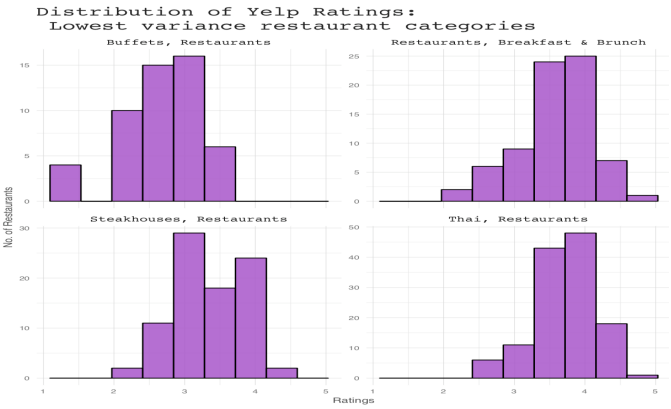



**Figure 10: Distribution of Yelp Ratings for lowest Variance restaurant categories**

We also analyzed the types of ratings that people leave for restaurants at Yelp and found that the number of good reviews were greater than the number of bad reviews. It implies that people are more likely to leave good reviews on Yelp than bad reviews. By this analysis, we can have an initial estimate of the distribution of classes (in this case ratings) of our dataset.

# 5 MODEL BUILDING

## 5.1 Feature Selection

When dealing with high volume text data we cannot directly apply machine learning algorithms as it will complicate the model, add redundancy and might lead to learning of low variance and dependent attributes. Thus feature selection and reduction is one of the most important step in any data analytics algorithm. Here we use five approaches namely Unigram, Unigram and Bigram, Unigram, Bigram and Trigrams, Latent Semantic Indexing(LSI) and Word2Vec mostly commonly used for feature selection in the field of Natural Language Processing. Instead of applying all the feature selection procedure with each of the learning approaches blindly, we combine only those feature selection techniques which are know to work well with a given learning methodology.

*5.1.1 Unigram.* In unigram model we consider each word appearing in the review corpus as a feature just like a bag of words model[5]. To construct a feature vector for a review we first create a corpus out of all the words occurring in all of the reviews. Then a matrix is created containing the frequency of the word. Finally we multiply these features with their TF-IDF Term Frequency and Inverse Document Frequency to give more weight-age to word that appear less frequently in the reviews and act as a good distinguisher in comparison to words that occur very frequently and does not help in identifying the class.

*5.1.2 Unigram and Bigram.* One of the disadvantage of only using unigram as features in natural language processing is the presence of word relations like "not good" in the reviews which are not captured by the unigram approach as it uses singular words in an isolated fashion. To cover such relations we extend the unigram approach by taking into consideration bigram features as well. The corpus is now appended with both unigram and bigram features with their respective TF-IDF scores matrix for feature selection.

*5.1.3 Unigram, Bigram and Trigram.* After addition of Unigram and Bigrams there are still some phrases like "really good taco" which are not captured by them. [6] To include such high TF-IDF phrases we incorporate the Trigram features as well in the corpus. However, same Trigram features rarely occur across different reviews as multiple users are unlikely to use same three words phrase in their review. Thus we don't expect the results of this technique to be very different from the Unigram and Bigram features.

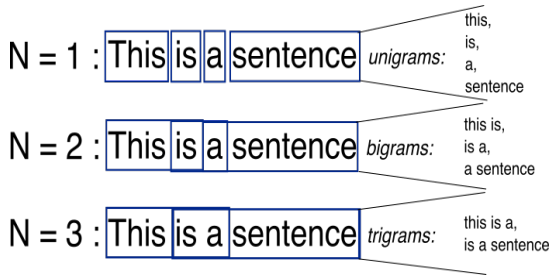



**Figure 12: A brief working of Unigrams, Bigrams and Trigrams**

*5.1.4 Latent Semantic Indexing.* One of the major problem in Natural Language processing is the presence of large number of features extracted from text which leads to complex models and high computation requirements, ultimately leading to low accuracy and poor results. [7] LSI match topics instead of exact words i.e. words occurring in similar context. We utilize the matrix generated by unigram[8] and apply Singular Value Decomposition(SVD)[9] to yield three matrices USV out of which V forms our final feature matrix.

We construct a word-review matrix $D$, of dimensions of the unigram($mxn$) corpus. We then perform the Singular Value Decomposition on $D$.

$$SVD(D) = U \cdot S \cdot V^T$$

Here,

$$U = word\ topic\ matrix\ of\ size\ mxm$$

$$S = Diagonal\ matrix\ of\ size\ mxn\ having\ n\ singular\ values$$

$$V = Transpose\ of\ topic\ review\ matrix\ of\ size\ nxn$$

The columns of $S$ corresponds to the topics in the reviews, the $i'th$ singular value is the measure of the importance of the $i'th$ topic. From these topics we choose the first $n*$ topics that represent the most important ones. Further we choose these $n*$ features from the V matrix. The value of $n*$ can be determined using the scree plot.

*5.1.5 Word2Vec.* The goal of word vector embedding models, or word vector models for short, is to learn dense, numerical vector representations for each term in a corpus vocabulary. [10]If the model is successful, the vectors it learns about each term should encode some information about the meaning or concept the term represents, and the relationship between it and other terms in the vocabulary. Word vector models are also fully unsupervised, they learn all of these meanings and relationships solely by analyzing the text of the corpus, without any advance knowledge provided.

At the start of the learning process, the model initializes random vectors for all terms in the corpus vocabulary. The model then slides the window across every snippet of text in the corpus, with each word taking turns as the focus word. Each time the model considers a new snippet, it tries to learn some information about the focus word based on the surrounding context, and it "nudges" the words' vector representations accordingly. One complete pass sliding the window across all of the corpus text is known as a training epoch. It's common to train a word2vec model for multiple passes/epochs over the corpus. Over time, the model rearranges the terms' vector representations such that terms that frequently appear in similar contexts have vector representations that are close to each other in vector space. A basic representation of word2vec model is provided in Fig.13.

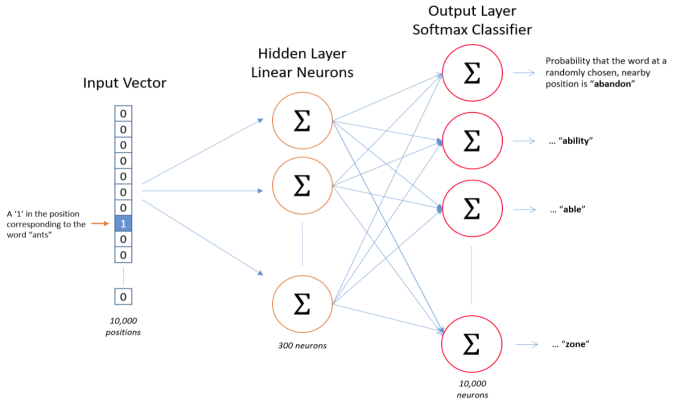



**Figure 13: An overview of word2vec working**

## 5.2 Learning

*5.2.1 Naïve Bayes Classification.*
The Naïve Bayes Classifier makes the assumption that given a class the conditional probability between any two features is independent of each other. [11] We calculate the posterior probability given a feature using this assumption and build the model. Then, when a new feature is encountered we compute all the joint probability values of the class for that feature and the highest probability is the output as the final class label for this new feature.

In this report we use multinomial Naïve Bayes Classifier, which assumes the Probability of some review $i$ given some rating is a multinomial distribution of all reviews the in the database. Multinomial Naïve Bayes is generally used in document classification as it works well for data that can be converted into frequencies, like the count of the words in the text.

*5.2.2 Logistic Regression.* In Logistic Regression we try to model the conditional probability function $P(s|r)$, where $r$ is the feature vector for the review and $s$ is the corresponding class label. [12]Once this probability is learned by the Regression model, it is able to predict the class label given some new unseen review in the feature space. Logistic Regression Models are know to work well with linear data which is certainly the case here given that the feature selection technique can represent the reviews in a linear fashion.

*5.2.3 Support Vector Machine(SVM).* An SVM model is a representation of the examples as points in space, mapped so that the

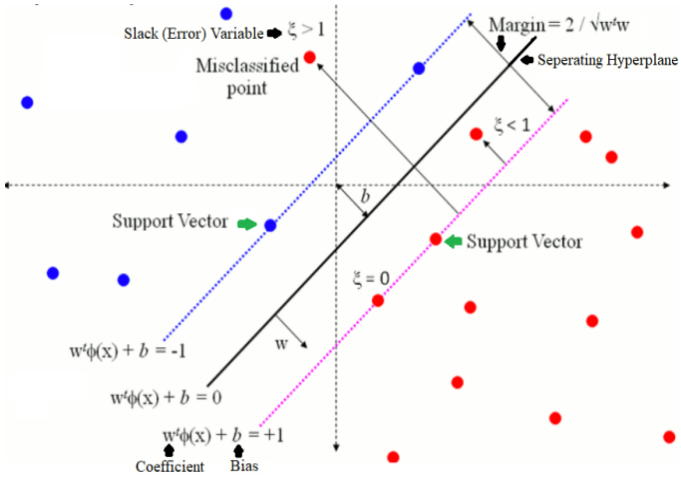



**Figure 14: Basic working of Linear SVM on binary class classification problem**

examples of the separate categories are divided by a clear gap that is as wide as possible. [13] New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. For our approach we will be using linear SVMs. SVMs are defined by hyperparameters like regularization(C) and Gamma. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassified more points.The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. In other words, with low gamma, points far away from plausible separation line are considered in calculation for the separation line. Where as high gamma means the points close to plausible line are considered in calculation. Fig.14 demonstrates the process of separating the data using hyperplane in a simplified fashion.

*5.2.4 Embedding Module and LSTM(RNN + LSTM).* Since reviews are sequential data and order of words encode lot of useful information, we can use Recurrent Neural Network(RNN's) with Long-Short Term Memory Units(LSTM's) to learn these sequential patterns in the data. [14] We can design a network that can learn some low level embeddings which can be passed through LSTM units to add recurrent information about the sequence of words in the reviews. For better understanding a simplified depiction is given in Fig.16. The embeddings are similarly learned as word2vec and produce similar results. In our method we use Glove (Global Vectors for Word Representation) [15] embeddings which is trained on large datasets of words crawls on the internet. We use the Twitter 27 Billion tokens embedding's to form the embedding layer before the LSTM Network.

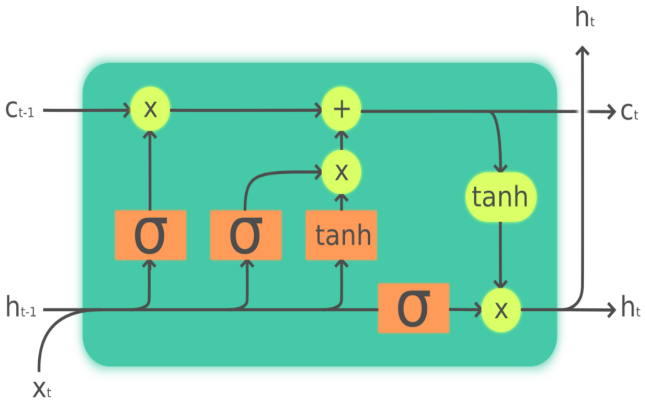



**Figure 15: An LSTM unit, here the $h_t$ is the hidden state vector that can be maintained by the network while the $c_t$ is cell state vector and $x_t$ are the word embeddings.**

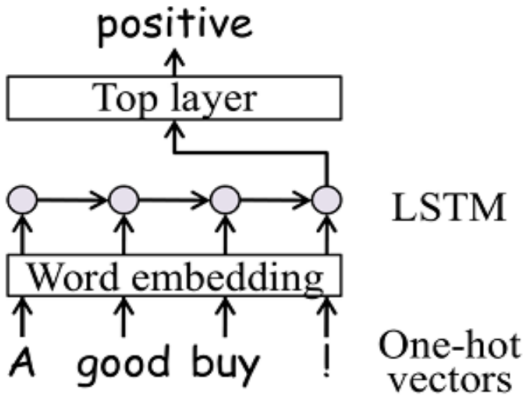



**Figure 16: Recurrent neural networks with LSTM units and word embeddings**

## 6 RESULTS AND ANALYSIS

### 6.1 Implementation Details

For experiment purposes we split the review data as 80% training and 20% test data. We then perform a 5-fold cross validation on the training set while minimizing the Root Mean Square Error(RMSE) on each training and validation fold.

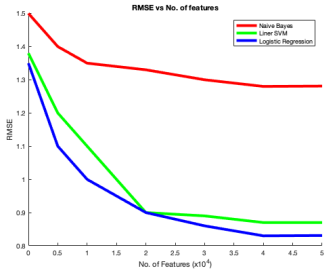
We used Virginia Tech's high performance compute systems(ARC) with 2 Nvidia P100 GPU's with 12 GB RAM each for training and testing the approaches. All our code is written in Python with the help of Scikit Learn and Pandas Library. All the code is provided in the interactive Ipython notebooks for result replication.

### 6.2 Results and Experiments

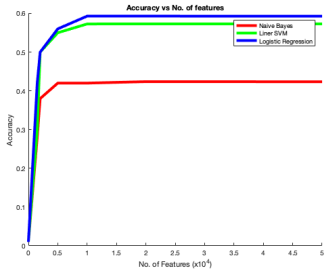
*6.2.1 Unigrams.* Fig. 17 depicts the performance of the three classifiers on unigram features. The RMSE suddenly decrease as number of features are increased as expected, but they level off around 50,000 unigram features with an exception of Naive Bayes whose RMSE starts marginally after 50,000 features.

| Feature Selection | Unigrams | Unigrams | Unigrams | Uni+Big | Uni+Big+Tri | LSI | LSI | word2vec | word2vec | embedding |
|---|---|---|---|---|---|---|---|---|---|---|
| Learning | Naive Bayes | SVM | Log. Reg. | Log.Reg. | Log. Reg. | SVM | Log. Reg. | SVM | Log. Reg. | LSTM |
| RMSE | 1.28 | 0.87 | 0.83 | 0.78 | 0.778 | 0.723 | 0.63 | 0.60 | 0.587 | 0.516 |
| Accuracy(%) | 42.38 | 57.29 | 59.30 | 64 | 64.2 | 68.93 | 69.62 | 71.28 | 73.51 | **78.62** |

**Table 1: This table represents the accuracy and Root Mean Square Error we get on the Yelp Test Dataset. As Logistic Regression perform best with Unigram features we continue to use it with Bigram and Trigrams as well. The highest accuracy is achieved by the word embedding and LSTM model.**

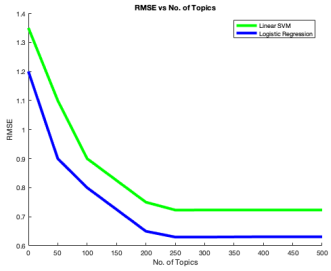



**(a) Plot of RMSE against the number of unigram features used for training and classification**

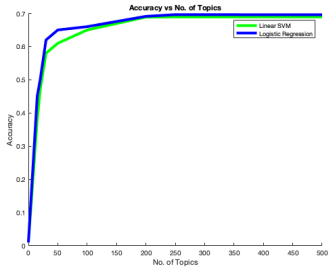



**(b) Plot of Accuracy against the number of unigram features used for training and classification**

**Figure 17: RMSE and Accuracy curves against the number of unigram features taken for learning on Naive Bayes, SVM and Logistic Regression**




**(a) Plot of RMSE against the number of topics used for training and classification**




**(b) Plot of Accuracy against the number of topics used for training and classification**

**Figure 18: RMSE and Accuracy curves against the number of topics taken for learning on SVM and Logistic Regression**

We can clearly infer that Logistic Regression performs best out of the three classifiers acheiving an accuracy **59.30%** with an RMSE of 0.83, while Support Vector Machine(SVM) closely follows Logistic Regression with an accuracy of **57.29%** and RMSE of 0.87. Due to brevity of space and computation we only apply Logistic Regression on the unigram, bigram and Trigram features.

*6.2.2 Unigrams and Bigrams.* As expected we see considerable increase in accuracy of Logistic Regression to **64%** and reduction in RMSE to 0.78 when using the Unigram and Bigram features as well. This indicates that Bigrams occur frequently enough in the corpus and capture a lot of information that are not taken into account by the Unigram features alone.

*6.2.3 Unigrams, Bigrams and Trigrams.* The results for this feature extraction method are al- most exactly the same as those for the 'Unigrams Bigrams' method with slight increase in accuracy. The best RMSE and accuracy scores are 0.778 and **64.2%**, achieved by logistic regression.

Adding trigrams to the previous model does not help, because trigrams repeat rarely. It is unlikely that two different user would use the exact same 3-tuples to describe a restaurant. The TF-IDF weighting technique weights almost all the 3-tuples as very rare, therefore they are not very useful as features.

*6.2.4 Latent Semantic Indexing.* While experimenting with LSI we found that the top 250 topics contains the most information and thus can be used as an input to SVM and Logistic Regression classifiers. Fig.18 depicts the RMSE and accuracy measures with change in the number of topics taken for training and testing.

From the Fig.18 we infer that Logistic Regression performs the best with accuracy of **69.62%** and RMSE of 0.63. The plots for logistic regression show another interesting pattern. As the number of features increases, the RMSE remains constant around 0.82, but the

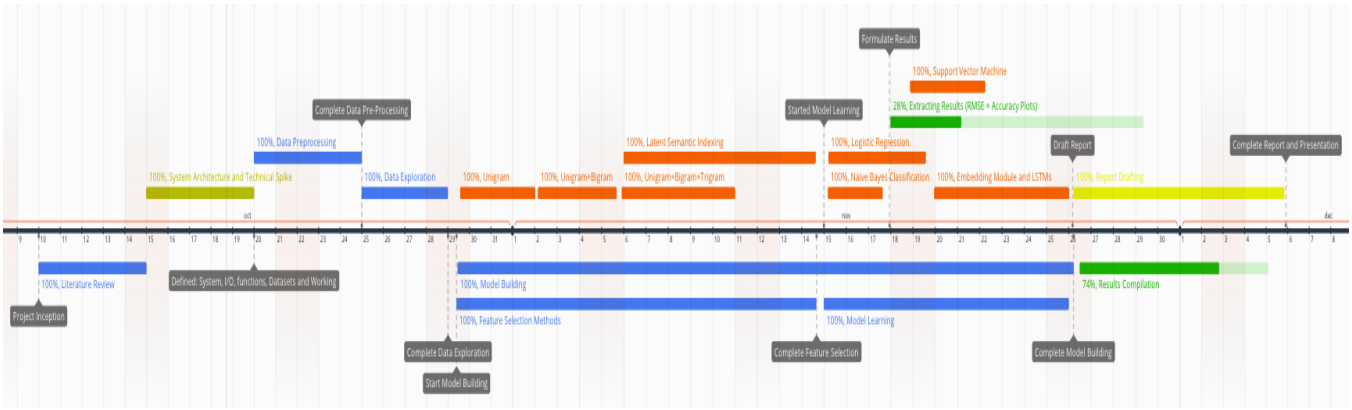



**Figure 19: Project Timeline**

accuracy increases from 0.65 to 0.69. One explanation for this apparent anomaly is that, more examples get classified accurately, but the RMSE for the misclassified examples increases and overshadows the decrease in the RMSE due to better accuracy.

*6.2.5 Word2Vec.* From Table. 1, we can infer that word2vec embeddings performs well as features to both SVMs and Logistic Regression learning technique. We can also see that Logistic Regression continues to perform well with accuracy of **73.51%** and RMSE of 0.587, while again SVM follows closely with accuracy of **71.28%** and RMSE of 0.60. This trend is apparent for all of the previous feature creation and selection techniques used.

The unsupervised topic or context learning of word2vec seems to form better features for both SVM and Logistic Regression learning techniques.

*6.2.6 Embeddings and LSTM.* For embeddings we used Glove's word embeddings which is trained on 27 Billion Tokens taken directly from the Twitter feed. We use these embeddings as they contain most of the frequently occurring english words mapped to correct context which thus forms more robust features when fed into the LSTM network.

The Recurrent nature of the LSTM's certainly seems to improve the accuracy a lot, outperforming all the previous techniques with an accuracy of **78.62%** and RMSE of 0.516 which is arguably quite good on 5 class text classification problem with high number of features. Thus we can infer that the methodology of considering text as a sequential data is along the right lines and can be extended with the help of convolutional LSTMs or GRUs(Gated Recurrent Units).

## 7 CONCLUSION

In this report, we tackle the problem of Review Rating Prediction for restaurants on Yelp. We divide the problem as a 5 class classification problem. We apply various feature extraction and selection techniques with various learning methodologies to find how they well each of them leverage their learning power to learn the trends in the review data.

From the results and analysis section we can infer that the right selection of both the feature selection and learning methodology play a crucial role in governing the accuracy and error estimates of the classification problems. Here, we see that conventional approaches like Logistic Regression and SVMs perform well if the initial features given to them are good, while deep learning techniques like LSTMs [16] are really powerful in learning the structure and manifold of the data and producing good results even for multiclass classification problems.

## 8 EXTRA CREDIT

We expanded our approach to build a minimilitic Flask based web app which can be used by restaurant owners to predict what there cumulative restaurant rating will be given a single or multiple reviews. A detailed demo of our approach is described in this video: Demo

## 9 FUTURE WORK

Though the methods tested in this paper are extensive, they are by no means exhaustive. In fact, there are many avenues for improvements and future work, some of which are discussed below:
We can apply more complex techniques like Bidirectional LSTMs and Convolutional LSTMs/GRUs as they can learn more difficult mappings from the data. Further we can improve the features by introducing topic modelling using Latent Dirichlet Allocation(LDA), Non-negative Matrix Factorization and Independent component Analysis. Traditional techniques like collaborative filtering can also be applied to compare how they score against the newer approaches. A good evaluation of our model will be its prediction capability on other business like travel, stores, shops etc. which will tell us the generalization capability of our models.

## 10 RESPONSE TO PROPOSAL COMMENTS

*10.0.1 Timeline.* Fig. 19 showcase a detailed timeline followed by us to develop our project. Here we detail the major milestones during our project and there current status.

*10.0.2 Contributions.* Both team members have equal contribution in developing the whole project. A detailed breakup is given below: Saksham (50%): wrote code for Latent Semantic Indexing, Word2Vec, LSTM and Logistic Regression. Worked on Report and Slides.

Pronnoy (50%): wrote code for Unigrams, Bigrams, Trigrams, Naive Bayes, SVMs and Logistic Regression. Worked on Report and Slides.

## 11 TAKE HOME MESSAGE

From this project we can take home the following points:

- Choosing correct Feature Selection technique is as important as choosing the correct learning technique.
- Conventional Machine Learning techniques perform well on the text provided that the features fed into them are robust and good.
- Newer Deep Learning techniques like LSTMs and Word2Vec display considerable improvements on the conventional techniques, the process of considering text as a sequential feature is certainly a step in right direction as confirmed by our result and analysis.

## REFERENCES


[1] Chrysanthos Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 49:1407–1424, 2003.

[2] Shin-yi; Chen, Pei-Yu; Wu and Jungsun Yoon. "the impact of online recommendations and consumer feedback on sales" (2004). icis 2004 proceedings. 58.

[3] Yelp. Yelp dataset challenge'2018.

[4] B. P. Pande, Pawan Tamta, and H. S. Dhami. Generation, implementation and appraisal of a language independent stemming algorithm. *CoRR*, abs/1312.4824, 2013.

[5] Riadh Bouslimi, Abir Messaoudi, and Jalel Akaichi. Using a bag of words for automatic medical image annotation with a latent semantic. *CoRR*, abs/1306.0178, 2013.

[6] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 697–702, Washington, DC, USA, 2007. IEEE Computer Society.

[7] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '98, pages 159–168, New York, NY, USA, 1998. ACM.

[8] Nabiha Asghar. Yelp dataset challenge: Review rating prediction. *CoRR*, abs/1605.05362, 2016.

[9] V. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176, April 1980.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.

[11] I. Rish. An empirical study of the naive bayes classifier. Technical report, 2001.

[12] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, 2002.

[13] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML'98, pages 137–142, Berlin, Heidelberg, 1998. Springer-Verlag.

[14] Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015.

[15] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, 2014.

[16] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, November 1997.

# YELP RATING PREDICTION CHALLENGE

## CS/STAT 5525 - DATA ANALYTICS
## [Fall 2018]

**Final Project Presentation**


**Submitted By:**

Saksham Gupta [saksham@vt.edu]

Pronnoy Goswami [pronnoygoswami@vt.edu]


VIRGINIA TECH

# YELP RATING PREDICTION CHALLENGE

## PROBLEM FORMULATION


**Given:** Dataset of 5,996,996 text reviews about 188,593 businesses containing 280,992 pictures belonging to 10 metropolitan areas.

**Problem:** For a given customer review predict the rating which the customer will give to a certain business based on the review.

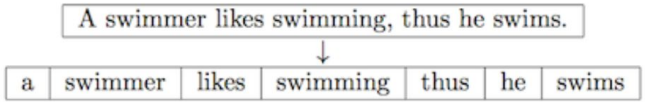
**Task:** Learn the mapping between the text reviews and the associated star rating. This boils down to a 5-class classification problem with each of the star rating {1, 2, 3, 4, 5} forming a class.

# YELP RATING PREDICTION CHALLENGE

## DATA PRE-PROCESSING

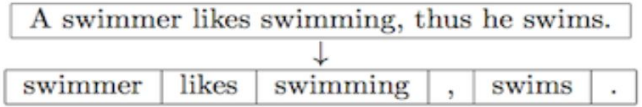
### 1 Tokenization

Breaking down text corpus into individual elements



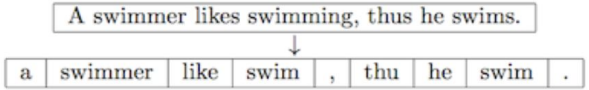

### 2 Stop Words Removal

Removing un-inforamtive words like as, so, the, etc.



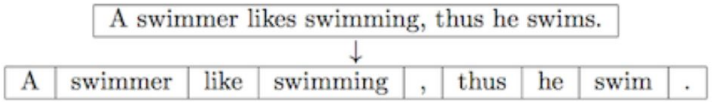

### 3 Stemming

Transforming a word into its root form




### 4 Lemmatization

Obtain grammatically correct form of words by vocabulary and morphological analysis



# YELP RATING PREDICTION CHALLENGE

## DATA EXPLORATION


*Fig: Highest frequency words in reviews*

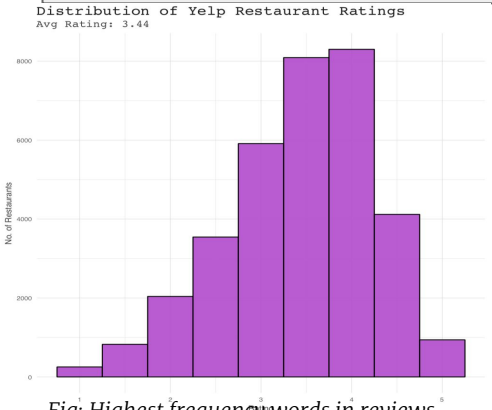



*Fig: Highest frequency words in reviews*

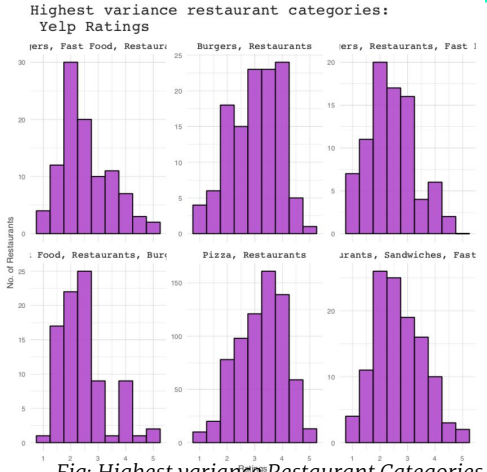



*Fig: Highest variance Restaurant Categories*

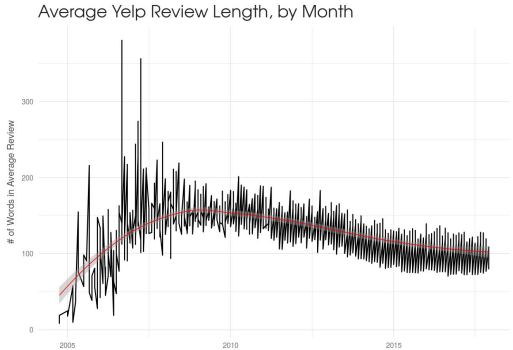



*Fig: Average Review Length by Month*

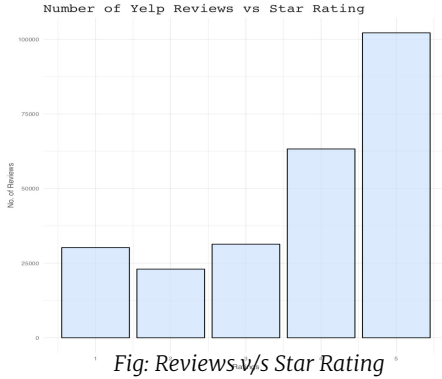



*Fig: Reviews v/s Star Rating*

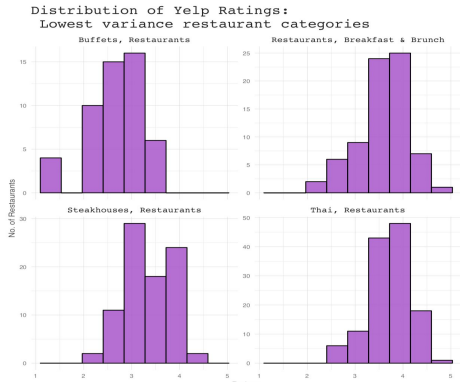



*Fig: Lowest Variance Restaurant Categories*

# YELP RATING PREDICTION CHALLENGE

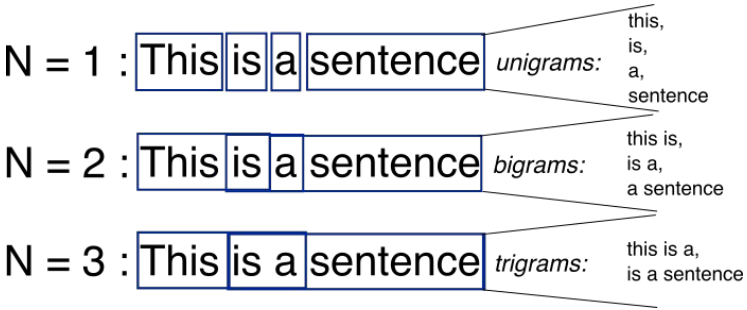
## FEATURE SELECTION

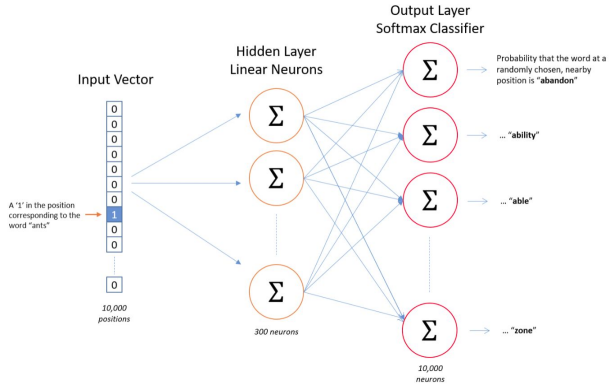
### Unigram, Bigrams and Trigrams

**Unigram:** Each word appearing in the review corpus considered as a feature.

**Bigram:** Each pair of words appearing in the review corpus considered as a feature.

**Trigram:** Each triplet of words appearing in the review corpus considered as a feature.




### Word2Vec

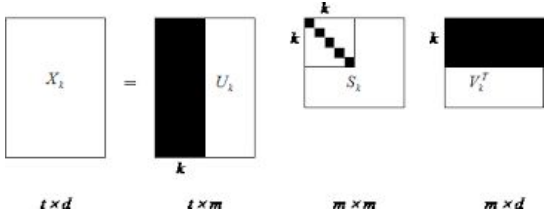
Learn Dense numerical vector representation for each term in a corpus.

Unsupervised approach which yields vector representations such that words that appear in similar context have similar vector representations.

LSI matches topics instead of exact words.

LSI's important topics are extracted by performing Singular Value Decomposition on the word corpus.

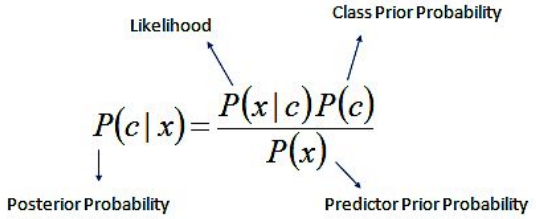
### Latent Semantic Indexing



# YELP RATING PREDICTION CHALLENGE

**LEARNING - I**

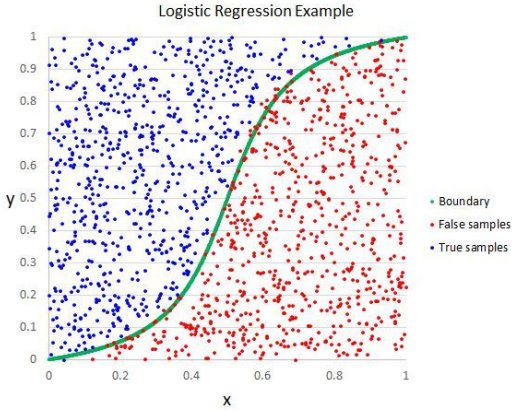
## Naive Bayes




$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

- Makes the assumption that probability between 2 features is independent.
- Using this assumption, build a model to predict the posterior probability.
- Given a new feature we calculate the all the joint the joint probability values and the maximum value is the predicted class label.

## Logistic Regression

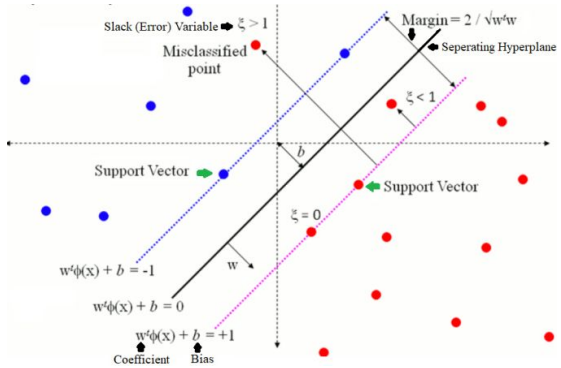



- Attempts to model the conditional probability function, **p(s|r)**, where,
  *r = feature vector for review*
  *s = corresponding class label*
- Post learning the model is able to predict the class label given an unseen feature vector generated from the review

# YELP RATING PREDICTION CHALLENGE

**LEARNING - II**

## Support Vector Machines




A representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

## Recurrent Neural Networks and LSTMs

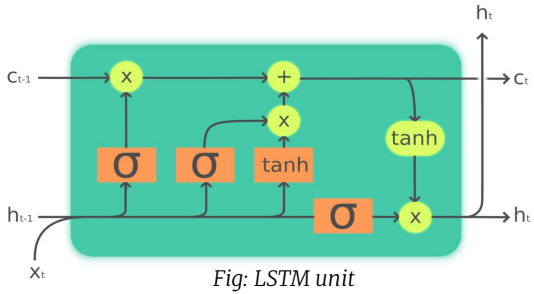



*Fig: LSTM unit*

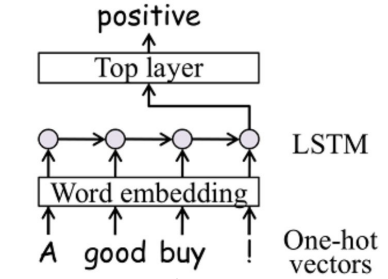



*Fig: LSTM network with word embeddings*

LSTM(Long Short Term Memory Units) learn the sequential patterns in the data.

First the low level embeddings are learned in an unsupervised fashion using an embedding module and then series of LSTM networks are applied to learn the mapping trend in the data.

# YELP RATING PREDICTION CHALLENGE

## RESULT AND ANALYSIS - I

| Feature Selection | Unigrams | Unigrams | Unigrams | Uni+Big | Uni+Big+Tri | LSI | LSI | word2vec | word2vec | embedding |
|---|---|---|---|---|---|---|---|---|---|---|
| Learning | Naive Bayes | SVM | Log. Reg. | Log.Reg. | Log. Reg. | SVM | Log. Reg. | SVM | Log. Reg. | LSTM |
| RMSE | 1.28 | 0.87 | 0.83 | 0.78 | 0.778 | 0.723 | 0.63 | 0.60 | 0.587 | 0.516 |
| Accuracy(%) | 42.38 | 57.29 | 59.30 | 64 | 64.2 | 68.93 | 69.62 | 71.28 | 73.51 | **78.62** |

**Table 1: This table represents the accuracy and Root Mean Square Error we get on the Yelp Test Dataset. As Logistic Regression perform best with Unigram features we continue to use it with Bigram and Trigrams as well. The highest accuracy is achieved by the word embedding and LSTM model.**

- LSTM with embedding modules perform best amongst all the approaches.
- SVM and Logistic Regression follow closely in terms of accuracy.
- Word2Vec is the best feature selection technique.
- With each n-gram features we some improvement with major improvement from Unigrams to Bigrams while marginal improvement from Bigrams to Trigrams.

# YELP RATING PREDICTION CHALLENGE

## RESULT AND ANALYSIS - II

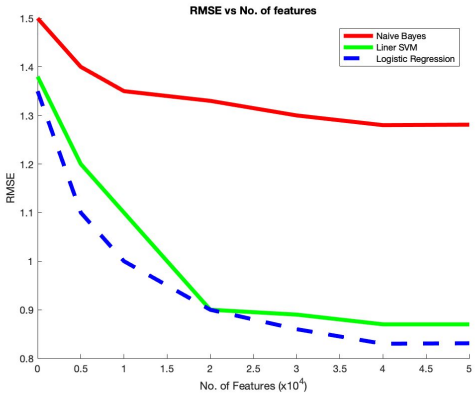



*Fig: Root Mean Squared Error 'vs' No. of features*

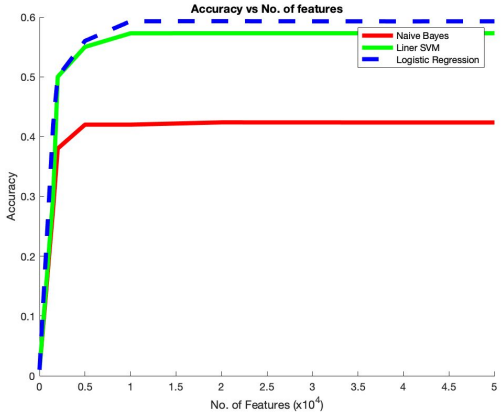



*Fig: Accuracy 'vs' No. of features*

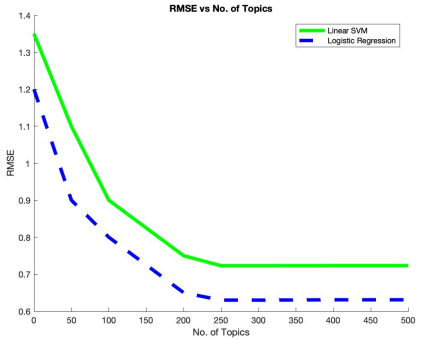



*Fig: Root Mean Squared Error 'vs' No. of Topics*

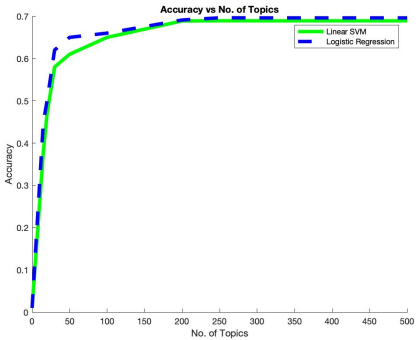



*Fig: Accuracy 'vs' No. of Topics*

# YELP RATING PREDICTION CHALLENGE

## RESPONSE TO PROPOSAL COMMENTS

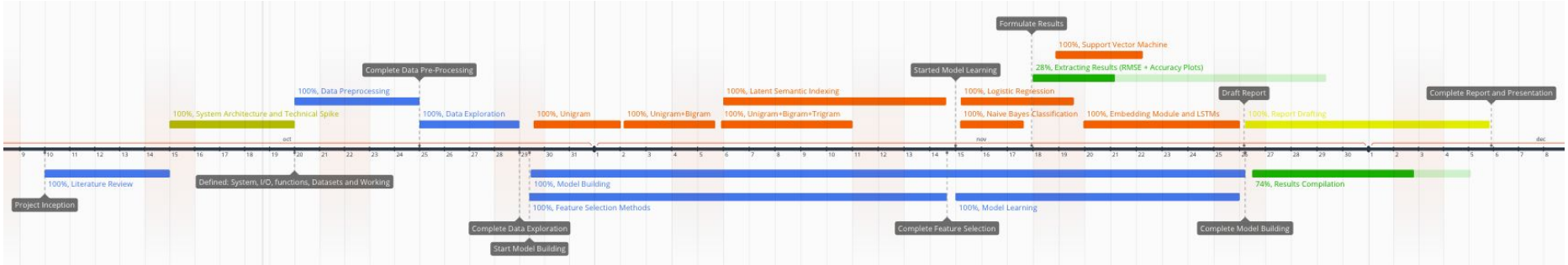
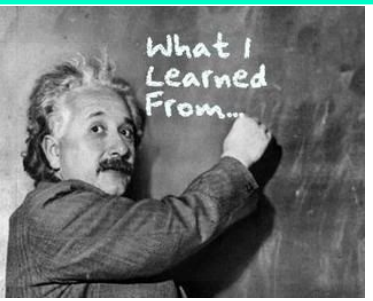



*Fig: Timeline of the project*

| Name (Contribution) | Coded | Wrote |
|---|---|---|
| Saksham (50%) | Latent Semantic Indexing, Word2Vec,LSTM, SVMs and Logistic Regression | Project Report, Slides and web application |
| Pronnoy (50%) | Unigrams, Bigrams, Trigrams, Naive Bayes, SVMs and Logistic Regression | Project Report, Slides and web application |

*Table: Contribution of each team members*

# YELP RATING PREDICTION CHALLENGE




## TAKE HOME MESSAGE

- Choosing correct Feature Selection technique is as important as choosing the correct learning technique.

- Conventional Machine Learning techniques perform well on the text provided that the features fed into them are robust and good.

- Newer Deep Learning techniques like LSTMs and Word2Vec display considerable improvements on the conventional techniques, the process of considering text as a sequential feature is certainly a step in right direction as confirmed by our result and analysis.

# YELP RATING PREDICTION CHALLENGE

**EXTRA CREDIT**


*Demo Video for our rating predictor web application*