Machine Learning for Cities

Final Report

Can we improve Yelp's services by precise user-business match-up?

Team: Chang Du (cd2682@nyu.edu), Hanxing Li (hl3282@nyu.edu), Junru Lu (lj1230@nyu.edu), Shijia Gu (sg5718@nyu.edu)

Instructor: Neill Daniel

May 9, 2019

Introduction

Nowadays, consumers attach more and more attention to the online reviews of businesses. Most of the users prefer to read the reviews before they attend to a store or restaurant. Take Yelp users for example, they will go through the stars, reviews, and pictures of one restaurant to decide which one to choose. Besides, they would like to comment on the restaurant's online rating page after dinner. The businesses have realized that online reviews have increasingly significant impact on their reputation and are essential to improving their revenue. We've now been thinking if some interactive systems can be built based on the reviews and stars to benefit both sides.

Literature Review

According to a Local Consumer Review Survey (Murphy, 2018), 86% of consumers are reading online reviews regarding services and products and 57% of consumers will only use a business of 4 or more stars. Luca (2016) identified a positive correlation between Yelp reviews and revenue of independent restaurants. "It is imperative for businesses to focus on receiving positive reviews and high ratings from all of their customers" as Shellenberger (2017) said. Hence, one important question arises that what strategy could be utilized to motivate businesses to receive higher ratings.

To answer this question, numerous prior studies that have examined review data are creating rating predictive models. Tang, Qin, Liu, Yang (2015) proposed a new neural network and introduced a user-word composition vector model (UWCVM) for review rating prediction, showing superior performances over several strong baseline methods. Li and Zhang (2014) predicted a customer's star rating for a business based on sentiment analysis of yelp review texts using machine learning algorithms, including Logistic Regression (LR), Naive Bayes (NB) and SVM.

Our novelty is that we are the pioneer to propose a high-accuracy machine learning regression model using combinations of business profile, user profile, and sentiment analysis of reviews as the features to predict a star rating of a business. Previous work only perform sentiment analysis to predict star rating instead of incorporating business profile and user profile into model.

Problem Statement

Our project is to study how can a Yelp business acquire potential reviews with high-stars on Yelp. The Yelp provides a huge dataset including more than 66 million reviews as well as related user and business information. We aim to build a model that can help those merchants to predict the level of star rating on themselves given by the customer, when the customer is using Yelp to

check nearby business. If the customer is more likely to present a review with high stars, then merchant can provide special coupons to appeal this customer.

From the perspective of Yelp users, they are also beneficial because our model will take user preference into account during the process of operation. For instance, a customer will be very pleased that when trying to find a business on Yelp, one coupon provided by some of his favored restaurant nearby pops out. Therefore, we expect this project to not only help in offering recommendations for businesses' precision marketing but also enhance Yelp user satisfaction and ultimately improve Yelp public service.

In this project, we will employ machine learning algorithms to conduct regression models based on the six datasets such as review, tip, user, business, check-in, and photo, including information about local business in 10 metropolitan areas across 2 countries, which are most recently released for the Yelp Dataset Challenge Round 13[1]. Review stars contained in the review dataset is defined as target values. The relevant and useful features corresponding to each dataset for the prediction task are created and selected by following steps.

Data Preprocessing & Analytics

Dataset 1: Review

The review dataset has 6,620,363 reviews, consisted of nine columns, including review id, user id, business id, stars, date, text (the review itself), useful (number of useful votes that reviews received), funny (number of funny votes that reviews received) and cool (number of cool votes that reviews received).

The distribution of review star ratings is in Figure 1. It is obvious from the plot that most of the reviews star ratings are pretty high, and not many terrible of the reviews. To be more specific, star ratings of 4 and 5 are 66% of all reviews while the star rating of 1 is 15% of all reviews. In other words, consumers tend to give a rather good rating.

It is apparent that the content or sentiment of reviews plays a critical role in predicting a consumer's star rating for any business. Correspondingly, we perform sentiment analysis for the texts of reviews and calculate sentiment scores for them, which determines the reviews as the degree of positive, negative and neutral. During this process, we utilized three lexicon-based methods for sentiment analysis appropriate for social media to get sentiment scores of reviews as the features of predicting star ratings given by a customer.

VADER[2] (Valence Aware Dictionary and sEntiment Reasoner) is the first approach we used. It relies on lexical resources of over 9000 token features rated on a scale from -4 (extremely negative) to 4 (extremely positive), with allowance for 0 (Neutral). The VADER sentiment

[1] https://www.yelp.com/dataset/challenge
[2] https://github.com/cjhutto/vaderSentiment

lexicon is sensitive both the polarity and the intensity of sentiments expressed in social media contexts. Pandey (2018) concludes that VADER analyses sentiments primarily based on certain key points, such as punctuation, capitalization**,** degree modifiers, conjunctions and preceding tri-gram, and performs very well with emojis, slangs and acronyms in sentences, a great advantage compared to other algorithms. Vader can directly work on the texts of reviews and would give us four sentiment scores for each text of reviews, including positive, negative, neutral and compound scores. Positive, negative and neutral scores represent the proportion of text that falls in these categories, added up to 1. The compound score is the sum of all the lexicon ratings normalized between -1(most extreme negative) and +1 (most extreme positive).

The remaining two lexicon-based methods are TextBlob[3] and AFINN[4] depending on different lexicons and attuned to social media like VADER. The AFINN lexicon is a list of English terms manually rated for valence with an integer between -5 (negative) and +5 (positive) by Finn Nielsen between 2009 and 2011.We average these ratings for the text of reviews and get the average AFINN scores from -5 to +5 for each text of reviews. The TextBlob Polarity calculated from the lexicon ranges from -1 (most negative) to 1 (most positive). Thus, we get three kinds of sentiment scores based on three lexicon-based methods as our features in review dataset to predict star ratings. And we calculate the length of texts as another feature in review dataset.

In summary, in review dataset, we selected ten features, including sentiment scores (AFINN scores, TextBlob polarity, VADER positive, negative, neutral and compound scores), text length, and number of useful, funny and cool votes that reviews received.

For analysis part, the Figure 2 shows, most TextBlob polarity, AFINN scores, and Compound scores are above zero, meaning most of the reviews are positive sentiment in the data, which is consistent with the star rating distribution we generated in Figure 1. Meanwhile, the boxplots in Figure 3 indicate that with the increase in the star rating for businesses, the average sentiment scores of reviews is increasing, which is in accordance with our expectation.

The heatmap of Figure 4 reveals correlations between review features. Consistent with our analysis below, AFFIN score, TextBlob score, VADER compound, and VADER positive score, have very positive correlation with review stars; whereas, there are high negative correlations between review stars, and features such as negative and neutral scores. Therefore, sentiment scores could be considered as significant features to predict review stars. Furthermore, text length is negatively correlated with review stars, while it is positively correlated with three marked reviews and its correlation with useful reviews is the highest among these three. This result may imply that even though longer reviews are informative that is beneficial to other users, the content more likely to be complaints or improvement suggestions to the business, which may lead to less review stars. Besides, three types of marked reviews, as other users' assessment towards the certain user's review comments, are barely correlated with review stars, which imply us that they may hardly contribute to review stars prediction.

Dataset 2: Tip

[3] https://github.com/sloria/textblob
[4] https://github.com/fnielsen/afinn

The tip dataset has 1,223,094 observations, with five columns, including text (text of the tip), date, compliment count (how many compliments the tip has), business id and user id. We apply to texts of tips the same three methods of sentiment analysis as reviews to get three kinds of sentiment scores as the features in tip dataset. Similarly, we calculate the length of texts as an important feature in tip dataset.

Thus, in tip dataset, we selected eight features, including sentiment scores (AFINN scores, TextBlob polarity, VADER positive, negative, neutral and compound scores), text length, and number of compliments that reviews received.

Figure 5 demonstrates that Tip sentiment scores based on User id aggregation, including TextBlob Polarity, AFINN scores and VADER Compound scores are mostly concentrated near zero, indicating standard deviation is small, which means Tip sentiment scores could not be an important feature of predicting reviews star rating. However, in Figure 6, most Tip sentiment scores based on Business id aggregation are positive, consistent with the distribution of review star ratings. What makes the difference between Figure 5 and 6 is the level of aggregation. One business could have more tips than one user, which is the reason why tip sentiment scores in Figure 6 is more positive than that in Figure 5.

In Figure 7 and 8, it is apparent that tips sentiment scores based on User id aggregation has no correlation with review stars, while tips sentiment scores based on Business id aggregation is correlated with review stars to a certain degree, which corresponds with our expectation.

Dataset 3: User

The user dataset after dropping has 6,685,898 lines of records. It includes 22 features as followings: user id, name, average stars given by user, ‘cool’, ‘funny’, ‘useful’, elite years, fans, friends, amount of reviews, ‘yelping since’, ‘compliment cool’, ‘compliment cute’, ‘compliment funny’, ‘compliment hot’, ‘compliment list’, ‘compliment more’, ‘compliment note’, ‘compliment photos’, ‘compliment plain’, ‘compliment profile’ and ‘compliment writer’.

The ‘average stars’ indicates the average star score of all this user’s reviews. ‘Cool’, ‘funny’, and ‘useful’ are the tags received by the user. The ‘elite’ feature lists every single year one user was elite. ‘Fans’ is the number of fans of the user. ‘Friends’ lists the id of user’s friends. ‘Review count’ is the number of reviews the user has written. ‘yelping since’ is the date user joined Yelp. Finally, the features beginning with ‘compliment’ are the numbers of different types of compliments received by the user’s reviews.

After clarifying the meaning of each feature in the User dataset, we did the pre-processing process. In this process, we dropped the least relevant string format feature ‘name’ and take ‘user id’ representing each user. Besides, we replaced the missing value with 0 and dealt with the features which not represented in the format of numbers. Firstly, we calculated the sum of elite years of the user. Then, do the same calculation on feature ‘friends’. Furthermore, we also transformed the ‘yelping since’ feature to the days one user with Yelp. As a result, we got a tidy data frame indicating 20 user features.

To know better the processed user dataset and the relationship between review star and user related features, we did the visualization among all these features. We plot the correlation between these features by heatmap, plot the histogram of all user features to show the distribution of those features, and plot the boxplot to show relationship between review star and user features. According to our results, we found some of the user features have significant relationship with the stars of one review. Take user's average stars for example, Figure 9 shows a high correlation of 0.56 with review stars. Besides, most user's average stars are between 3 and 5 stars, and the number of user average 5 stars has a peak. The number of user average 1 stars also is a peak of range [1,3) in Figure 10.The boxplot in Figure 11 shows a positive trends between user average review stars and review ratings.

Dataset 4: Business

The business dataset contains 192,609 businesses that mostly are restaurants. Each business connects to some attributes, including 22 character unique business id, name, address, located city, located state, postal code, specific coordinates, half-stars business rating from 1.0 to 5.0, number of review, categories, opening hours during a typical week and so on.

We suppose that postal code, address, located city and state information can be contained by specific coordinates. Thus we only keep 6 features: business id, coordinates, ratings, number of review, categories and opening hours, in which the first four is alphanumeric. The categories is an array of strings of business categories, such as ["Mexican", "Burgers", "Gastropubs"]. The opening hours is an object of key day to value hours, hours are using a 24hr clock, such as {'Friday': '9:0-1:0', 'Sunday': '9:0-0:0', 'Tu…}.

The preprocessing work for 'categories' is to simply count the categories it has. In fact, we lost a lot of detailed information here, which is because it will be impossible to generate a set of multi-level cuisine dictionary based on this feature. Meanwhile, we map the hours into a seven-column matrix and calculate the opening duration for every single day. The limitation that must be pointed out is the Yelp did not give a clear description about the missing values in the hours. In many cases, the opening hours contains data like "Friday": None. We don't know exactly if it refers the business not open on Friday or they just not post their opening hours during Fridays. There are about 23% businesses without any opening information, and about 57% contains some Nones. We suppose it will be a great loss to entirely remove business records if they provide vague opening hours, so we decide to use zero values to fill these nan values.

The average opening hours over a typical week is shown in figure 12. We can see that businesses tend to have a rest in weekends.

Figure 13 shows a positive trends between review ratings and business ratings. However, it is not exactly in a diagonally equal relation.

Dataset 5: Checkin

The check-in data is simple, containing business id and a string which is a comma-separated list of timestamps for each check-in, each with format YYYY-MM-DD HH:MM:SS. After careful

look-up, we calculates the time gaps between all timestamps and generate three features based on these gaps: the total amount, average value and the variance of the gaps.

The observation on average amount of check-in over current business ratings is kind of anti-intuitive. The figure 14 shows that not a higher reputation business deserves more crazy check-in waves.

Dataset 6: Photo
The photo dataset includes four features: 22 character unique photo_id, business_id related with checkin and business data, captions of the photo and labels imply the category the photo belongs to. The captions are abandoned since they are as impossible as the 'categories' feature in the business data to be dealt with. For the labels, there are 5 distinct values: food, inside, outside, menu and drink. Just like the 'opening hours' feature in the business data, we count the number of each type in each business as a set of five-column features.

Statistics on photo data show that a business will provide 0.6 drink photo, 3.8 food photo, 1.7 inside photo, 0.1 menu photo and 0.4 outside photo.

Coding Framework

Data Merging
Some of our datasets is massive, such as review dataset contains 6,620,363 records. It will cost too much time to load through pandas. We apply a file reading stream with a self-defined dict structure to load the data, and then transform the dict into a dataframe. After pre-processings on individual datasets, we firstly merge all business infos with an order of tip, checkin and photos, then merge all data with an order of all business info, user, tip and reviews[5]. The final data looks like Table 1.

Feature Engineering[6]
Clean Noisy Data Remove all samples with nan values, and reset numeric data into float or int type. 4 noisy data are detected and deleted, and the final size of the merged data is 6,685,898.

Normalization Normalize 67 numeric features by formula: norm = (x - average(x)) / standard deviation(x). By normalization, the data is redistributed into a gaussian distribution. The purpose of normalization is to eliminate the influence brought by different scales between features.

Extreme Binary Check To check if some real-valued features distributed extremely over its range. If true, we need to discretize the features into discrete forms.

[5] https://github.com/LuJunru/MLC2019_Final_Project/blob/master/Codes/Data_Preprocessing.ipynb
[6] https://github.com/LuJunru/MLC2019_Final_Project/blob/master/Codes/Modelling_LIMIT.ipynb

Feature Selection We tried to use pearson correlation, l1 regularization, l2 regularization, PCA and LDA methods to optimally select features. However, the model result demonstrates using all features will be the best.

Anomaly Detection
One of the easiest way to check the anomalous data is to use PCA decomposition. We decompose the features into 2 dimensions, and compare the 1st component versus the 2nd component. Figure 15 shows that only very tiny amount of anomaly exist.

Next step is following modelling part.

Modelling[7]

Metrics
The metrics we are going to use is precise accuracy, recall and benefit accuracy. The first two are traditional indicators. Since actually recall is over 0.9999, it will be proper to only use accuracy other than f1 scores. Except for that, we self-define a benefit accuracy, which means when the predicted review rating is no less than the true review rating, we count it as a true prediction. The reason that we can use benefit accuracy here is even if the business owner hooks potential guests by predicted value, they will receive a positive return on their reputation.

Cross Validation
We randomly split the data into 46 million train set and 20 million test set. We suppose to use train set for training and use the out-of-sample test set for cross validation.

Regression
Although the review ratings are discrete, we apply regression model on it, as classification will ignore the relative relations between labels. For example, 5.0 should be naturally bigger than 4.0, but classification will neglect that.

SVR Vladimir N. Vapnik, Harris Drucker, Christopher J. C. Burges, Linda Kaufman and Alexander J. Smola proposed this regression version of SVM (SVR) in 1996. We test this model on a limited subset, which offers almost best precise accuracy among all of our models. However, SVR fits almost infinite time on the training set.

Random Forest Random Forest is one of the ensemble model, which takes both accuracy and efficiency into account. The strategy of RF is to randomly select sample data and sample features to fits a number of decision trees, and average the trees to improve the predictive accuracy and control over-fitting. The RF model works around 3% lower in precise accuracy compared with SVR on limited dataset, but 0.5% higher in benefit accuracy. We finally runs the RF model on the 46 million train dataset, and obtains 0.559 precise accuracy and 0.789 benefit accuracy on the 20 million test dataset, as well as 100% recall. The interpretability of the model is plotted by the feature importance as figure 16.

[7] https://github.com/LuJunru/MLC2019_Final_Project/blob/master/Codes/Modelling.ipynb

XGboost Tianqi Chen and Carlos Guestrin raised XGboost in 2016, which is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGboost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way[8]. XGboost can reach a better results on both precise accuracy and benefit accuracy compared with RF on the same limited sample dataset, while we have to give up it since it does not run fast as it declares on the whole big dataset.

LGBM Microsoft and PKU invented Light Gradient Boosting Machine in 2017, which is an another tree-based gradient boosting framework. The difference between LGBM and other tree-based model such as XGboost is that LGBM grows tree vertically while other algorithm grows trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithm grows level-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm[9].

The LGBM model runs as good as SVR model on the limited samples, and can reach a best performance on the 66 million dataset. We finally get 0.578 precise accuracy and 0.785 benefit accuracy with 99.999% recall rate. The interpretability of the model is plotted by the feature importance as figure 17. That's why we choose LGBM to be our final solution.

Prediction
We will implement model for a review rating in following two cases. First, for current yelp users, we've got their past review, user and tip data, as well as the business data. These information will be combined to build a test feature group and used for prediction. Second, for new users or users without any information, we will put 0.0 in all 38 features that required from the user, and then do the prediction. Out test shows that even for these users, our model will reach a 36% probability to attract businesses to offer discounts.

Conclusion

The LGBM provides a model with approximate 80% accuracy that a business can use to get a precise or higher rating expectation on any potential good review.

When we focus deeper on the feature importance explained by the model in Figure 17, we will get unexpected consistency with what we have found below within the data analytics part.

The average ratings of a user indicates his or her general tendency on judging a business, thus is the most important features to predict the potential rating. In Figure 4, the correlations between

8 https://xgboost.readthedocs.io/en/latest/
9 https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc

review stars and user stars is up to +0.55. The next is the average ratings of a business, which indicates a common phenomenon called congregational psychology. We tend to judge a business following what it is judged in a long term. The Figure 13 shows a positive trend between review ratings and average business stars. What's more, 6 review features continuingly show up from the third important features. According to our analysis about Figure 3 and 4, information from the review itself has strong indications on its ratings. Another interesting thing is about the tip data. As we can see in both the importance Figure 17 and tip plots Figure 5, 6, 7 and 8, almost every feature generated by tip is not very contributing.

Future Improvement

Review Length We now only count the character length of the review sentences. In fact, we should do some natural language processing work: split the sentence, remove punctuations and stopwords, then count the amount of words. The new feature will be more meaningful than the current one, as what meaningful in a review is only the actual words.

Business Categories All of our data should be further decomposed by business categories, since many of people have distinct interests on different type of categories provided by businesses.

Parameters Optimism As we have done on the limited dataset, we can use grid search to find best parameters for our models. We meet some overfitting problem right now, but we will try to realize in the near future.

Ensemble Stacking What we have achieved are single models, though contains boosting and bagging models as ensemble methods. We can try stacking, which is the last ensemble model, to improve the performance of single models. By the way, another problem that will be raised on the stacking model is time complexity, since the second phase of stacking is not parallel.

Team Roles

Junru Lu Data preprocessing on business, checkin and photo dataset & Modelling
Shijia Gu Data preprocessing on review, and tip dataset & correlation analysis
Chang Du Data preprocessing on user dataset & analysis
Hanxing Li Data preprocessing on review, and tip dataset & sentiment analysis

References

[1] Drucker, Harris; Burges, Christopher J. C.; Kaufman, Linda; Smola, Alexander J.; and Vapnik, Vladimir N. (1997); *Support Vector Regression Machines*, in Advances in Neural Information Processing Systems 9, NIPS 1996, 155–161, MIT Press.

[2] Tang, D., Qin, B., Liu, T., & Yang, Y. (2015). User Modeling with Neural Network for Review Rating Prediction. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 1340-1346.

[3] Luca, M. (2016). Reviews, reputation, and revenue: The case of Yelp. com.

[4] Murphy, R. (2019). Local Consumer Review Survey | Online Reviews Statistics & Trends. Retrieved from https://www.brightlocal.com/research/local-consumer-review-survey/

[5] Li, C., & Zhang, J. (2014). Prediction of Yelp Review Star Rating using Sentiment Analysis. Retrieved from http://cs229.stanford.edu/proj2014.

[6] Shellenberger, P. (2017). Predicting Yelp Food Establishment Ratings Based on Business Attributes. Retrieved from https://scholars.unh.edu/honors/374.

[7] Pandey, P. (2018). Simplifying Sentiment Analysis using VADER in Python (on Social Media Text). Retrieved from https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f.

[8] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In 22nd SIGKDD Conference on Knowledge Discovery and Data Mining, 2016.

[9] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[C]//Advances in Neural Information Processing Systems. 2017: 3146-3154.

Appendix

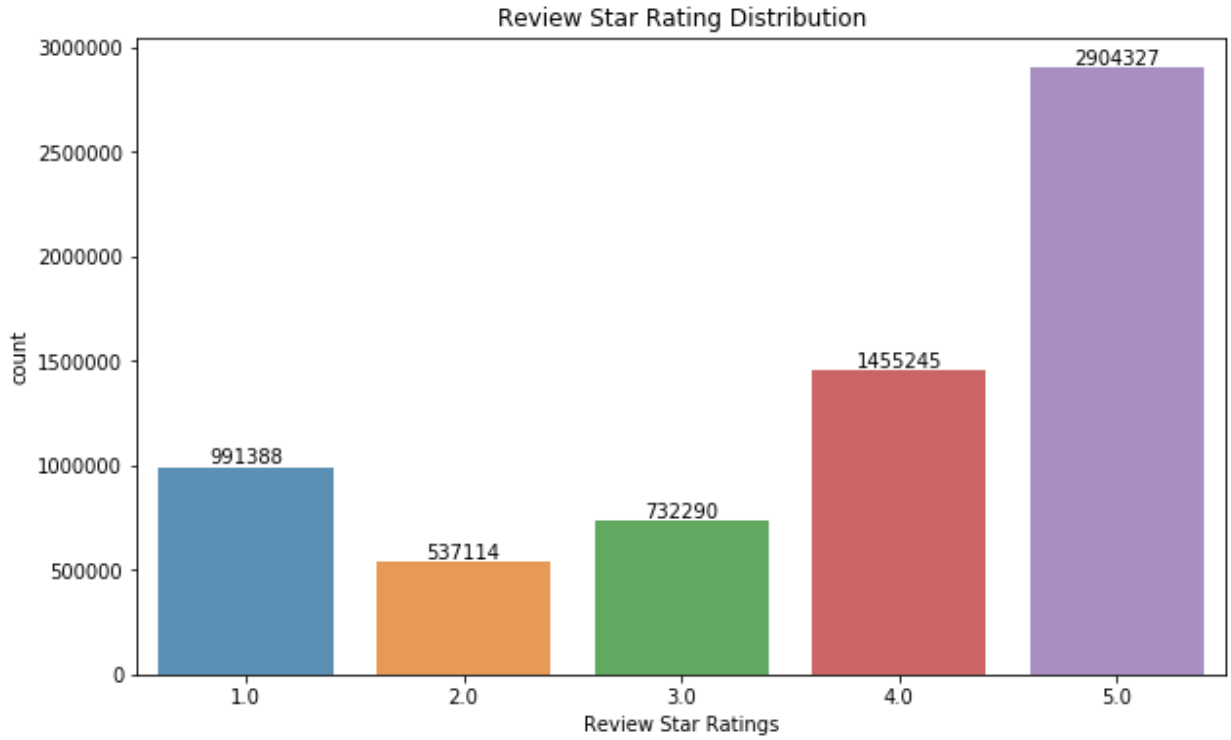

Figure 1: the Distribution of Star Ratings

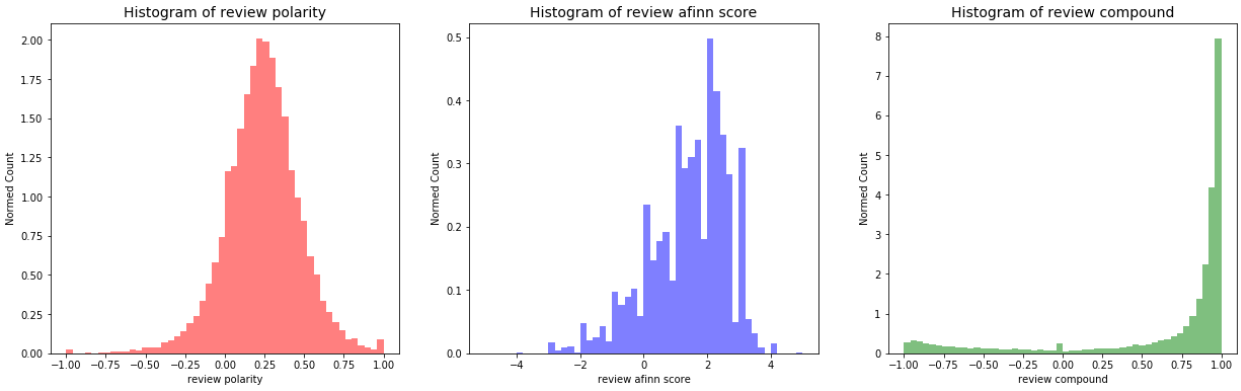

Figure 2: Distribution of Reviews Sentiment Scores

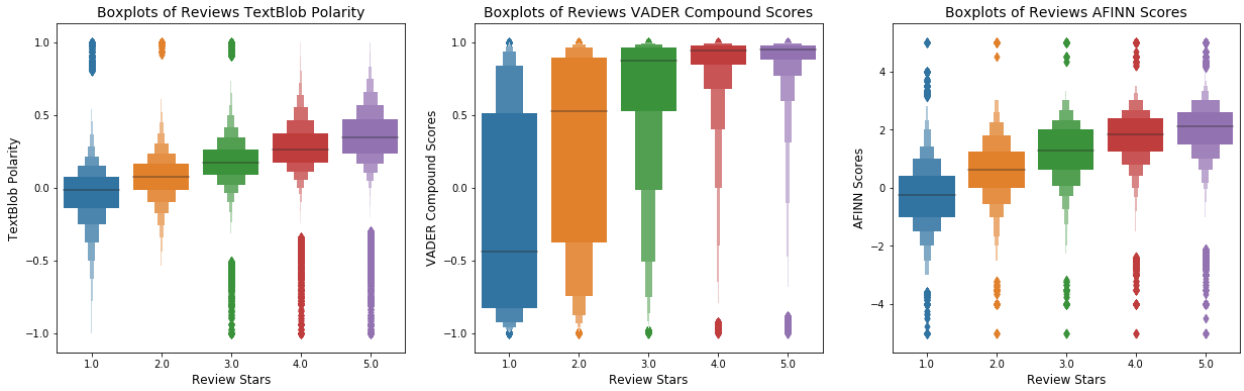

Figure 3: Boxplots of Reviews Sentiment Scores over Review Stars

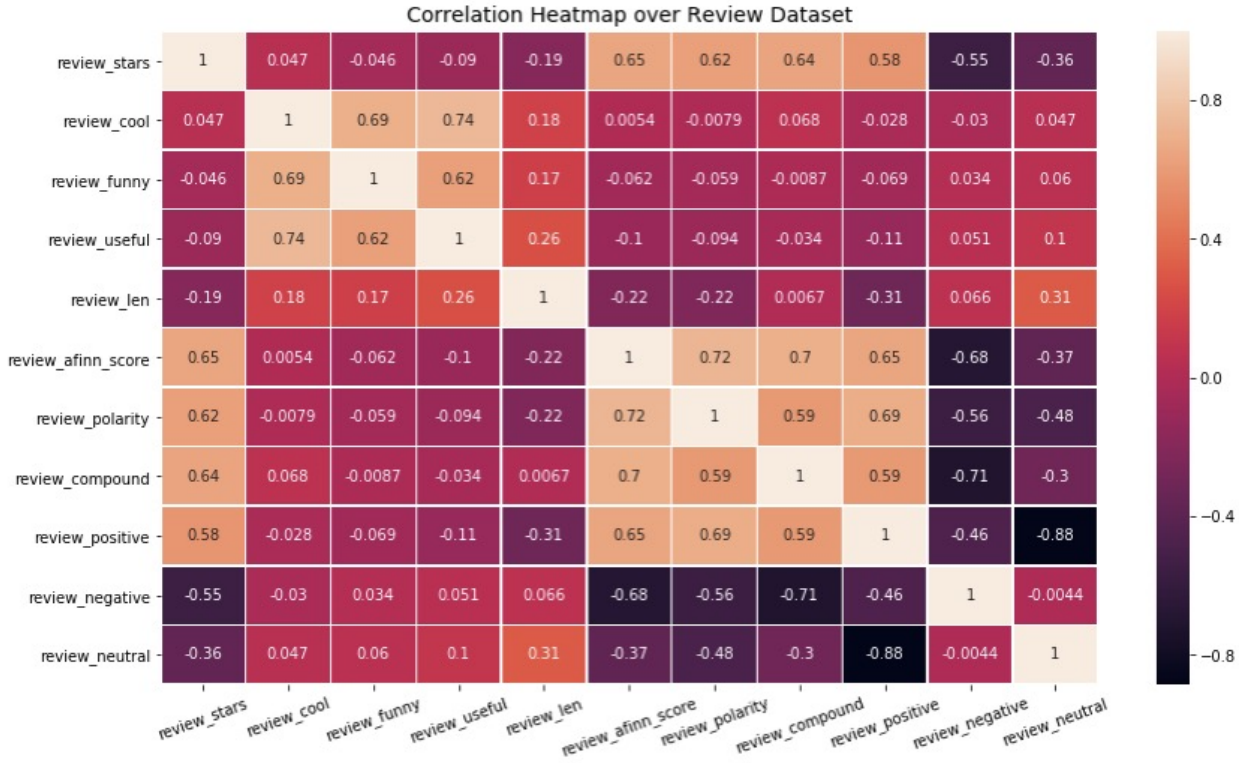

Figure 4: Correlation Heatmap over Review Dataset

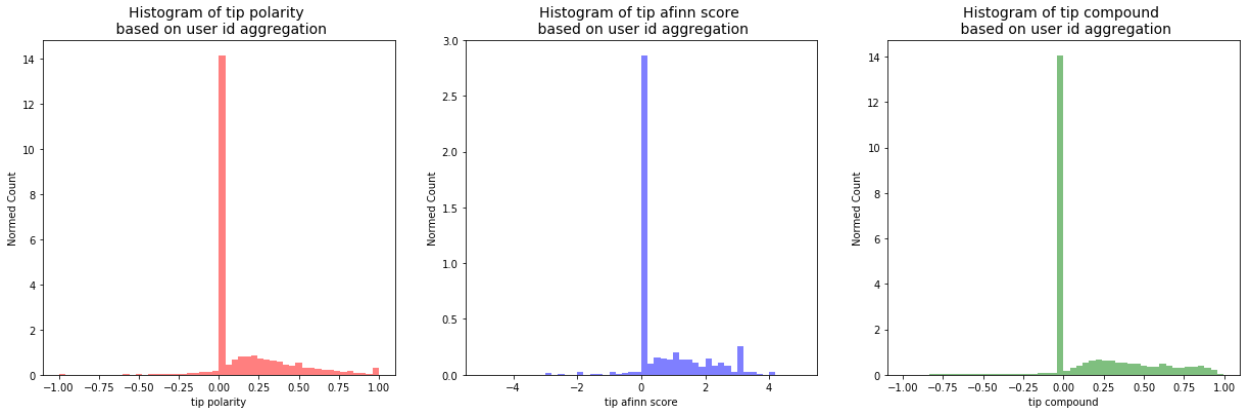

Figure 5: Distribution of Tips Sentiment Scores based on User id aggregation

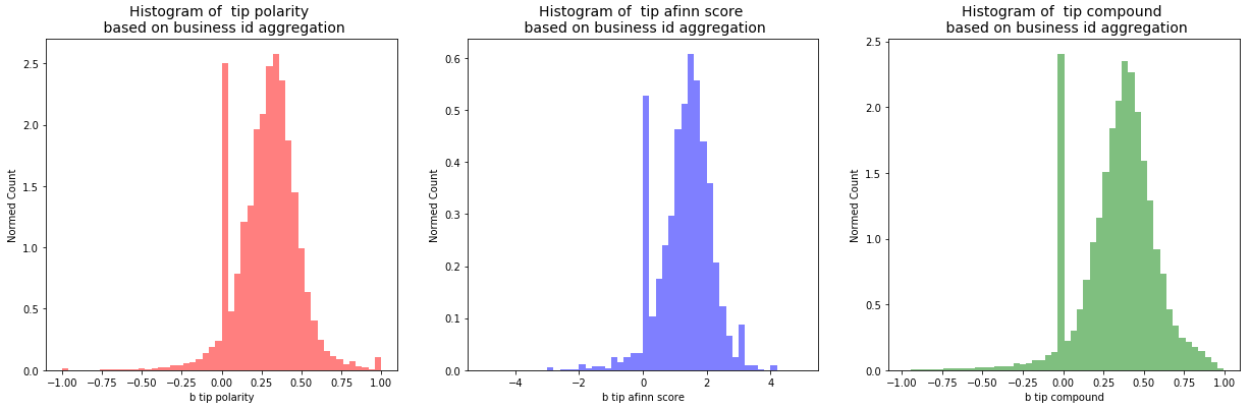

Figure 6: Distribution of Tips Sentiment Scores based on Business id aggregation

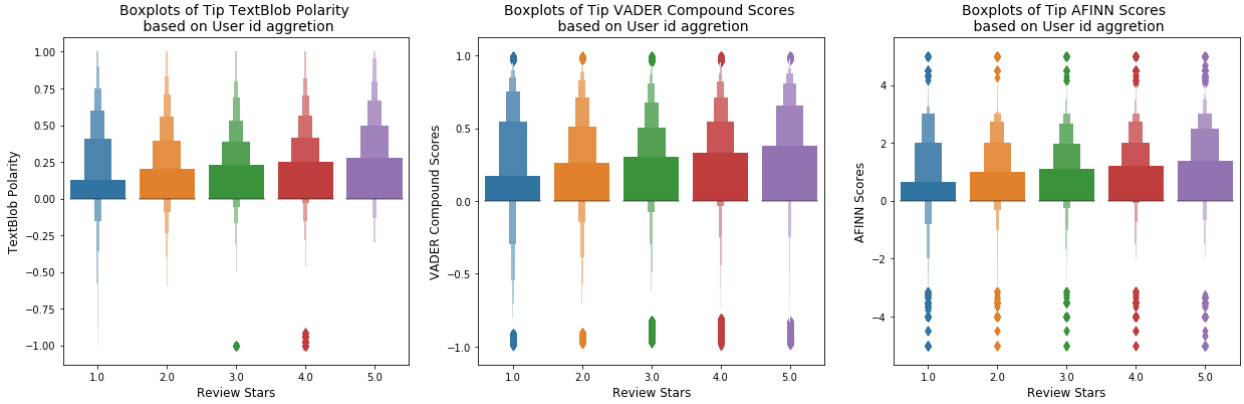

Figure 7: Boxplots of Tips Sentiment Scores over Review Stars based on User id aggregation

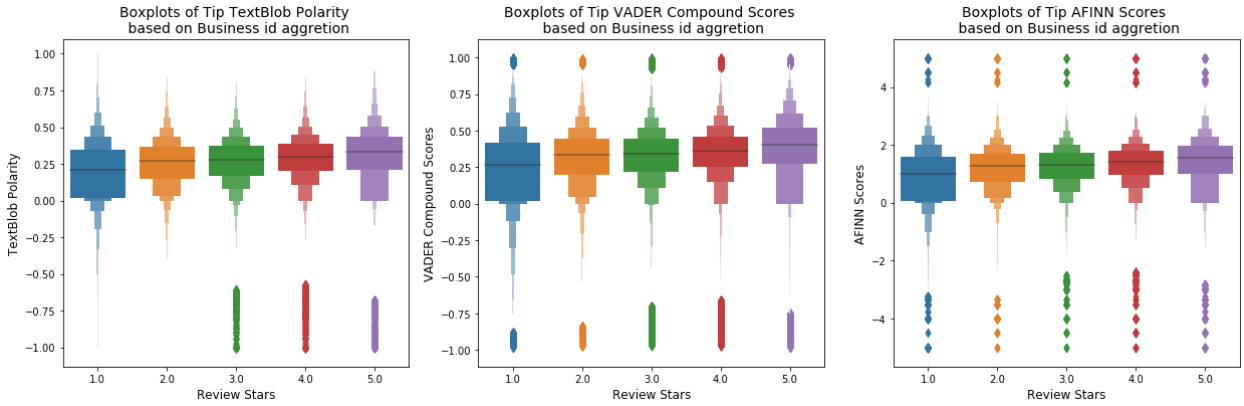

Figure 8: Boxplots of Tips Sentiment Scores over Review Stars based on Business id aggregation

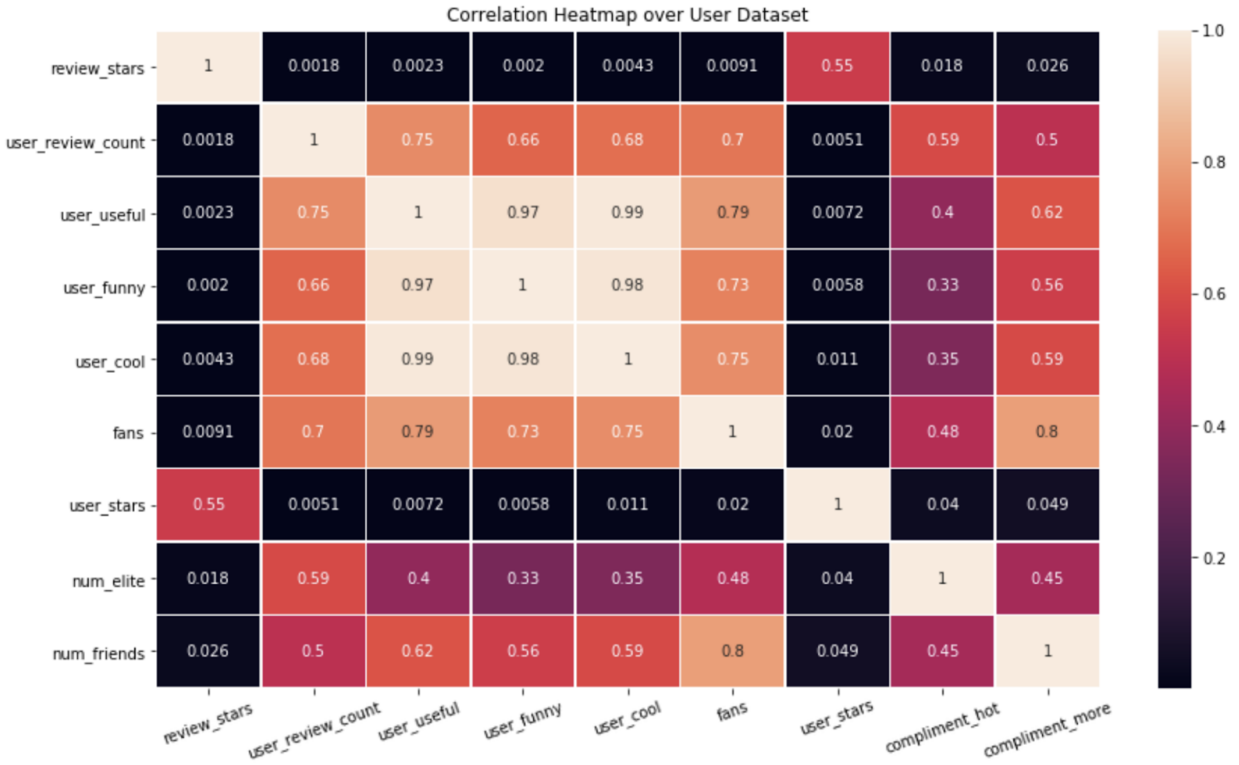

Figure 9: Correlation Heatmap over User Dataset

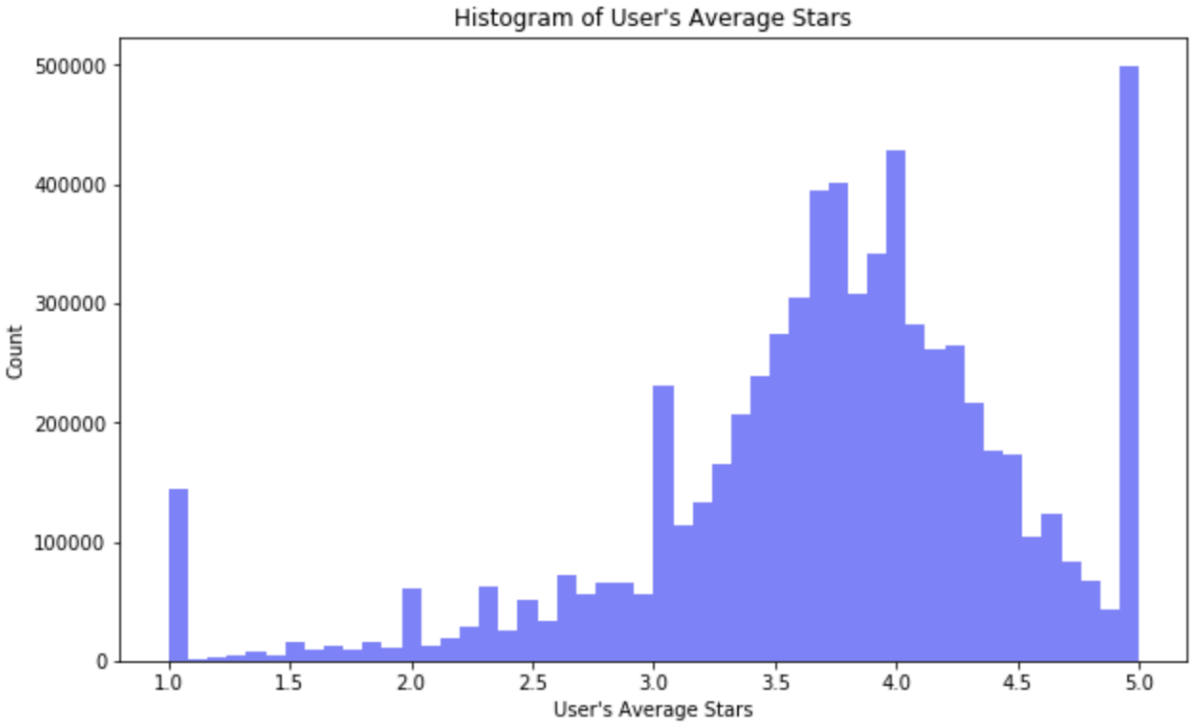

Figure 10: Histogram of User's Average Stars

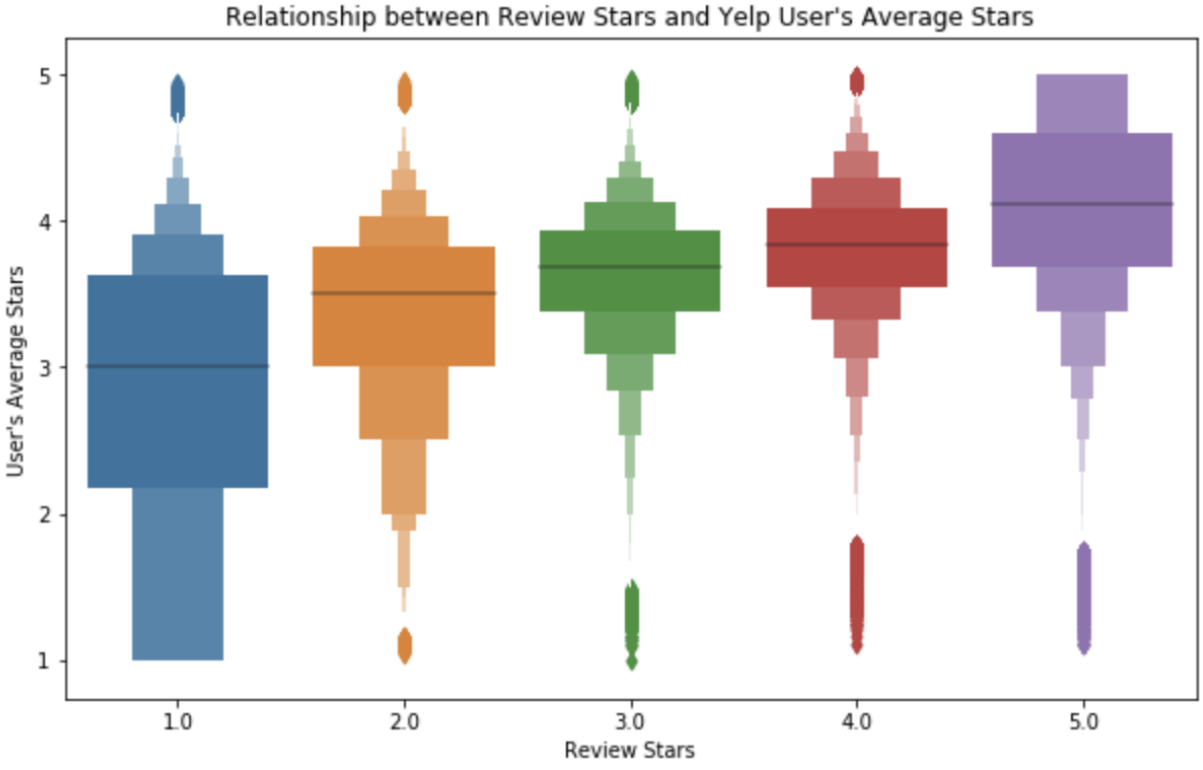

Figure 11: Relationship between Review Stars and Yelp User's Average Stars

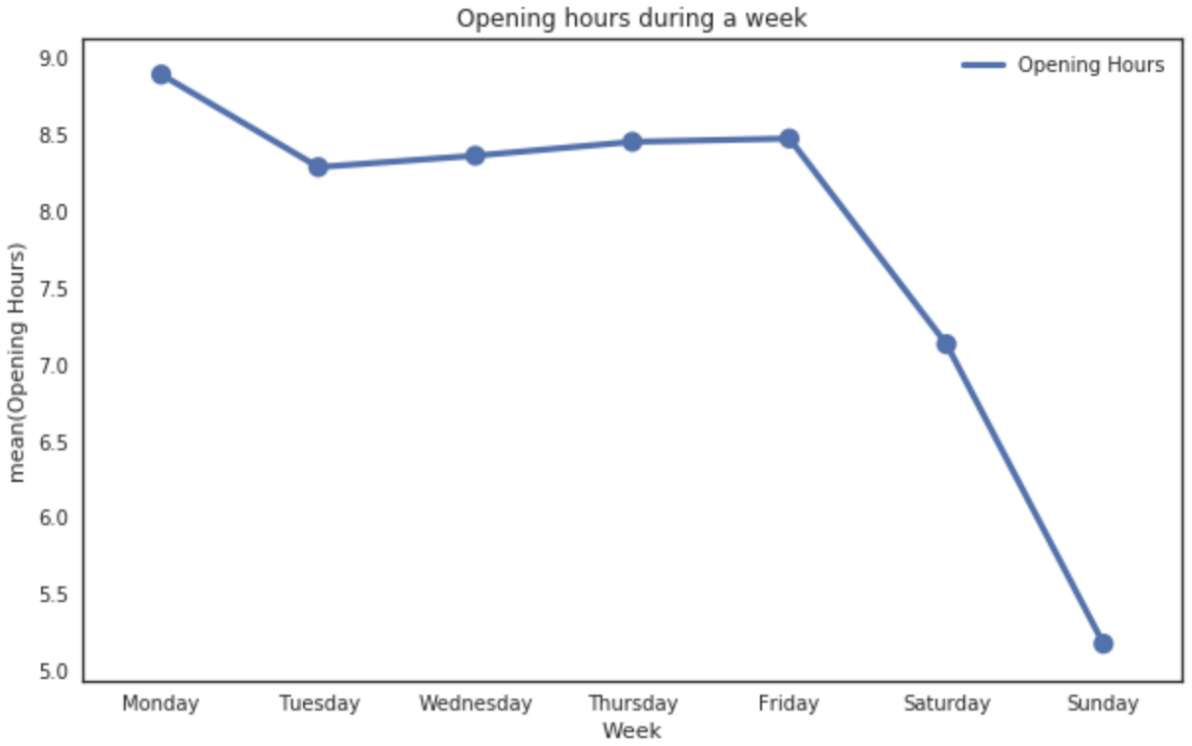

Figure 12: The average opening hours of a average business over a typical week

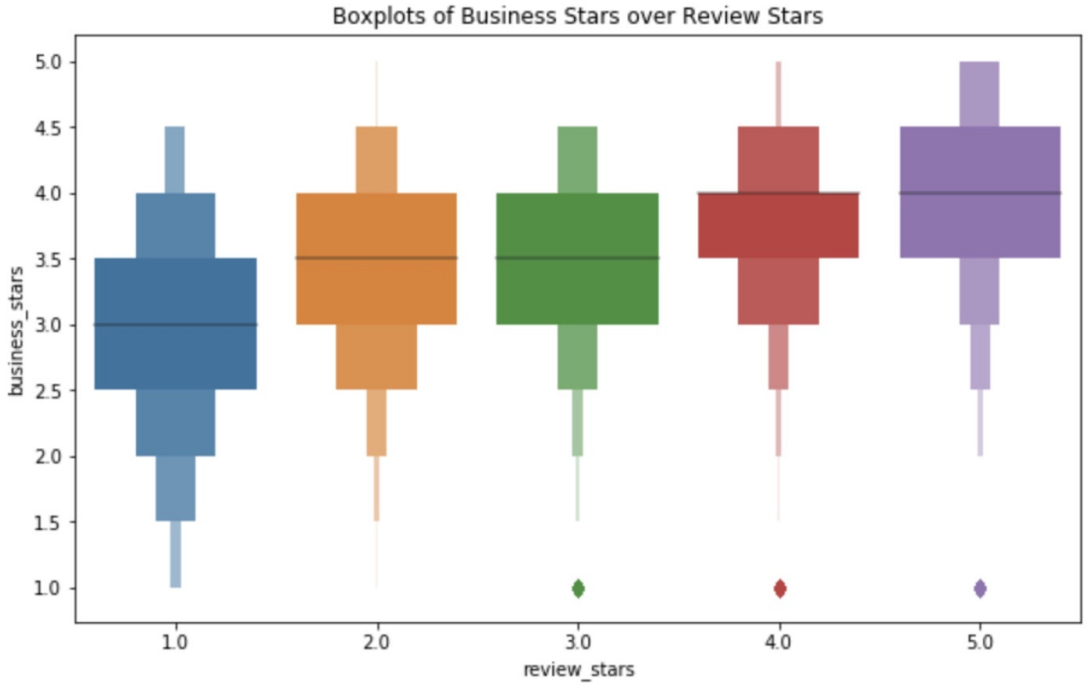

Figure 13: Boxplots of Business Stars over Review Stars

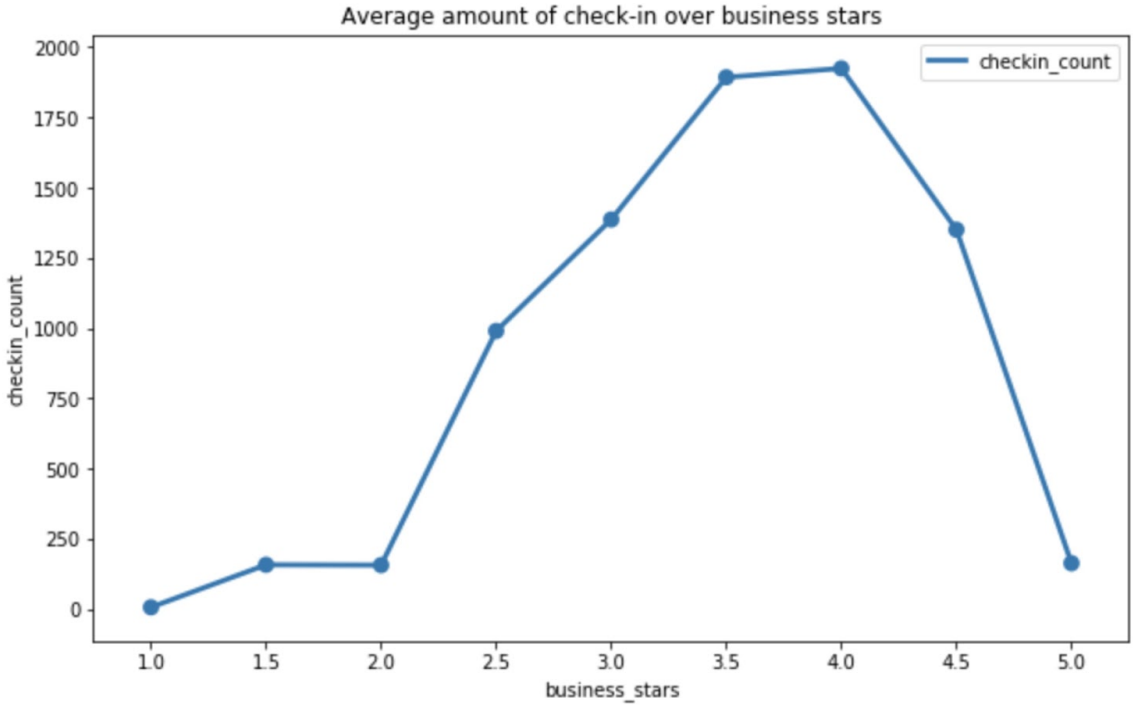

Figure 14: Average amount of check-in over business stars

	Review Features	User Features	Business Features	User & Tip	Business & Tip	Checkin Features	Photo Features	Target Value
Sample Field	Review Scores	User Stars	Business Stars	Avg Tip Star	Avg Tip Star (B)	Checkin Counts	"Food" count	Review Stars
Sample	0.80	3.6	4.5	4.2	3.8	29	20	4.0

Table 1: Sample of Final Dataset, 67 numeric features and 1 target value.

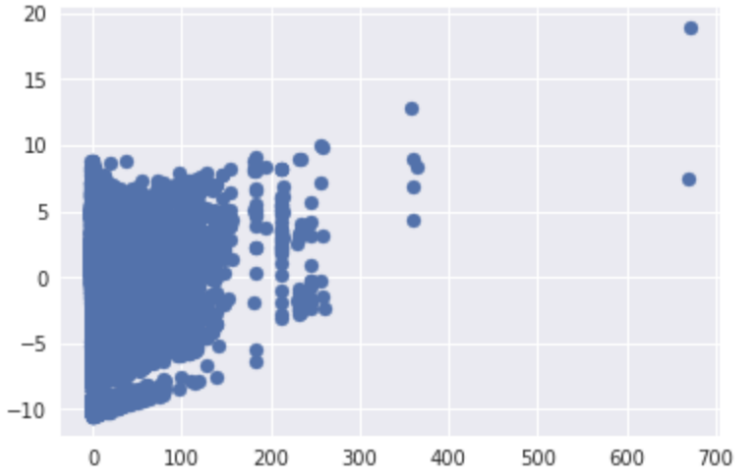

Figure 15: Using 2D PCA Components to show anomaly samples

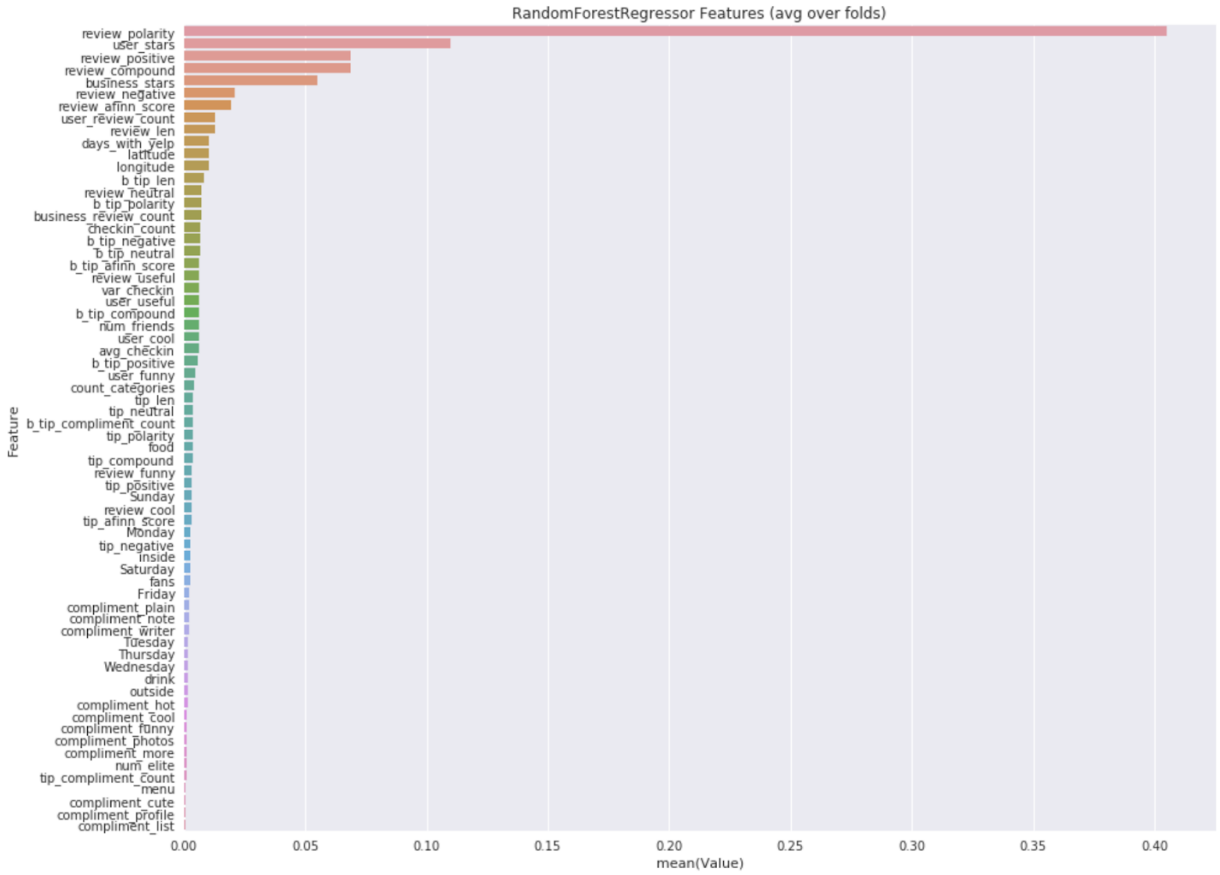

Figure 16: Feature Importance revealed by Random Forest Regressor on 46 million train set

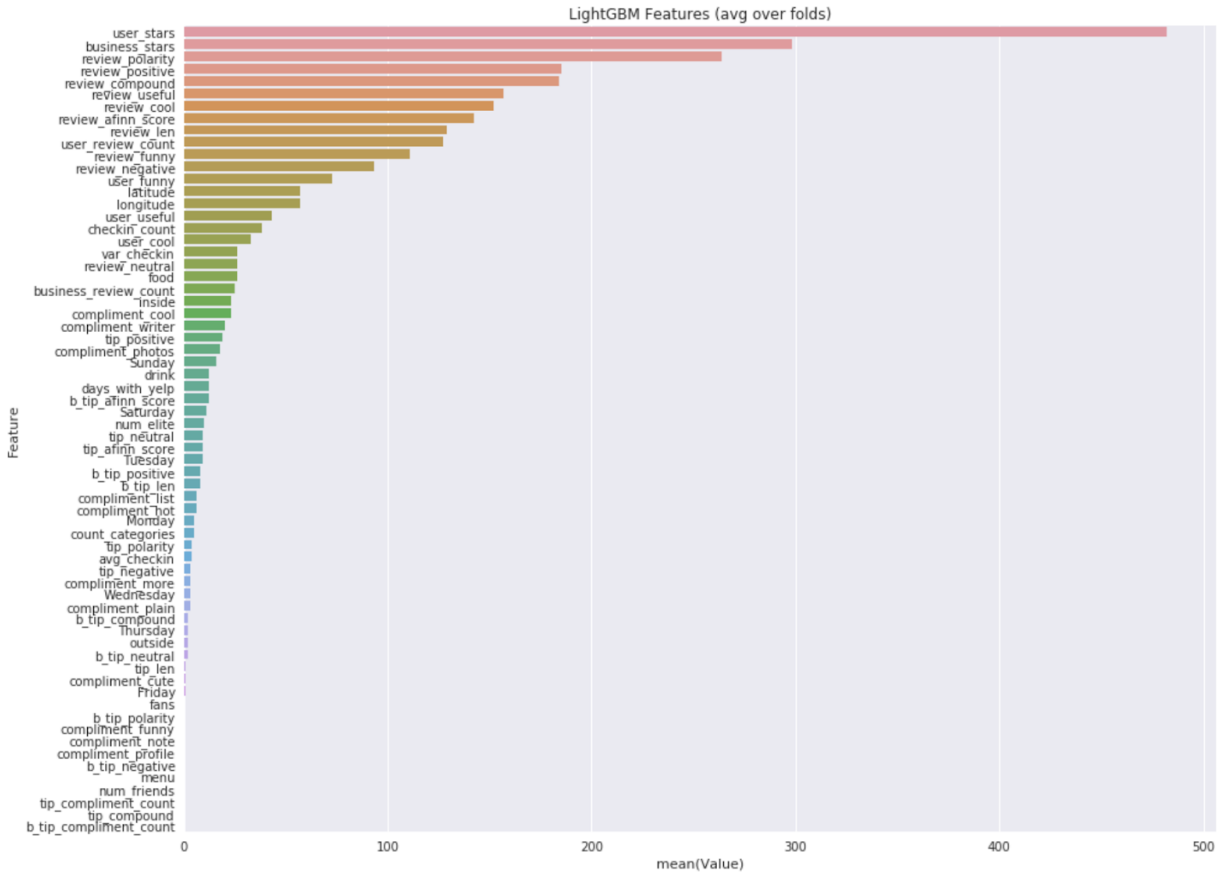

Figure 17: Feature Importance revealed by LGBM on 46 million train set