DEBIASING ONLINE REVIEWS WITH MACHINE LEARNING

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Shivank Goel

May 2020

ABSTRACT

Online review aggregation platforms suffer from biases which can lead to distress to both the platforms and their consumers as the average rating crowd sourced on these platforms do not represent the correct perceived quality of the product or service. We look at the problem of polarization bias on Yelp and present the evaluation of an estimation model to determine the unbiased average rating. We explore how the Yelp's elite membership program helps in cutting down the bias. Our results propose that the average biased rating listed on platforms is correlated with the true unbiased rating and by including more information from online reviews as input features of our model we can get a reasonably well estimate of the average unbiased ratings. We propose and compare several predictive models to estimate unbiased average ratings and show how textual data can play a critical role in enhancing the predictive power. Our results can help review aggregation platforms to determine mechanisms to cut down the bias on their platforms and can benefit businesses and consumers on their platform to access a fairer metric for service quality.

BIOGRAPHICAL SKETCH

Shivank Goel is a second-year MS candidate in the Operations Technology & Information Management department at Cornell University's SC Johnson College of Business. Shivank's research interests are to examine the impact of policy and economic factors using data-driven analytical methods from the intersection of Econometrics and Machine Learning. Shivank graduated with a Bachelor's degree in Computer Science from the Indian Institute of Technology (IIT) Delhi in 2018; in 2017, Shivank served as a software engineering intern with the big data team at Microsoft in Bangalore. At IIT, he held part in several data science projects. After joining Cornell's prestigious business school, Shivank was further reinforced of his enthusiasm for data science and its transformational power in dealing with complex business problems. After graduating from Cornell, Shivank will join Amazon's headquarters in Seattle, beginning in Fall of 2020, as a full-time software engineer. In his leisure time, Shivank runs a YouTube channel, where he regularly posts video tutorials for algorithmic coding, statistics and data science.

This is dedicated to all Cornell graduate students.

ACKNOWLEDGEMENTS

I would first like to express my sincere thanks to my thesis advisors Prof. Li Chen and Prof. Shawn Mankad of the SC Johnson College of Business at Cornell University for their continuous guidance and support. The gate to Prof. Chen and Prof. Mankad's office was always open for me whenever I got caught or had a query about my research or writing. They always heard to accommodate the project scope corresponding to my interests and expertise so I could add more towards my thesis. They always allowed this paper to be my own work, but drove me in the right direction whenever I called for it.

I would also like to thank all the faculty members of Operations Technology and Information Management (OTIM) department who helped me in this journey and backed me. Special thanks to Prof. Andrew Davis, Prof. Vishal Gaur, Prof. Chris Forman and Prof. Karan Girotra who were in close touch with me and provided their valuable advice throughout my graduate studies. I would also like to specially thank the director of graduate studies for Johnson, Prof. Vrinda Kadiyali for her guidance and support during my stay at Johnson.

I would also like to acknowledge my friends and colleagues at Cornell University as the second reader of this thesis, and I am gratefully thankful to them for their precious comments on this thesis. Finally, I would express my gratitude to my parents for giving me with unfailing support and unceasing encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been achievable without them. Thank you.

Shivank Goel

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vi
List of Tables . . . vii
List of Figures . . . viii

1 Introduction **1**
1.1 Introduction . . . 1
1.2 Literature Review . . . 3
1.2.1 Economic Impact of Online Reviews . . . 4
1.2.2 Acquisition and Under-Reporting Bias . . . 5
1.2.3 Textual Data Analytics on Online reviews . . . 7
1.3 Dataset . . . 7
1.3.1 Performance Metrics . . . 9

2 Preliminary Observations **11**
2.1 Acquisition Bias . . . 11
2.2 Under-Reporting Bias . . . 13

3 Results **16**
3.1 Relation Between Average Elite Ratings and Non-Elite Ratings . . . 16
3.2 Adding Non-Elite Ratings Distribution to Regression . . . 20
3.3 Adding Text Information : Tf-Idf . . . 24
3.4 Pushing Boundaries with Machine Learning . . . 26
3.4.1 Bagging with Linear Regression . . . 26
3.4.2 Random Forest Regression . . . 27
3.4.3 Support Vector Regression . . . 31
3.4.4 Gradient Boosted Decision Trees . . . 32
3.4.5 Neural Networks . . . 32
3.5 Trade-Off: Heterogeneity and Amount of Data . . . 33

4 Conclusion **36**

A Yelp's Elite Program **37**

B ML Algorithms **38**
B.1 Decision Trees . . . 38
B.2 Bagging . . . 40
B.3 Boosting . . . 41
B.4 Principal Component Analysis . . . 42
B.5 Support Vector Regression and Kernels . . . 43
B.6 Neural Networks . . . 46

LIST OF TABLES

1.1 Summary for different categories in Yelp dataset 9

2.1 Number of distinct businesses reviewed : elites vs non-elites 12
2.2 Comparison of distribution of average business ratings 13
2.3 Linear Probability Model Results 15

3.1 Distribution of non-elites for businesses with higher elite rating vs lower elite rating 19
3.2 Performance of regression and identity on test dataset 20
3.3 Using non-elite distribution as regression inputs 21
3.4 Using the buckets in the ridge regression 23
3.5 Adding text information to regression 25
3.6 Comparison of bagging with linear regression 27
3.7 Random forest comparison with linear regression without text . . . 28
3.8 Random forest comparison with linear regression with text 30
3.9 Relative feature importance 31
3.10 SVR comparison with linear regression 32
3.11 Gradient boosted trees comparison with linear regression 32
3.12 Neural networks comparison with linear regression 33
3.13 Comparison between all machine learning models 33

LIST OF FIGURES

1.1 Top five business categories 8

2.1 Non-elite ratings distribution 14
2.2 Elite ratings distribution 14

3.1 Comparison of average ratings by non-elites and elites 17
3.2 Businesses with higher average elite rating than average non-elite . 18
3.3 Businesses with lower average elite rating than average non-elite . . 18
3.4 Debiasing effect of elites assumin all non-elite ratings are same . . 22
3.5 De-biasing effect of elites using non-elite buckets 24
3.6 Comparison of average ratings by non-elites and elites 29
3.7 Comparison of average ratings by non-elites and elites 34
3.8 Comparison of average ratings by non-elites and elites 34

B.1 Entropy variation with probability of first label 39
B.2 Working of random forests 41
B.3 Support vector regression 44

CHAPTER 1

INTRODUCTION

1.1 Introduction

Online reviews are a powerful resource for consumers to determine the quality of service. In today's tech savvy world, 90% of consumers look for a business online and read online reviews before visiting a business.[1] Star ratings provide a quick way for search engines and websites to share the experience of consumers with a particular product or service. These ratings significantly impact revenues. Luca (2016) shows that a one star increase in a Yelp rating contributes to a 5 to 9% hike in the business' revenue.

Considering this economic impact of reviews, it is vital for both the businesses and the consumers that the platforms provide an unbiased estimate of the star ratings of the products and services offered. Specifically, a partisan review for a product/service may have three negative effects. First, a biased review may *directly* impact sales and revenues for the business. Second, it could also lead to a bad customer experience, which hinders the long-term customer commitment, thereby *indirectly* impacting the business in the long run. A poor customer experience also leads to a reputation loss for the platform.

Most users voluntarily report ratings on the online review platforms. This inevitably contributes to the ratings being distorted. For example, Hu et al. (2017) identify two self-selection biases prevalent on the review platforms: (i) *acquisition bias* and (ii) *under-reporting bias* (also known as *polarization bias*). Acquisition

[1] https://www.brightlocal.com/research/local-consumer-review-survey

bias occurs because consumers self-select to acquire a service, i.e., they opt in if they already perceive that the quality of service would be great a-priori. On the other hand, under-reporting bias occurs because online platforms tend to aggregate reviews from the users with the most polarized opinions (extremely positive or negative). Thus, these reviews give a misleading representation of the product/service quality.

In this work, we are interested in developing models to estimate the unbiased star rating of businesses. We propose a method that leverages a subset of potentially unbiased reviews to formulate a de-biasing model for the (biased) average rating across all reviewers. Furthermore, our larger purpose is to demonstrate how non-monetary incentives can turn out to be an efficient and economical way for the online review aggregation platforms to curate such a pool of unbiased users or reviews. Specifically, we study Yelp's loyalty program, also known as Yelp's elite squad, to see how we can use reviews from these elite members to get an estimate of unbiased average star ratings. To know further about Yelp's loyalty program, please refer to Appendix A.

For this study we use Yelp dataset.[2] Yelp dataset contains $6,685,902$ reviews out of which Yelp's elite members have written 18.4%, i.e., $1,231,492$ reviews. We describe this dataset in more detail in Section 1.3. We found in our data that the Yelp's elite members help to reduce the *acquisition bias* and the *under-reporting bias*. Elite members obtain non-monetary utility by writing online reviews that occurs by reinforcing their track record to protect their elite membership for the subsequent year. The non-monetary utility provides an incentive for elite members to report more reviews online and visit new businesses. In this study, we statistically find that elite members provide less extreme (polarized) opinions and exhibit

[2]https://www.yelp.com/dataset

lower acquisition bias.

Furthermore, to come up with an estimation model we presume that the star ratings provided by elite members are unbiased. Under this presumption we develop a forecasting model using supervised learning to predict unbiased ratings from ratings data of non-elite members (which is biased). We also use the text-data by non-elite members to enhance forecast accuracy. Our best performing prediction model has a predicted R-Squared of 84.58% as measured on the out-of-sample data (the test set).

Our work helps inform online platforms to take initiatives to come up with fresh mechanisms to compile more credible (less biased) data. Further, we illustrate ways in which they can use this unbiased data to extrapolate and predict unbiased ratings for products or services for which such a trustworthy source of data is limited or not available. An unbiased star rating by the online platforms would strengthen the experience for both the businesses and consumers as well as improve the review quality reputation for the platform.

1.2 Literature Review

We arrange the literature review section into three parts. First, we examine the monetary impact of online reviews. Previous studies have determined that online reviews significantly impact sales and revenues, and that is why getting an unbiased metric of perceived service quality is crucial for the businesses and consumers. Second, we review the literature, examining how acquisition bias and under-reporting bias creeps in because of self-selection. Finally, we will review a few studies which have utilized text reviews to have a richer insight using the information in them

to gauge the perceived quality of products and services.

1.2.1 Economic Impact of Online Reviews

Chintagunta et al. (2010) empirically determine that mean user ratings of movies on Yahoo Movies website has a significant positive impact on box office returns by the movie. Cui et al. (2012) shows that reviews of new electronic products on Amazon has a considerable impact on their sales. This impact is greater in the early launch period and declines in later periods. Hence they recommend encouraging quality reviews during the initial launch time. In hospitality literature, Ye et al. (2009) shows a substantial impact of online reviews on Chinese travel agency website Ctrip on hotel sales. Zhang et al. (2010) studies difference in impact of user-created reviews and expert-editor reviews from a famous consumer advice website in China, Dianping, on online popularity of restaurants as measured by the volume of web traffic incoming to restaurant webpages. Moe and Trusov (2011) notes that rating behavior of online consumers is significantly influenced by previously posted ratings on the online platform. They also find evidence that although these ratings can directly improve sales, the effects are somewhat short lived once we consider indirect effects. This gives hint on why early unbiased ratings if given by elite Yelp's elite members can determine future ratings of businesses and the bias. Finally, Luca (2016) show that a one star increase in a Yelp rating contributes to a 5 to 9% increase in the business' revenue. Considering this economic impact, it is imperative for platforms to present an impartial evaluation of the star ratings.

1.2.2 Acquisition and Under-Reporting Bias

Hu et al. (2017) found evidence of two types of self-selection biases that influence online review distributions. They call these biases *acquisition bias* and *under-reporting bias*, both of which render average online ratings a biased estimator of perceived product quality by the consumers. Acquisition bias exists because merely a small percentage of population will acquire the service and that sub-population would have a stronger perceived utility of using that service compared to other members of the population outside this sub-population. Since this sub-population is not a randomized sample from the full population, we will not get the genuine distribution of service quality. On top of that, in practice we observe that not all the members of this sub-population who acquire a service report their opinions online. This leads to another self-selection bias known as *under-reporting bias* which is also called as *polarization bias* in some research. Under-reporting bias arises because the users of an online platform leave an online review after weighing the tradeoff between the cost and benefit of doing so. The cost involved in writing an online review is not monetary, but it takes time and effort from the user's end. Since the users with more polarized negative or positive opinions perceive a higher psychological benefit from voicing their opinions online, it makes it easy for them to overpower the cost of effort in doing so. Moderately opinionated users are reluctant to write online reviews. Hence, online reviews contain more samples that convey extreme positive or negative opinions as compared to the moderate ones.

Hu et al. (2009) explain why products (or services) on online review platforms have J-shaped distribution of star ratings. Because of voluntary under-representation of moderate opinions on review sites, the reviews on such platforms follow a bimodal or J-shaped distribution. Polarized opinions received from vol-

untary contributions make this J-shaped pattern. The distribution from polarized ratings does not produce the actual underlying distribution of the quality of service. In their experimental study, Hu et al. (2009) found that the actual distribution for quality was almost a bell-shaped or normal distribution if the entire population report their opinions. In their controlled lab experiment they mostly collected moderate (2, 3 and 4-star) opinions. Further, there was almost an equal representation of positive and negative reviews. Therefore, experimental evidence suggests that online reviews present a distorted view of the underlying distribution of the quality of service.

Hu et al. (2006) shows that 53% of the products on Amazon have a bimodal distribution and hence their average rating does not portray the true quality of the products. They come up with a reason of this finding based on consumers' motivation (brag and moan) to leave an online review. Chamberlain and Smart (2017) shows that this bimodal distribution is present in many of the online review platforms, including Yelp and Glassdoor. They further find a statistically significant evidence that Glassdoor's give to get policy aided in reducing polarization bias in company ratings on their platform. They recommend review aggregation platforms to provide non-monetary economic motivations that can encourage moderate online reviews. Special incentives to promote more moderate users to share their opinions online can help the platforms to secure a more accurate distribution for perceived service quality by reducing under-reporting bias. For example, Chamberlain and Smart (2017) shows that Glassdoor's "give to get" policy helped in reducing the under-reporting bias on their website for company reviews on their platform. The "give to get" policy restricts the unrestricted access to the Glassdoor's content until a user also submits her own contribution. Koh et al. (2010) suggest that consumers' rating behaviors and the bias are affected by cultural in-

fluences. Their results show that the under-reporting bias is more prevalent in United States compared to China based on the data collected from IMDB.

1.2.3 Textual Data Analytics on Online reviews

Archak et al. (2011) illustrates how to use textual data present in Amazon reviews to understand consumers' relative preferences for different products' features. They also use text for predicting prospective changes in product sales. Shin et al. (2019) investigate the impact of textual concreteness of online reviews on the perception of information value of travel reviews. Textual concreteness captures how much concrete or abstract words are used in text. They compare impact of specific and objective facts in text, with the abstract or emotional content based on a subjective experience. They find that most travelers perceive abstract reviews to be more helpful. Korfiatis et al. (2019) studies knowledge gain by using topic modeling on review text data of airline passengers to further grasp customer satisfaction and service quality.

1.3 Dataset

For this study we use Yelp dataset[3]. The Yelp dataset is a subset of businesses, reviews, and user data from the Yelp database. This data is openly accessible for personal, scholarly, and academic purposes. This data consists of information from ten large metropolitan areas within United States and Canada. The Yelp dataset has a record of $1,637,138$ users out of which nearly 4.4%, i.e., $71,377$ users were

[3]https://www.yelp.com/dataset

part of elite-squad at-least once in the past. Yelp dataset contains 6,685,902 reviews out of which Yelp's elite members wrote 18.4%, i.e., 1,231,492 reviews over the past. The dataset associates each review on Yelp with an exclusive user-id which refers to the user who wrote the review, and a business-id which connects to the business for which the review was written.

Yelp has data of 192,609 businesses out of which nearly 30.8%, i.e., 59,371 are tagged as "Restaurants" or "Food". We use the Yelp tags, which describes the type of business, to classify the businesses into distinctive categories. Figure 1.1 shows top five categories in terms of number of businesses that fall into those categories in the Yelp dataset. A business can have multiple tags. To make a category, we combine similar tags such as "Food" and "Restaurants". For every business on Yelp, the dataset contains its name, location, average rating and other business specific attributes.

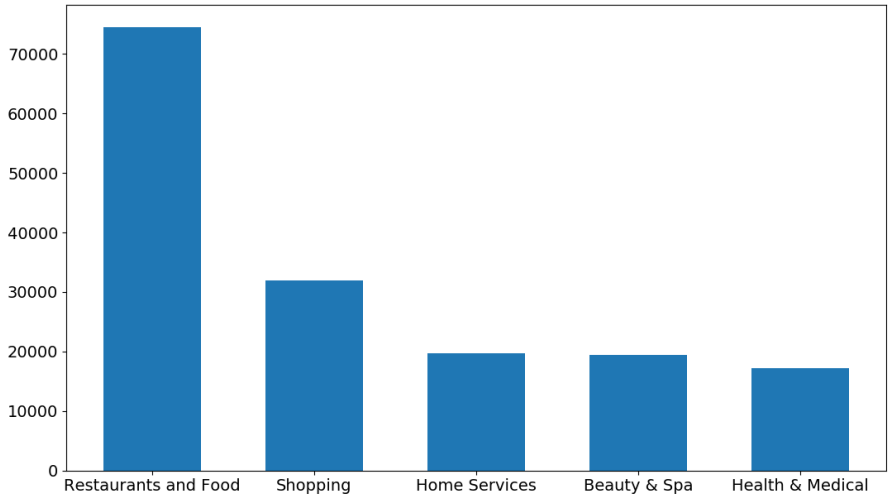

Figure 1.1: Top five business categories

From Table 1.1 we can see that we have 5928 businesses with at least 40 elite

Categories	**Avg # of elite**	**Num businesses with ≥ X elite**			
		X = 25	**X = 30**	**X = 35**	**X = 40**
All Categs	6.40	10,412	8,407	7,021	5,928
Food & Restaurant	12.68	9,046	7,335	6,150	5,219
Shopping	3.15	471	357	282	222
Home Services	0.73	28	19	17	12
Beauty & Spa	2.10	129	100	78	69
Health & Medical	1.10	24	17	13	10

Table 1.1: Summary for different categories in Yelp dataset

reviews. In machine learning it is usual to split the data into two sets: (i) *training data*: which we use to learn the parameters of our model and (ii) *test data*: which we will use to estimate the predictive performance of our models. Following this, we divide the data of the businesses with at least 40 elites into two categories. We use 80% of this data for training and 20% of this data for testing. We report all the prediction results in this paper on the unseen test data, unless stated otherwise.

1.3.1 Performance Metrics

We will use three performance metrics to test and compare our prediction models. We would calculate all the three performance metrics on the test dataset. Mathematically, let $\hat{y}$ denote the predicted value and y denote the actual value. Let N be the total number of test examples. Then the three performance metrics can be described as:

- Mean Squared Error (MSE) : It is calculated by taking the average of the squared error terms, where error is the difference between predicted and actual value (Lehmann and Casella 2006).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

- Predicted R-Squared : We calculate the R-Squared value on test data set and call it as predicted R-Squared. The predicted R-squared indicates how well a regression model predicts responses for new observations. Predicted R-squared is always lower than R-squared and it penalizes if the model is over-fitted.[4] Let $\bar{y}$ denote the average of all the actual values in our test data set. The the predicted R-Squared is defined as:

$$R - Squared = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2}$$

- Marginal Accuracy : We also define a metric for accuracy to compare the prediction of different models as follows. We first consider a margin $X \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. We define that the prediction for a particular sample point is accurate if $abs(\hat{y} - y) \leq X$, where abs denote the absolute value function. Then we calculate the percentage of data points which were correctly predicted for that margin to calculate the final accuracy. Mathematically, let the margin we chose be X. Let $1\{.\}$ denotes the indicator function, which returns 1 if the input is true, otherwise it returns 0. Then the accuracy is given by:

$$Accuracy = \frac{\sum_{i=1}^{N} 1\{abs(\hat{y}_i - y_i) \leq X\}}{N}$$

[4] https://statisticsbyjim.com/regression

CHAPTER 2

PRELIMINARY OBSERVATIONS

2.1 Acquisition Bias

Acquisition bias (Hu et al. 2017) is generated by self-selection of consumers to pick up a specific service if they have a stronger perceived utility of the service a-priori. Acquisition bias in businesses can arise because different users have different predilections for cuisines, brand, proximity to their place, etc. For example, only those customers which have a prior perceived higher utility from a spa will go there. Hence it is tough to get the accurate distribution because we do not have a random sample of overall population who picked up the service. Acquisition bias makes online ratings distribution right-skewed, as only those consumers gains a service who already think they will like it and such consumers are likely to give higher ratings than a random sub-population. We see acquisition bias as squeezing the probability mass distribution of the original ratings a bit towards the right side (more towards 3, 4, 5 star ratings). Acquisition bias makes the ratings distribution right skewed without changing the principal bell-shape of the distribution.

Non-elite members reduce their risk by not trying a new business outlet or minimize their effort by not visiting a business far away from their place. However, elite members gain an increasing utility by writing reviews for more businesses, which help them pursue their elite membership in the long run. We have information about $1,637,138$ users in our dataset. Out of these users, $71,377$ users have been part of the elite loyalty program for at least once in one of the preceding years. The remaining $1,565,761$ users have never been a part elite squad. To examine the difference in acquisition bias, we compare the number of distinct businesses

reviewed by the two groups of users on Yelp. Table 2.1 shows that elite members play an important role in exploring new businesses with a high uncertainty of quality. Elite members visit 21 distinct businesses on average and non-elite members just visit 3 distinct businesses on average. The difference between these averages is significant under t-test with a p-value of zero. To calculate p-values for comparing Std-Dev and Median, we use Levene test and Mood's median test, respectively. Non-elites help Yelp to gain reviews for a broader range of businesses and decrease acquisition bias in ratings for a business.

Group	**Number of Users**	**Mean**	**Std Dev**	**Median**
Elite Squad	71,377	21.19	54.73	5
Non-Elites	1,565,761	3.16	6.72	1
Overall	1,637,138	3.95	13.69	1
P-Values		0.00	0.00	0.00

Table 2.1: Number of distinct businesses reviewed : elites vs non-elites

To better investigate the reduction in acquisition bias, we also notice that business ratings produced by elite members are less skewed than ratings issued by non-elite members. To examine this, we shortlist businesses which have at least 40 reviews from elite members out of the total business. We have 5,928 such businesses. For each business we measured the total, mean, median, standard deviation and skewness of star ratings separately for elite members and non-elite members. We then average out these values across all the businesses for the two groups. We present the results in Table 2.2. The ratings by elite members have a skewness of 0.77, while the non-elite members have a significantly higher skewness 0.98. We further discover that median for elite members is 3.99 and median for non-elite members is 4.14. Half of the ratings by non-elites is above 4.14 which supports the intuition that their distribution is more skewed. We further observe that mean rating given by elite members is higher because neutral (2, 3 and 4 star)

ratings are under-reported by non-elites. It is appealing to watch that despite the larger mean the distribution for elite members is less shifted to the right compared to non-elites.

Group	**Num-Reviews**	**Mean**	**Std-Dev**	**Skewness**	**Median**
Elite Squad	95.31	3.88	0.90	−0.77	3.99
Non-Elites	318.94	3.78	1.24	−0.98	4.14
P-value	0.00	0.00	0.00	0.00	0.00

Table 2.2: Comparison of distribution of average business ratings

2.2 Under-Reporting Bias

Not all consumers who visit a business and pick up a service record an online review because of the cost in terms of time and struggle of writing a review. Consumers who want to boast about the service or who wish to criticize about it get a higher utility from writing an online review. This contributes to a J-shape or bi-modal distribution of online ratings. Unless positive and negative influence of this bias cancels out, the mean rating is not the correct impression of the service condition.

For the $5,928$ businesses with at least 40 ratings from the elite-members, we merge all the star ratings provided by elites and non-elites. We obtain a collection of roughly $565k$ ratings from the elite members and $1,890k$ ratings from non-elites. We show the distribution of these ratings in Figures 2.1 and 2.2. The figures illustrate that elite members do not display an under-reporting bias since their distribution is bell shaped, while it is J-shaped for non-elite members. For non-elite members, the distribution of online reviews is characterized by more than 50% of 5-star or 1-star reviews combined and just 35% of 3 and 4 stars reviews. In contrast, elite members have 60% of 3 and 4 star reviews. Since elites have

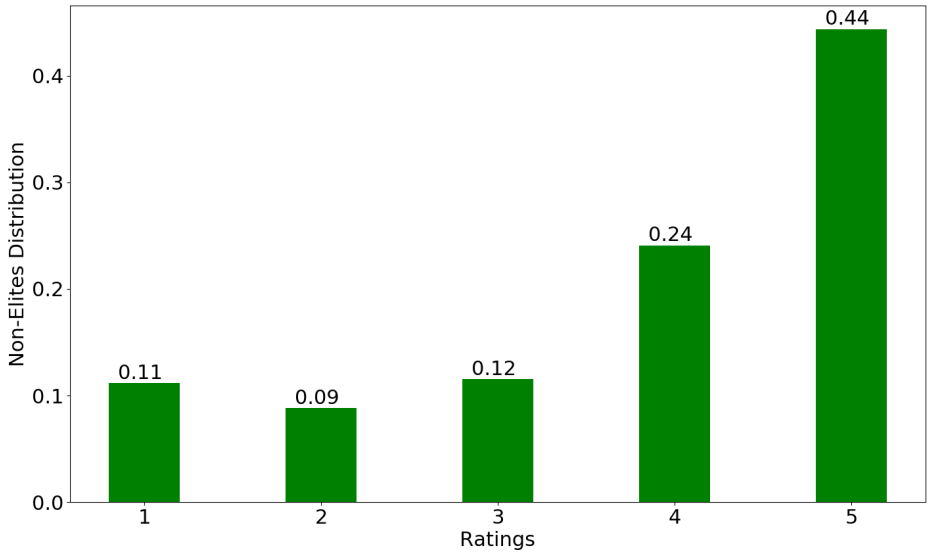

Figure 2.1: Non-elite ratings distribution

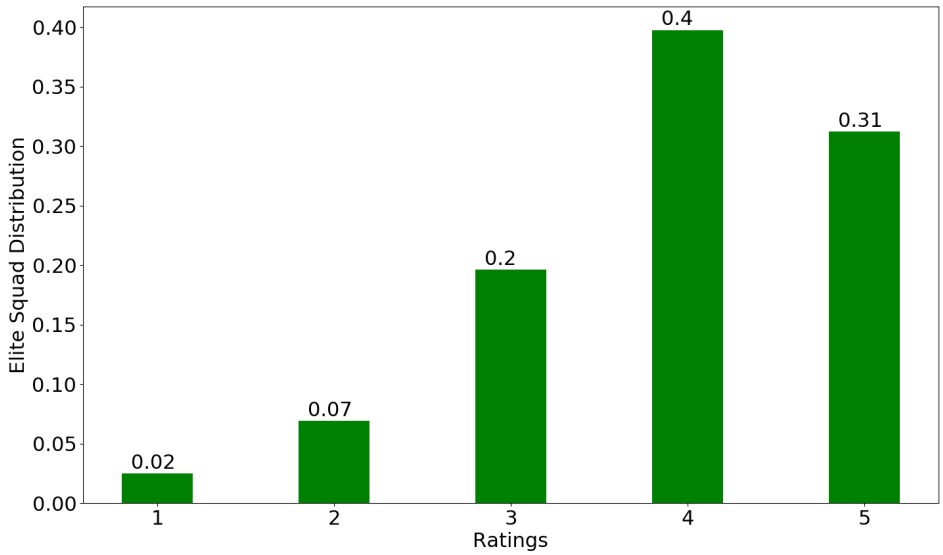

Figure 2.2: Elite ratings distribution

an increasing utility of recording a review online to defend their elite status, they write online reviews even if their judgments are moderate. To better measure under reporting bias, we use a more thorough procedure that studies how the elite user alters the probability of each 1-to-5 star ratings on Yelp. Using OLS, we estimate five "linear probability models" — one for each of the five star ratings. The table 2.3 shows the results of regressing a binary indicator for each star rating on an indicator for the review being written by an elite user along with accessible controls for users such as average user rating, total ratings given by user, total votes received by user, etc. We use 17 such controls. We can observe that probability of 1 star rating reduces by 5.5% and the probability of 5 star reduces by 12.7% . On the other hand, we can notice that probability of 3 star rating increases by 6.6% and probability of 4 star increases by 13.2%.

	1 star	2 star	3 star	4 star	5 star
Elite Review = 1	-5.51***%	-1.06%***	6.62%***	13.19%***	-12.76%***
P-Value	0.000	0.000	0.000	0.000	0.000
Num Controls	17	17	17	17	17
Adjusted R-Squared	0.062	0.067	0.136	0.291	0.476
Num Observations	2455301	2455301	2455301	2455301	2455301

Table 2.3: Linear Probability Model Results

CHAPTER 3

RESULTS

3.1 Relation Between Average Elite Ratings and Non-Elite Ratings

Existing research establishes that the mean rating for a product or service does not portray its true perceived quality because of numerous self-selection biases such as *acquisition bias* and *under-reporting* bias. In this section we will discover if there exists any relationship between the mean rating by non-elite members, which we consider being a biased measure of perceived quality, and the mean rating by elite members, which we consider being a true unbiased measure of perceived quality. In Figure 3.1, we have a scatter plot where each dot shows a data point conforming to a particular business and stands for the average rating by non-elite members and average rating by elite-members for that business. From the scatter plot, we observe that the relationship between average non-elite ratings and average elite ratings is linear. However, for any particular average non-elite rating, we can have many potential values of average elite rating. This reveals that average non-elite rating does not totally justify the variation in average elite-rating, and therefore there are other components which need to be examined to comprehensively justify this variation.

For better interpretation, we sketch an *identity line* in red in Figure 3.1. Each point on this line represents an equal value of average non-elite rating and average elite rating. Thus, the red line represents an optimal scenario if the non-elite ratings had not been biased. We note that most of the businesses which are rated

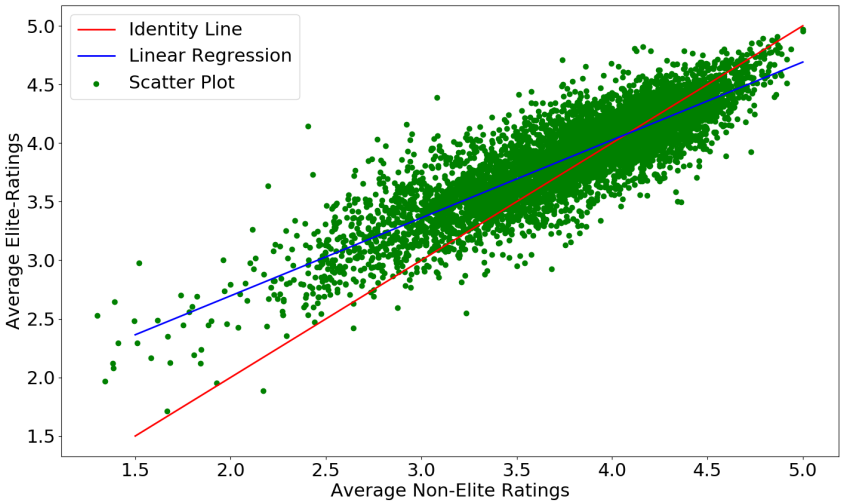

Figure 3.1: Comparison of average ratings by non-elites and elites

below an average of 3 by non-elites have higher star ratings by elites as most of the green dots lie above the red line. This suggests that the true rating for these businesses should be higher if there was no bias. However, for businesses with an average non-elite rating greater than 3, the unbiased average rating can be higher or lower than average non-elites' rating, but to ascertain that we require further information. To delve deeper into this, we pay attention to the distribution of ratings by elite members on Yelp. We have shown this distribution in Figure 2.2. We observe that elite members give just 2% of their overall ratings as *1-star* ratings and just 31% of their overall ratings as *5-star* ratings. On the other hand, Figure 2.1 shows that non-elite members over represent both the *1-star* and *5-star* ratings because of under-reporting bias.

We split the businesses into two groups. The first group included businesses for which the average elite rating was greater than the average non-elite rating, and the second group include businesses for which the average elite rating was less than the average non-elite rating. Then we compare the distribution of non-elite

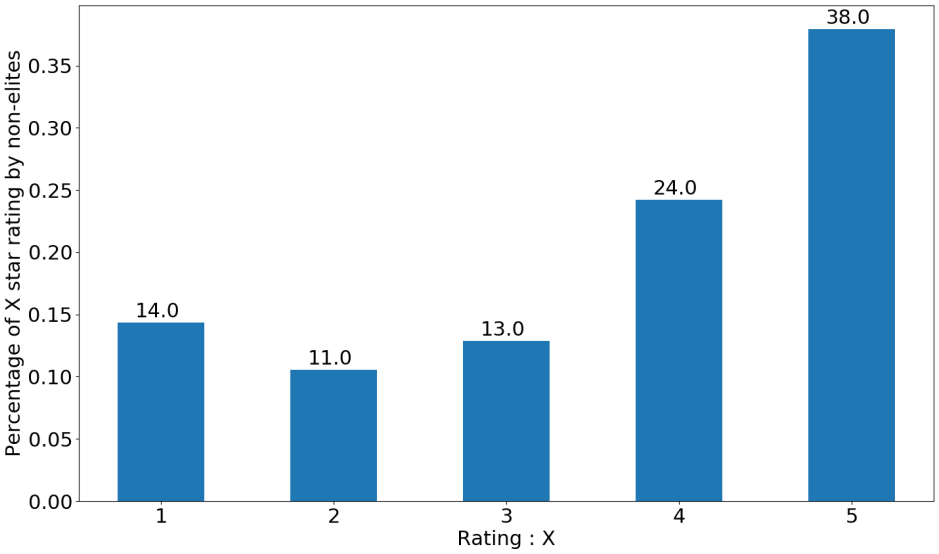

Figure 3.2: Businesses with higher average elite rating than average non-elite

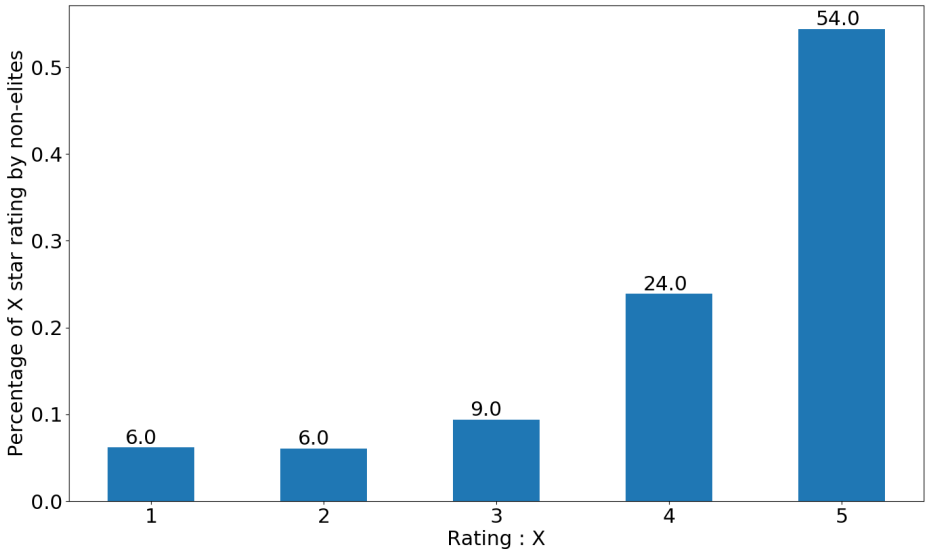

Figure 3.3: Businesses with lower average elite rating than average non-elite

ratings for the two groups of businesses. The distribution of non-elite ratings for these two groups is shown in Figure 2.1. We notice in Figure 3.2 that for the first group the number of *1-star* ratings are vastly over-represented (14%) compared to elite-members (2%). Hence, after removing under-reporting bias, the average elite ratings are higher for this group. For the second group, the number of *5-star* ratings are hugely over-represented (54%) compared to elite-members (31%). Hence, after removing under-reporting bias, the average elite ratings are smaller for this group. We note these conclusions in Table 3.1.

Group	**Non-Elite %**		**Elite %**		**Difference in %**	
	1-star	**5-star**	**1-star**	**5-star**	**1-star**	**5-star**
Higher Elite Avg	14	38	2	31	12	7
Lower Elite Avg	6	54	2	31	4	23

Table 3.1: Distribution of non-elites for businesses with higher elite rating vs lower elite rating

To study the relation between average non-elite rating and average elite rating quantitatively, we run a linear regression where independent variable is average non-elite rating and dependent variable is average elite-rating. The regression line is shown with blue color in Figure 3.1. The regression equation obtained is as follows:

$$EliteAvg = 0.66 * NonEliteAvg + 1.37 + \epsilon$$

From the regression equation, we see that the regression predicts a rating higher than identity line if the non-elite average lies below 4 and lower than identity line if the non-elite average lies above 4. Further, we compare the goodness of fit for the blue regression line and the red identity line by contrasting the Predicted R^2 values on our test dataset. We notice that blue line explains 76.83% of variance

while the red line explains just 57.00% of the variance. The *Mean Squared Error* (MSE) for the blue regression line is 0.0408 while the MSE for red identity line is 0.0757. We have specified these results in Table 3.2. A low value of R-Squared for identity line illustrates that non-elite average rating is not a good metric for true service quality. In the later sections we further raise the accuracy and get a better fitting prediction model for predicting average elite rating.

Model	R^2	MSE	Accuracy for Different Margin Values				
			0.1	0.2	0.3	0.4	0.5
Blue Regression Line	0.7683	0.0408	40.13	70.07	87.52	94.44	98.06
Red Identity Line	0.5700	0.0757	33.39	59.36	76.31	85.33	92.58

Table 3.2: Performance of regression and identity on test dataset

3.2 Adding Non-Elite Ratings Distribution to Regression

As we explained in the preceding section, average non-elite rating does not adequately predict unbiased average elite rating. We discovered that whether the average elite rating would be greater than the average non-elite rating or not depends on the distribution of non-elite rating. Hence, in this section we will investigate how does our model performance varies by introducing the distribution of non-elite star ratings as regressors in our regression. For this we determined the percentage of ratings out of overall ratings belonging to each of the five ratings. We run a linear regression with L2 regularization (also known as ridge regression) with the non-elite distribution as regressors. L2 regularization helps to deal with multi-collinearity. For the future, all of our linear regressions would use L2 regularization. We present the results of the regression on our test dataset in Table 3.3.

Regression Inputs	R^2	MSE	Accuracy for Different Margin Values				
			0.1	0.2	0.3	0.4	0.5
Non-Elite Average	0.7683	0.0408	40.13	70.07	87.52	94.44	98.06
Non-Elite Distribution	0.7814	0.0377	41.23	71.67	88.03	95.11	98.48

Table 3.3: Using non-elite distribution as regression inputs

We notice that the R^2 value has stepped up from 0.7683 to 0.7814 after employing the distribution of non-elite ratings in our linear regression model instead of simply using non-elite average. We obtain the following regression equation:

Input Variable	**Coefficient**
% of 1-star ratings	-0.86
% of 2-star ratings	-0.56
% of 3-star ratings	-0.61
% of 4-star ratings	0.57
%of 5-star ratings	1.46
Intercept	3.34

From the regression equation we notice that the intercept value is 3.34, which means that in the absence of any information, i.e., when values corresponding to all input coefficients is zero the predicted rating provided by the elite members to a business would be 3.34. We further notice that, let us assume % of 1-star ratings for non-elites is 100%. In that situation our predicted elite average would be $3.34 - 0.86 = 2.48$, which is much bigger than the non-elite rating, i.e., 1. Similarly, if we assume % of 5-star ratings for non-elites is 100%. In that case our predicted elite average would be $3.34 + 1.46 = 4.80$, which is lower than 5. We call this observation as *De-biasing Effect* of elites, where elites average ratings are bigger than non-elites average for low-rated restaurants and on the other hand elites average is smaller than non-elites if the non-elites average is high. We demonstrate this de-biasing effect in Figure 3.4 where we plot the adjusted rating if we assume that 100% of non-elite ratings were reduced at a single star rating. From the figure we also notice that the de-biasing effect is greater if the non-elites average is low

while it is much smaller if the average non-elite rating is high.

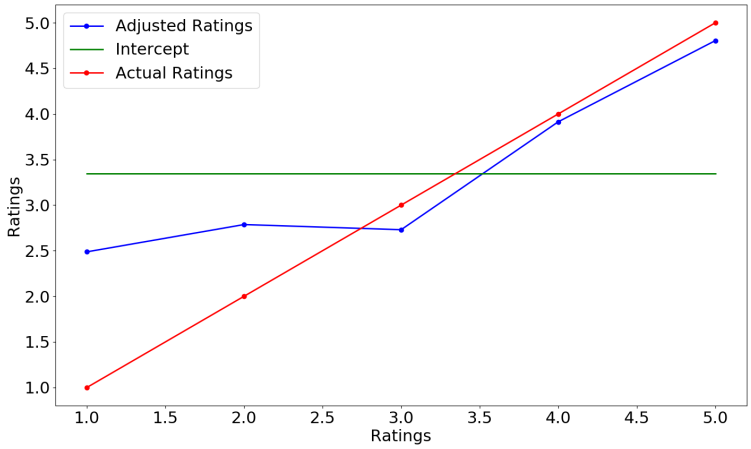

Figure 3.4: Debiasing effect of elites assumin all non-elite ratings are same

Yet Another Way to Capture Debiasing

We learned that only using non-elite average in our regression does not obtain satisfactory results. In one the previous sections we identified that the coefficient of the input variable *Non-Elite-Average* was positive, i.e., with a value of 0.66. This means that larger the non-elite average higher will be the elite average. However, this works against our de-biasing intuition as explained in the preceding section where we say that for higher non-elite averages the corresponding elite-average should be lower. This better explains the flaw in our initial formulation where we were taking only non-elite average as our input regressor. To deal with this problem, we decided to break down the average non-elite distribution into granular classes. We added 41 dummy variables as regressors into our model. These dummy variables correspond to intervals $[1-1.1), [1.1, 1.2), ..., [4.9, 5), \{5\}$, we call these intervals as 41 buckets. For a business we first measure the non-elite average

and identify in which of the intervals would this average lie. We thus make the entry of the dummy variable corresponding to that interval as 1 and rest of the dummy variables would be 0 for that business. We refer to these dummy variables as non-elite buckets for future reference. We use these dummy variables as our inputs for the linear regression. We draw the following results as recorded in Table 3.4. The table reveals that using dummy variables does not achieve better than using non-elite average as our input variable. However, adding dummy variables alongside with the non-elite distribution lead us the finest results so far. Further also including non-elite average alongside with the non-elite distribution and non-elite dummy variables does not produce any noteworthy enhancement in the ridge regression model performance.

Non-Elite Features	R^2	MSE	Accuracy for Different Margin Values				
			0.1	0.2	0.3	0.4	0.5
Buckets	0.7569	0.0419	39.38	69.14	86.00	94.69	98.23
Average (Avg)	0.7683	0.048	40.13	70.07	87.52	94.44	98.06
Distribution (Dist)	0.7814	0.0377	41.23	71.67	88.03	95.11	98.48
Dist+ Avg	0.7816	0.0377	40.89	71.67	87.86	95.11	98.40
Dist + Buckets	0.7863	0.0369	41.32	72.34	89.12	95.28	98.48
Dist + Avg + Buckets	0.7864	0.0368	40.64	72.68	89.12	95.19	98.57

Table 3.4: Using the buckets in the ridge regression

To better visualize the de-biasing effect of elites, we look at the ridge regression with entirely non-elite dummy variables (buckets) as regression inputs. We then plot the output elite-average obtained for all the non-elite averages from 0.4 to 5 at the intervals of 0.1 using the regression coefficients. We present this plot in Figure 3.5. Here also we can observe that elite average is higher than non-elite average initially, but as the non-elite average increases above a specific threshold the elite-average is lower than the non-elite average. This threshold is different in Figure 3.4 and Figure 3.5 because in Figure 3.4 we are predicting elite average

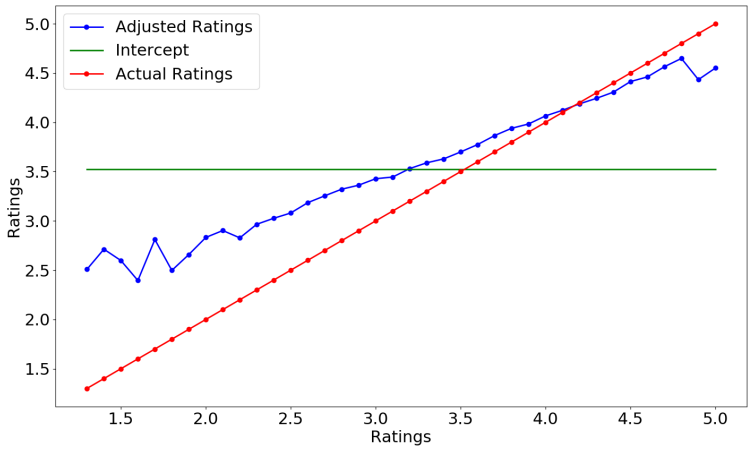

Figure 3.5: De-biasing effect of elites using non-elite buckets

based on non-elite distribution, i.e. by assuming that all the non-elite ratings are concentrated on a single star level, which is not practically appropriate. Figure 3.5 shows a more reasonable scenario where we are predicting elite average based on non-elite average and not by presuming that all the non-elite ratings are focused at a single value.

3.3 Adding Text Information : Tf-Idf

Term Frequency Inverse Document Frequency (Tf-Idf) helps to summarize unstructured text in form of a numerical vector based on how many times a word turns up in a document weighted by significance of that word in overall corpus (Rajaraman and Ullman (2011)). Let us consider that we have a set of documents, D, as our corpus. The method first determines a set of appropriate words, V, called vocabulary from this corpus D. Let the frequency of the word $w_j \in V$ in given document $d \in D$ is denoted by $f_{(w_j,d)}$. Then the term frequency $Tf(w_j, d)$

is defined as, $Tf(w_j, d) = \frac{f_{(w_j,d)}}{max_k\ f_{(w_k,d)}}$, where denominator represents a normalizing constant which amounts to the number of times the most frequent term in document d occurs. Let size of corpus D is N, i.e., we have a total of N documents. Let word $w_j \in V$ appears in $N_j \leq D$ documents. Then inverse document frequency for word w_j is given by, $Idf(w_j) = log(N/N_j)$. The term N_j serves as the weight for word w_j based on the 'uniqueness' of the word w_j in the corpus. If w_j is a rare word and appears in few documents, then $Idf(w_j)$ would be higher and vice versa. The idea behind this approach is that a word which appears in too many documents might be less pertinent as a discriminating feature for machine learning algorithms. On the contrary, a word which is more peculiar to only specific documents can help us identify these documents with more certainty. Then to vectorize a document $d \in D$ we calculate the product of term frequency and inverse document frequency, i.e., $TfIdf(w_j, d) = Tf(w_j, d) * Idf(w_j)$ for all the words in vocabulary, i.e., $\forall w_j \in V$. Thus, this approach simultaneously takes into account the number of times a word appears in a document and how important that word is as a discriminating factor with respect to other documents in the corpus.

Regression Inputs	R^2	MSE	Accuracy for Different Margin Values				
			0.1	0.2	0.3	0.4	0.5
TfIdf	0.7537	0.0434	41.23	71.84	86.68	94.10	97.55
Dummy	0.7569	0.0419	39.38	69.14	86.00	94.69	98.23
Avg	0.7683	0.0408	40.13	70.07	87.52	94.44	98.06
Dist	0.7814	0.0377	41.23	71.67	88.03	95.11	98.48
Dist, Avg, Dummy	0.7864	0.0368	40.64	72.68	89.12	95.19	98.57
Sub Dist, Avg, Dummy	0.8047	0.0344	42.92	75.13	90.22	95.87	98.57
Dist,Avg,Dummy,TfIdf	0.8433	0.0276	48.65	79.17	92.66	97.47	99.16

Table 3.5: Adding text information to regression

For further enhancement of our model, we again sought to utilize the text data present on the Yelp platform. For each business we concatenate all the text reviews received by the business from non-elite members on Yelp. Then using this corpus

of reviews we determined a vocabulary of size $15,374$ to construct a vectorized representation of text data for every business using Tf-Idf algorithm. We display the results for using text data as our regressors in Table 3.5. From the table we can notice that the non-elite distribution is the most powerful feature to predict average elite rating, followed by non-elite average and non-elite dummy variables. On the other hand, applying just the Tf-Idf vector from text comes last in this list as it merely explains 75.37% of variance in our dependent variable. However, using all four features together gives us the finest performing model so far. The MSE of of this model is 0.0276 and it explains 84.33% variance of the dependent variable which has been a substantial improvement from our previous best model with just an R^2 of 78.64%.

3.4 Pushing Boundaries with Machine Learning

In this section we will examine if we can further add to our model performance and interpretation by applying non-linear machine learning models. We will compare the relative performance of the ridge regression model with several machine learning models. All the models employed in this section have been better described in Appendix B.

3.4.1 Bagging with Linear Regression

Bagging, also known as bootstrap (Tibshirani and Efron (1993)) aggregation, is one of the powerful ensemble methods in machine learning. Ensemble methods use different learning algorithms or multiple models with same algorithms to obtain

a stronger prediction result. In bagging we first appoint a base regressor, i.e., a machine learning algorithm that we will adopt for each of our estimators. Then we select the number of independent estimators for bagging. Each independent estimator then fits a regression model on random subsets of the original dataset where random subsets are collected with replacement. Finally, the results of all the estimators are aggregated by averaging the results of independent estimators to obtain the final prediction. Such an ensemble model is beneficial over using a single estimator as it helps to reduce the variance of the overall learning model. As we learned that, for our case relation between the independent and dependent variables is intrinsically linear, we prefer our base estimator to be a linear regression model for our bagging algorithm. We present are results for three different values of the parameter for the number of estimators for robustness. We detail the results in Table 3.6. We notice that bagging has a bare improvement over running a single estimator.

Regression Model	R^2	MSE	Accuracy for Different Margin Values				
			0.1	0.2	0.3	0.4	0.5
Linear Regression	0.8433	0.0276	48.65	79.17	92.66	97.47	99.16
Bagging (n=10)	0.8435	0.0276	48.90	79.93	92.66	97.47	99.16
Bagging (n=100)	0.8445	0.0274	49.07	79.51	92.83	97.47	99.16
Bagging (n=200)	0.8448	0.0273	49.41	79.51	92.92	97.47	99.16

Table 3.6: Comparison of bagging with linear regression

3.4.2 Random Forest Regression

Random Forest (Geron 2019) is another bagging algorithm where the base estimator is a Decision Tree. To learn further about decision trees, please refer to appendix section B.2. Random forest fits several decision trees on different subsets

of the data. Table 3.7 shows that without the use of text features, the random forest performs only marginally better than the linear model. This further reinforces our hunch that the relation between the dependent variable and the independent variable is intrinsically linear in our case, since non-linear models are a little helpful in boosting model performance.

Regression Model (Dist + Avg + Dummy)	R^2	MSE	Accuracy for Different Margin Values 0.1	0.2	0.3	0.4	0.5
Linear Regression	0.7864	0.0368	40.64	72.68	89.12	95.19	98.57
Random Forest(n=2000)	0.7928	0.0365	41.74	72.51	89.04	96.04	98.48

Table 3.7: Random forest comparison with linear regression without text

To delve further into the random forest algorithm we restricted the height of the decision trees up-to level 4. Then we picked a decision tree estimator from random forest and plotted its binary tree structure which explains the flow-chart about how each internal node performs a condition on one of the input-features based on which the data points are divided at that node of the tree. Finally, each leaf node of the tree shows the final prediction. We have shown the decision tree in Figure 3.6. We notice that the algorithm bases the first split on Average Non-Elite rating (NE-Avg). If the NE-Avg is greater than 3.7 then we branch to the right side of the tree otherwise we branch to the left. This aligns with our instinct that for the higher NE-Avg ratings the de-biasing by non-elites brings lowers the rating given by non-elites and for businesses with lower NE-Avg ratings the de-biasing by non-elites increases the actual rating of business. This de-biasing effect is more clear if we look at Level-3 nodes in the tree. The first node of Level-3 shows that for all the businesses with NE-Avg less than 2.34 the predicted elite average should be 2.7. Similarly last node of Level-3 represents that whenever the NE-Avg is greater than 4.75 the elite average should be equal to 4.70. Thus we can observe that nodes to the right of root nodes tend to bring down the NE-Avg, and nodes

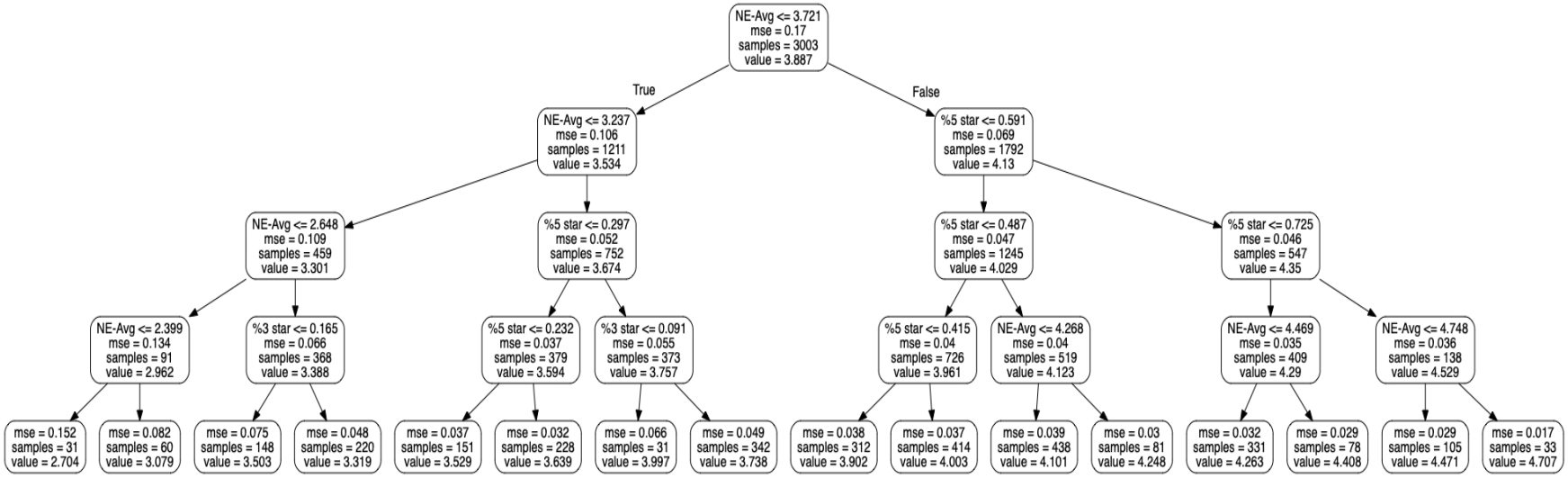

Figure 3.6: Comparison of average ratings by non-elites and elites

to the left of the root node tend to bring up the NE-Avg ratings. This observation aligns well with our de-biasing effect of elites theory. After the NE-Avg, as we can notice from the tree that the next important variable in deciding the bias is the percentage of 5-star ratings. If the percentage of 5-star ratings is higher, the predicted elite average rating should be higher too, which means the right child gets a higher prediction label than the left child at such a node. However, for the node labeled percentage of 3 star ratings, we can observe that a higher percentage of 3-star ratings leads to lower prediction label compared to parent node and vice versa. This shows that too many 3-star ratings by non-elites is a negative indicator for the business.

However, if we also include the text features into our model, the random forest could not beat our linear model. This is because, as we have previously observed, that underlying relationship between dependent and independent variables is intrinsically linear. Also, when there are many features with high noise and low useful signals, random forests find it difficult to model linear combinations of such a large number of features. We note the performance comparison between random forest and linear model in Table 3.8. For robustness check, we measure results with two different number of estimators.

Regression Model	R^2	MSE	Accuracy for Different Margin Values				
			0.1	0.2	0.3	0.4	0.5
Linear Regression	0.8433	0.0276	48.65	79.17	92.66	97.47	99.16
Random Forest(n=100)	0.8301	0.0299	46.21	78.16	92.66	97.47	99.16
Random Forest(n=200)	0.8318	0.0296	46.80	77.82	92.24	97.47	99.24

Table 3.8: Random forest comparison with linear regression with text

To delve deeper into the text data analysis, we glance at the relative feature importance of input features using the random forest model. We show the feature importance for top 10 features of random forest model in Table 3.9. As consistent

Input Variable	Importance (In %age)
NE-Avg	46.04
% of 5-star	28.56
ordered	0.40
% of 4-star	0.30
%of 3-star	0.29
good	0.25
decent	0.24
bad	0.16
tasted	0.15
place	0.14

Table 3.9: Relative feature importance

with the tree diagram we can see that the most important features are NE-Avg (46.04%) and percentage of 5-star reviews (28.56%) by non-elites. All other features serve a little fraction of less than half percent in terms of feature importance. This is in align with our intuition that we have too many input variables in our data-set but most of them are noisy, and hence linear regression model performs better than a random forest regression.

3.4.3 Support Vector Regression

Support vector regression (Cortes and Vapnik (1995)) seeks to fit the model such that maximum data points can remain inside a restricted window size of the predicted output. For points that exist outside that window, SVR penalizes them in its loss function. To learn more about SVR please refer to appendix section B.5. Table 3.10 compares performance of SVR with linear regression. Although vanilla SVR does not perform better because of the large size of the input space, cutting down the dimension of input space using Principal Component Analysis (Grey 1981) gave marginally better results than linear regression.

Regression Model	R^2	MSE	Accuracy for Different Margin Values				
			0.1	0.2	0.3	0.4	0.5
Linear Regression	0.8433	0.0276	48.65	79.17	92.66	97.47	99.16
SVR(eps=.01,C=1.5)	0.8199	0.0317	46.12	77.15	91.99	96.37	98.82
SVR(pca=1k,eps=.01,C=1.5)	0.8439	0.0275	48.74	78.67	93.34	97.13	99.41

Table 3.10: SVR comparison with linear regression

3.4.4 Gradient Boosted Decision Trees

Gradient boosting comes under the umbrella of boosting (Freund and Schapire (1995)) techniques. To learn further about boosting refer to appendix B.3. Gradient Boosting tries to minimize the residual errors, i.e., the difference between the predicted and actual values. In Gradient Boosting we try to fit the new weak learner to the residual errors generated by the previous learner. Table 3.11 compare gradient boosting with linear regression.

Regression Model	R^2	MSE	Accuracy for Different Margin Values				
			0.1	0.2	0.3	0.4	0.5
Linear Regression	0.8433	0.0276	48.65	79.17	92.66	97.47	99.16
Gradient Boosting(n=100)	0.8370	0.0287	47.47	77.40	92.66	97.39	98.99

Table 3.11: Gradient boosted trees comparison with linear regression

3.4.5 Neural Networks

We employ a neural network (Ripley (2007)) with two hidden layers of size 65 and 10 respectively. We apply the 'adam' optimization method for training. We also use an L2 regularization with $\alpha = 0.25$. To learn further about neural networks refer appendix B.6. Table 3.12 compare the performance of our neural network model with linear regression.

Regression Model	R^2	MSE	Accuracy for Different Margin Values				
			0.1	0.2	0.3	0.4	0.5
Linear Regression	0.8433	0.0276	48.65	79.17	92.66	97.47	99.16
Neural Networks	0.8458	0.0271	49.49	79.76	93.17	97.30	99.24

Table 3.12: Neural networks comparison with linear regression

Summary for ML models

Table 3.13 summarize the comparison of all the ML models with linear regression.

Model	R^2	MSE	Accuracy for Different Margin Values				
			0.1	0.2	0.3	0.4	0.5
Random Forest	0.8318	0.0296	46.80	77.82	92.24	97.47	99.24
Gradient Boosting	0.8370	0.0287	47.47	77.40	92.66	97.39	98.99
Linear Regression	0.8433	0.0276	48.65	79.17	92.66	97.47	99.16
SVR	0.8439	0.0275	48.74	78.67	93.34	97.13	99.41
Bagging Linear Reg	0.8448	0.0273	49.41	79.51	92.92	97.47	99.16
Neural Networks	0.8458	0.0271	49.49	79.76	93.17	97.30	99.24

Table 3.13: Comparison between all machine learning models

3.5 Trade-Off: Heterogeneity and Amount of Data

For robustness we examine how do the results differ if we alter the threshold of the minimum number of elites required for businesses to be incorporated in our dataset. All the results so far have been obtained by setting this threshold to be 40, i.e., we are including all those businesses with at least 40 elite reviews.

We observe that as we increase this threshold, we will have a lesser number of businesses in our dataset, as fewer businesses would meet that criteria. Since the number of businesses in our dataset decrease, the variance of average elite ratings of this dataset should increase. However, on the contrary, we observe that the

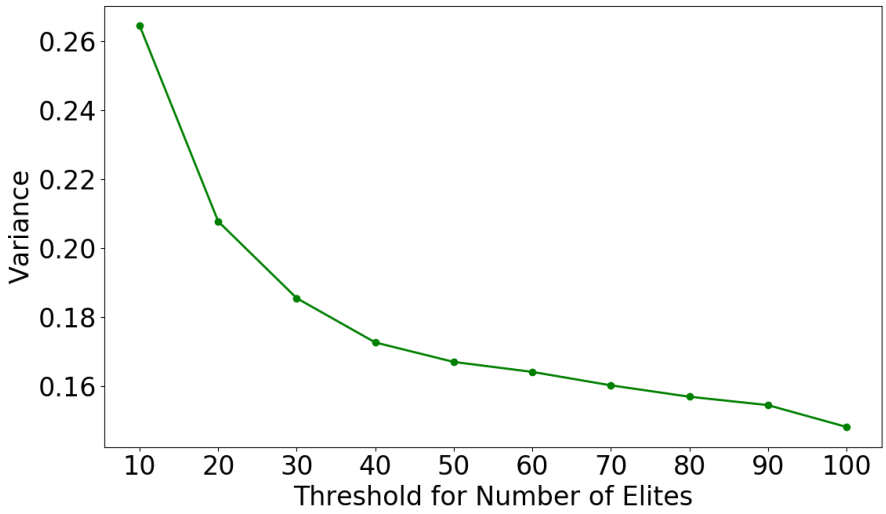

Figure 3.7: Comparison of average ratings by non-elites and elites

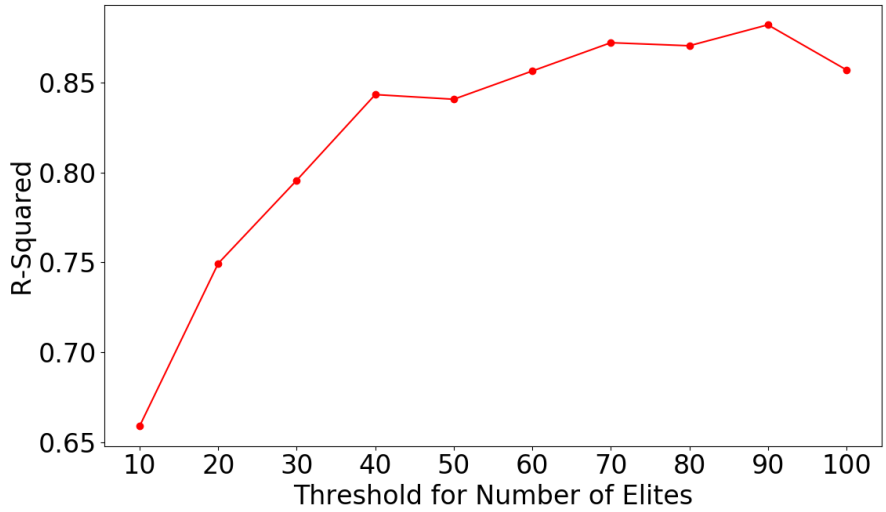

Figure 3.8: Comparison of average ratings by non-elites and elites

variance of the average elite ratings further decrease on increasing the threshold. Figure 3.7 displays the variance of average elite ratings for dataset obtained using various threshold values from 10 to 100 in the intervals of 10. This shows that the businesses included in the dataset become more similar to each other as we

increase the threshold, and hence the average elite ratings are also similar to each other for these restaurants. Since the variance of dependent variable decrease on increasing the threshold, the R-squared improves, and we get a better fit. Figure 3.8 shows the R-squared values for 10 different values of threshold from 10 to 100. We can see that R-squared value increases if we increase the threshold value.

CHAPTER 4

CONCLUSION

Online review platforms suffer from acquisition and polarization bias. Yelp's elite membership program is treating the platform to reduce this bias, and elite members do not suffer from these biases. However, for comparatively new businesses or the businesses which live in areas where concentration of elites is not satisfactory, it is challenging to obtain the unbiased metric for perceived service quality. We examine how the elites help in cutting down the two kinds of bias. Elite members de-bias the ratings provided by non-elite members by increasing the average rating if the non-elite average is too low, i.e., the business has too many 1-star reviews, or by decreasing the average rating if the non-elite average is too high, i.e., the business has many 5-star reviews. We then build various models to predict the unbiased average elite rating using the ratings and text data of non-elite members for a given particular business. We observe a compelling performance improvement in our models if we include text data as our inputs. Further, we also explore the relative importance of input features using random forest regression model. We found the NE-Avg and percentage of 5-star reviews by non elite members are the most important features to predict the average elite rating, which aligns with our intuition that besides the non-elite average distribution of non elite ratings play a vital role in determining elite average. For the businesses with a threshold of having at-least 40 elite reviews, our best model got the predicted R^2 value of 84.58%. We also observed that as we raise the threshold for the minimum number of elites, the businesses in our dataset have less variance and hence the model performance increases. Thus we present a mechanism to de-bias online ratings with the help of predictive models, which can enable review platforms to display more accurate average star ratings and help businesses and customers on their platform.

APPENDIX A

YELP'S ELITE PROGRAM

The Yelp's elite squad is an yearly loyalty program. Every year Yelp accepts nominations from its users to select members for Yelp's elite squad. The criteria for selection include a historical record of good quality reviews, a detailed and completed user's personal profile, and an active engagement (voting and complimenting record) by a user over the previous year. The elite badges are only valid for one year and existing members have to re-nominate themselves every year. Those who violate Yelp guidelines and code of conduct may get their membership revoked earlier than the designated period. Yelp designates members of the elite squad an elite badge on their account profile. After maintaining elite membership for five years, users receive the gold elite badge, and after ten years users receive the Yelp's coveted black elite badge. Apart from the elite badges, members of the elite squad can attend Yelp's exclusive events for elite members which are organised by different businesses for their promotion by contacting one of the Yelp's community manager. Organising such events help local businesses build a relationship with the Yelp and its elite members so that they can get more attention and reviews on the Yelp platform. They also enjoy a higher viewership of their reviews as Yelp prefers to display elite reviews by elite members on top of other reviews.

APPENDIX B

ML ALGORITHMS

B.1 Decision Trees

Breiman et al. (1984) introduced a comprehensive text on classification and regression trees. Decision tree models the classification or regression problem in the form of a binary tree. The decision tree picks an input feature at every node of binary tree and picks a threshold for that input feature. It divides all the points in the dataset into two parts, the right half and the left half based on this threshold. Thus it keeps on breaking down the dataset into smaller subsets as the depth of the tree increases. The leaf nodes of a decision tree represent the prediction value obtained by the algorithm. For any input for which we want to predict the label, we can follow the threshold criteria that our input satisfies at each step from the root till we reach a leaf node and the label for that leaf node will be our prediction.

Deciding Input Features to Split

The decision tree picks up an input feature so that overall entropy decreases. The entropy(H) is a measurement of randomness or disorder. Let we have N labels in our dataset. Then the entropy of a sample S is defined as,

$$H(S) = -\sum_{i=1}^{N} p_i log_2(p_i)$$

,where p_i represents the probability of i^{th} label in our sample, i.e., $p_i = \frac{n_i}{n_{total}}$, where n_i is the number of data points with label i and n_{total} represents the total

number of data points in our sample. Let us have only two labels in our sample, and the probability of first label is p. Then the probability of the second label will be $1-p$ and the entropy would be defined as, $H(S) = -plog_2(p) - (1-p)log_2(1-p)$. We note that if we have an equal number of labels then $p = 1-p = 0.5$ and $Entropy = 1$. On the other hand if we have only a single label then the *Entropy* would be zero. The variation of *Entropy* as the value of p varies from 0 to 1 is shown in Figure [1] B.1.

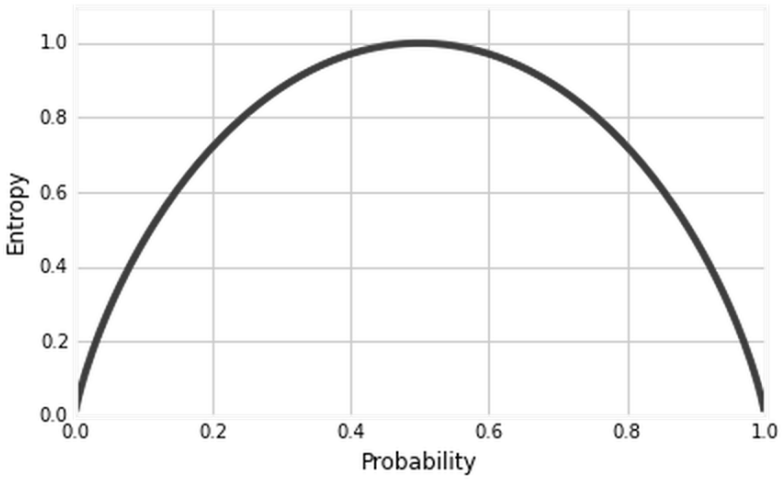

Figure B.1: Entropy variation with probability of first label

Let we split the sample S based on the value of an input feature X. The decrease in entropy is defined as *Information Gain (IG)* and we define it as follows,

$$IG(S, X) = H(S) - H(S|X)$$

,where $IG(S, X)$ represents *Information Gain* when we split sample S based on input feature X, $H(S)$ represents the entropy of sample S before the split and $H(S|X)$ represents the conditional entropy if we know the value of X, i.e., after

[1]Figure borrowed from: https://www.kdnuggets.com/2020/02/decision-tree-intuition.html

getting more information about X and making the split. The greater the reduction in entropy or uncertainty means we gain more information from that feature. At every node of the tree we chose the input feature which will give the maximum information gain for the data points which are part of that node. We usually stop when the entropy becomes zero for that node, i.e., the node contains all the data points with the same label. Thus the leaf usually contains all data points of same label in a decision tree. Since decision trees are prone to over-fitting, we sometimes limit the depth of a decision tree so that our model does not stop at leaves with zero entropy. This helps the model to avoid over-fitting on the outliers in the dataset and focus on a more generic learning.

B.2 Bagging

Tibshirani and Efron (1993) introduced initial ideas for bootstrap in their book. Bootstrap aggregating, also called bagging, is a machine learning approach where we train independent estimators on randomly sampled subsets of dataset, where samples are chosen with replacement. This algorithm falls under the umbrella of ensemble algorithms, and it helps to improve the stability and accuracy of machine learning classification and regression algorithms. Bagging reduces variance of the overall model and helps to avoid over-fitting. Bagging is popular for decision tree methods, but we can use it with any other machine learning algorithm.

Random Forests

Random Forests is a bagging method where we train multiple independent decision trees on randomly selected subsets of the original dataset where random subsets are taken with replacement for each decision tree. Finally, the result of all the decision trees is then aggregated by averaging the predictions to get the final prediction. Random Forests helps to avoid over-fitting, which is a common problem when we train decision trees. Figure B.2 borrowed from Verikas et al. (2016) explains the working for random forests. To read further please refer Géron (2019).

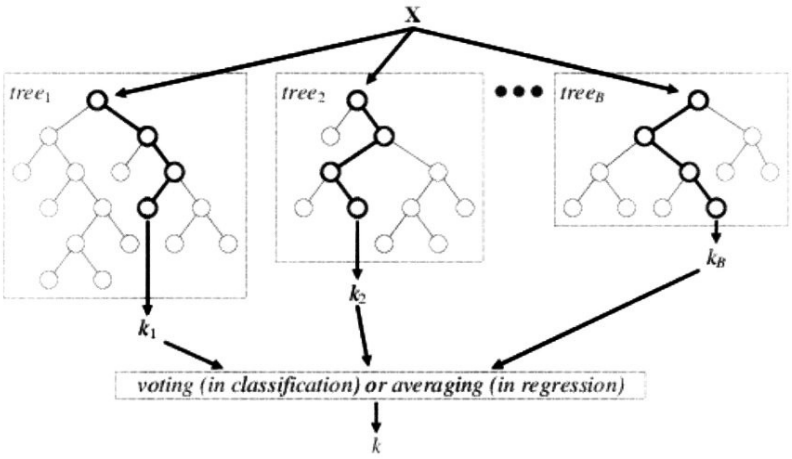

Figure B.2: Working of random forests

B.3 Boosting

Boosting was introduced by Freund and Schapire (1995). Boosting is another popular ensemble algorithm apart from bagging. Unlike bagging, 'Boosting' combines multiple weak learners to form a strong learner. We define a weak learner as a machine learning model that is slightly better than generating random prediction outputs, and the prediction outputs are only slightly correlated with the true val-

ues. We start with training a weak learner, then we train a new weak learner that tries to correct the mistakes of previous learner and this continues as each weak learner tries to correct its predecessor. Boosting helps in reducing both bias and variance for supervised learning, i.e., classification and regression problems.

B.4 Principal Component Analysis

Grey (1981) derives PCA in their book using maximum orthogonal variance. Principal component analysis is a dimensionality reduction technique that can help to find the important features for training of machine learning algorithms. PCA selects important features and removes the noise in our input data. PCA combines our input variables to form new variables that are all independent of each another. We call this as an orthogonal transformation of input features. The new linearly uncorrelated variables are called principal components. PCA is defined in such a way that the first feature or principal component has the largest variance (finds a direction of maximum variance in the hyperplane defined by input features) and each succeeding principal component recursively tries to maximize the variance under the constraint that it is orthogonal to the previous principal component. PCA is sensitive to the relative scaling of the original variables.

Suppose we have samples $X_1, X_2, ..., X_m$ from a random distribution $X \in R^n$. To apply PCA, we first subtract the mean of the distribution μ from each of the X_i so we can assume that the distribution is centered around origin. Now in the n-dimensional hyperplane we wish to find a unit direction w such that the variance of the projected points on w will be maximum. Mathematically,

$$\max_{w} \; var(w^T(X_i - \mu)) \; \forall i \in \{1, 2, ..., m\} \; s.t. \; || \; w \; ||= 1$$

$$i.e., \max_{w} \; var(w^T \bar{X}) \; s.t. \; || \; w \; ||= 1$$

,where $\bar{X}$ is a $n * m$ matrix such that i^{th} column of $\bar{X}$ equals $X_i - \mu$. Now $var(w^T \bar{X}) = E((w^T \bar{X})^2) - E(w^T \bar{X})^2$. But since $E(\bar{X}) = 0$, it implies that $E(w^T \bar{X}) = 0$. Hence we get, $var(w^T \bar{X}) = E((w^T \bar{X})^2) = E((w^T \bar{X}) * (w^T \bar{X})^T) = E(w^T \bar{X} \bar{X}^T w)$. We know that $\bar{X} \bar{X}^T = C$ (say), is know as var-covari matrix for matrix $\bar{X}$. Thus the problem reduces to:

$$\max_{w} \; w^T C w \; s.t. \; || \; w \; ||= 1$$

Now since C is a var-covar matrix it is a real and symmetric matrix of size $n * n$. Thus C is orthogonally diagonizable, i.e., it can be expressed in the form of $C = UDU^T$, where U is a orthogonal matrix consisting of eigenvectors of C and $D = diag(\lambda_1, \lambda_2, ...\lambda_n)$ is a diagonal matrix with eigenvalues as the diagonal entries. Then $w^T C w = w^T U D U^T w = v^T D v = \sum_{i=1}^{n} \lambda_i v_i^2$, where $U^T w = v$. To maximize this we can clearly see that $v_1 = 1$ and $v_2, v_3, ..., v_n = 0$, since λ_1 is the maximum eigen value among all the eigen values. Thus, the principal components are the eigen-vectors of the var-covar matrix of $\bar{X}$ taken in decreasing order of eigen values.

B.5 Support Vector Regression and Kernels

SVMs were first introduced by Cortes and Vapnik (1995). Support Vector Machines are models used for of supervised learning tasks. The SVM regression algo-

rithm finds a hyperplane in feature space such that maximum points can lie inside a boundary of fixed size. The goal is to find a function $f(X)$ s.t. $||y - f(X)|| \leq \epsilon$, where X represents input features and y represents true values to be predicted. All the points that lie inside the boundary contribute a zero loss to the loss function. Only the points which lie outside the boundary contribute towards the loss function. Thus given a fixed size band support vector regression tries to find the $f(X)$ such that the loss can be minimized for points lying outside the boundary. Figure B.3 borrowed from Lahiri and Ghanta (2008) explains the working of support vector regression.

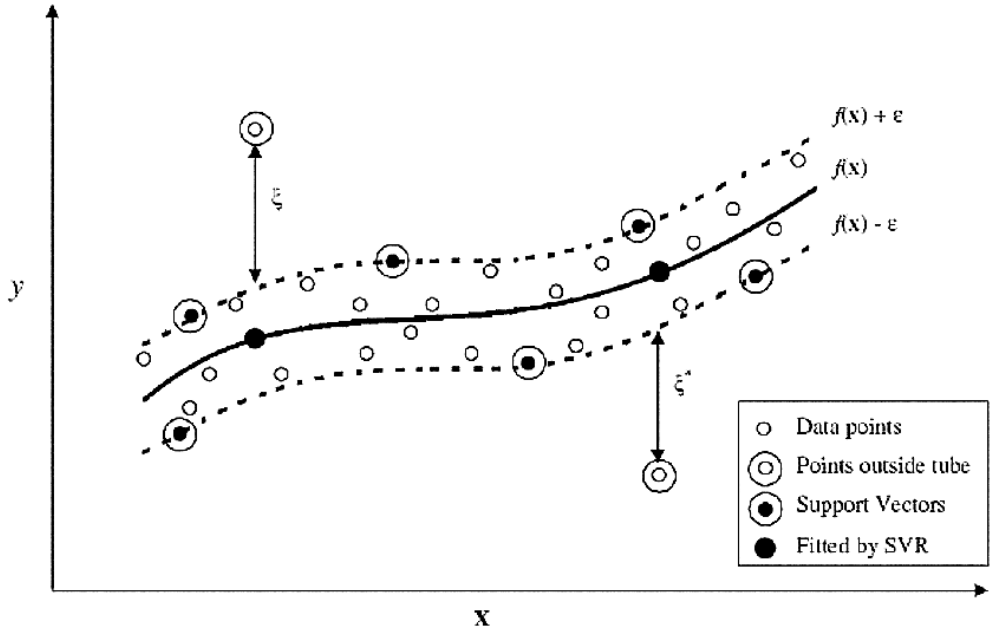

Figure B.3: Support vector regression

Kernels in SVM

Schölkopf et al. (2002) introduced learning with kernels. If the data points are not linear, we can project the data points into a higher dimensional space so that we can fit a hyper plane. Kernels help us get the same results which we would get

if we project the data points into a higher dimensional plane without projecting the data points into a higher dimensional place. With the help of kernels we can map the input feature vectors X into the higher dimensional space, also known as kernel space using the kernel transformation $\phi(X)$. We can reformulate the SVM optimization problem in it's dual form such that it only depends on the dot product of the input vectors corresponding to different training examples. Now a kernel function K is such that it satisfies the following property:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

,where $K(x_i, x_j)$ is called as a kernel of data points x_i and x_j and denotes the dot product of these data points after projecting them into the higher dimensional space defined by the corresponding kernel transformation ϕ. $K(x_i, x_j)$ is usually (not necessarily) of the form $K(x_i^T x_j)$, i.e., kernel function is applied after taking dot product of original feature vectors. Thus instead of first projecting the data points x_i and x_j into a higher dimensional space and then taking a dot product, we first take a dot product and then apply the K function on that dot product. One of the most famous kernel functions used with SVMs is *Radial Basis Function (RBF)* defined as $K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2),\ \gamma > 0$, and the corresponding transformation ϕ would project the input data points into an infinite dimensional space, which although cannot be stored in memory but still can be used to define the SVM optimization problem with the help of RBF kernel.

B.6 Neural Networks

Neural networks are motivated by the functioning of the brain. Brain releases electrical impulses which transfer through the inter-connected neurons. Ripley (2007) provide a comprehensive introduction to neural networks. Neural networks mimic human brain as it consists of the artificial neurons, which linearly combines its inputs (incoming signals) to produce the output (outgoing signal). This linear combination is a weighted linear combination of the inputs. Every neuron takes a weighted average of its inputs, pass this average to an activation function and send this output to next neuron. An activation function is a non-linear function so that the neural network can learn non-linear relations between the inputs and desired output. If neurons do not use any activation function, then the output of the neural network will be a linear function of inputs, which would be as good as a simple linear regression. In practice ReLU is the most widely used activation function and is defined as:

$$ReLU(x) = \begin{cases} x, \ x > 0 \\ 0, \ else \end{cases}$$

Artificial neurons determine the input weights using back-propagation algorithm. In back-propagation neural network first produces its output, which is then compared with the actual output it should produce to compute a loss function. For regression problems this loss can be same as the OLS loss used in linear regression, i.e., $L = (y - \hat{y})^2$. Then the back-propagation methods calculates the gradient of the loss function with respect to the neural network's weights in a reverse order such that, gradients for the last layer neurons are calculated first and then the gradients are calculated for preceding layer and so on. We adjust the model weights after calculating the gradients by taking small steps proportional to the negative

of the gradient of loss function (L), i.e., $w = w - \alpha \nabla_w L$, where α decides the step size, also known as learning rate. In fully connected neural networks, every neuron of one layer has a direct connection to every neuron of the subsequent layer. Fully connected networks are useful as they are structure agnostic and do not need any special assumptions for the input.

BIBLIOGRAPHY

Archak, N., Ghose, A., and Ipeirotis, P. G. (2011). Deriving the pricing power of product features by mining consumer reviews. *Management science*, 57(8):1485–1509.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Chamberlain, A. and Smart, M. (2017). Give to get: A mechanism to reduce bias in online reviews. Technical report, Research Report October 2017, Glassdoor. Hentet fra https://www. glassdoor ....

Chintagunta, P. K., Gopinath, S., and Venkataraman, S. (2010). The effects of online user reviews on movie box office performance: Accounting for sequential rollout and aggregation across local markets. *Marketing Science*, 29(5):944–957.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Cui, G., Lui, H.-K., and Guo, X. (2012). The effect of online consumer reviews on new product sales. *International Journal of Electronic Commerce*, 17(1):39–58.

Freund, Y. and Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

Grey, D. (1981). Multivariate analysis, by kv mardia, jt kent and jm bibby. pp 522.£ 14· 60. 1979. isbn 0 12 471252 5 (academic press). *The Mathematical Gazette*, 65(431):75–76.

Hu, N., Pavlou, P. A., and Zhang, J. (2006). Can online reviews reveal a product's true quality?: empirical findings and analytical modeling of online word-of-mouth communication. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 324–330. ACM.

Hu, N., Pavlou, P. A., and Zhang, J. J. (2009). Why do online product reviews have a j-shaped distribution? overcoming biases in online word-of-mouth communication. *Communications of the ACM*, 52(10):144–147.

Hu, N., Pavlou, P. A., and Zhang, J. J. (2017). On self-selection biases in online product reviews. *MIS Quarterly*, 41(2):449–471.

Koh, N. S., Hu, N., and Clemons, E. K. (2010). Do online reviews reflect a product's true perceived quality? an investigation of online movie reviews across cultures. *Electronic Commerce Research and Applications*, 9(5):374–385.

Korfiatis, N., Stamolampros, P., Kourouthanassis, P., and Sagiadinos, V. (2019). Measuring service quality from unstructured data: A topic modeling application on airline passengers' online reviews. *Expert Systems with Applications*, 116:472–486.

Lahiri, S. and Ghanta, K. C. (2008). The support vector regression with the parameter tuning assisted by a differential evolution technique: Study of the critical velocity of a slurry flow in a pipeline. *Chemical Industry and Chemical Engineering Quarterly*, 14(3):191–203.

Luca, M. (2016). Reviews, reputation, and revenue: The case of yelp. com. *Com (March 15, 2016). Harvard Business School NOM Unit Working Paper*, (12-016).

Moe, W. W. and Trusov, M. (2011). The value of social dynamics in online product ratings forums. *Journal of Marketing Research*, 48(3):444–456.

Rajaraman, A. and Ullman, J. D. (2011). *Mining of massive datasets*. Cambridge University Press.

Ripley, B. D. (2007). *Pattern recognition and neural networks*. Cambridge university press.

Schölkopf, B., Smola, A. J., Bach, F., et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Shin, S., Chung, N., Xiang, Z., and Koo, C. (2019). Assessing the impact of textual content concreteness on helpfulness in online travel reviews. *Journal of Travel Research*, 58(4):579–593.

Tibshirani, R. J. and Efron, B. (1993). An introduction to the bootstrap. *Monographs on statistics and applied probability*, 57:1–436.

Verikas, A., Vaiciukynas, E., Gelzinis, A., Parker, J., and Olsson, M. C. (2016). Electromyographic patterns during golf swing: Activation sequence profiling and prediction of shot effectiveness. *Sensors*, 16(4):592.

Ye, Q., Law, R., and Gu, B. (2009). The impact of online user reviews on hotel room sales. *International Journal of Hospitality Management*, 28(1):180–182.

Zhang, Z., Ye, Q., Law, R., and Li, Y. (2010). The impact of e-word-of-mouth on the online popularity of restaurants: A comparison of consumer reviews and editor reviews. *International Journal of Hospitality Management*, 29(4):694–700.