PREDICT FAKE REVIEWERS FROM YELP DATASET.

by

Tannistha Maiti

A REPORT

ON CAPSTONE PROJECT 1

CAREER TRACK DATA SCIENCE

October, 2018

Table of Contents

List of Figures and Illustrations

List of Tables

Chapter 1

Introduction

1.1 The problem statement

Online reviews are generated in an effort to improve and enhance businesses for online retailers and service providers.These reviews are helpful, but blindly trusting them are dangerous for both the seller and buyer. Hence, it is important to identify the reviewers that are generating unfair and wrong assessment regarding businesses. In this study a binary classification scheme of fake and real reviewers is performed on raw data collected from Yelp. Attributes are generated from the raw data which are then applied to machine learning models. Based on the model fake an real reviewers are predicted.

In the literature, fake reviews are categorized into three groups: (1) Untruthful Reviews (2) Reviews on brands where the comments are only concerned with the brand or the seller of the product and fail to review the product, and (3) Non-Reviews where the reviews contain either unrelated text or advertisements. The first category, untruthful reviews, is of most concern as they undermine the integrity of the online review system. Detection of this type 1 review is a challenging task. It is impossible, to distinguish between fake and real reviews by manually, reading them.

1.2 Why is data science used to identify fake reviewers

As of 2014 there were over 18 million reviews created on Yelp. Online reviews are constantly being generated on various websites across the Internet. Hence, Big Data techniques are needed to address the problem of fake reviewers. Big Data, is often quantified with (1) Volume and scale of the data, (2) velocity or rate at which new data are created and consumed by processing engines, (3) variety of the different formats that data may be stored in, and (4) veracity of the quality level of the data.

The volume and velocity of online reviews are noted by merely visiting e-commerce and customer rating sites, such as Yelp and Amazon. Also a great variety of data is possible for industry sectors such as hotels, restaurants, e-commerce, home services etc. However the vast majority of this dataset is unlabeled, which means it is not easily known whether the review is fake or not. Thus, review spam detection is a Big Data problem, as there are numerous challenges when analyzing and classifying varying reviews from disconnected sources.

1.3 What kind of problem is it ?

The problem is a binary classification problem (1) Fake reviewer (category 1 in this study) (2) Real reviewer (category 0 in this study). The dataset is hand labeled to fake or real review and is verified from Review Skeptic (http://reviewskeptic.com/). The dataset set consists of 200 samples 80% of which is used for the training set and the rest 20% for the test set.

A model is trained on the training set and the parameters generated from the model are then implemented on the test set. These parameters predict how accurately the model performs on the test set.

1.4 Data Acquisition and Cleaning

The dataset is acquire from Yelp (https://www.yelp.com/dataset). Yelp Dataset JSON contains files composed of a single object type, one JSON-object per-line.

1. business.json : Contains business data including location data, attributes, and categories.

2. review.json: Contains full review text data including the $user_id$ that wrote the review and the $business_id$ the review is written for.

3. user.json: User data including the user's friend mapping and all the metadata associated with the user.

4. photo.json: Contains photo data including the caption and classification (one of food, drink, menu, inside or outside).

5. checkin.json: Checkins on a business.

6. tip.json: Tips written by a user on a business. Tips are shorter than reviews and tend to convey quick suggestions.

In this study checkin.json, photo.json and tip.json are not used.

More than 200 user ids are extracted fron the user dataset (user.json file). These user ids are matched with review.json file to extract review texts. Many of the users have more than one review but only the first review is extracted for every users. In very few cases the user ids do not have a corresponding review in the review file. Those user ids are not used hence this is a biased sample. More details about acquiring and concatenating the data can be found in this IPython notebook.

The diverse array of tags associated with the business reviewed by the fake and real profiles are shown in Figure 1.1 and 1.2 below. Most of the categories are related to food followed by hotel reviews, automobile reviews and so on. The categories for fake and real profiles almost a balanced distribution of categories.

Figure 1.1: Business tags for fake reviewers. A wide range of tags based on restaurants and services are noted.

1.4.1 Natural language processing

Natural language processing (NLP) involves the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. NLTK is a leading platform for building python programs to work with human language data. Some of the features of NLTK are discussed below.

1. Tokenization: In language processing, the string is broken into words and punctuation. This step is called tokenization, and it produces a familiar structure, a list of words and punctuation.

2. Stop words: Text may contain stop words like 'the', 'is', 'are'. Stop words can be filtered from the text to be processed. There is no universal list of stop words in nlp research, however the nltk module contains a list of stop words.

3. Lemmatization: Lemmatisation in linguistics, is the process of grouping together the

Figure 1.2: Business tags for real reviewers. A wide range of tags based on restaurants and services are noted.

different inflected forms of a word so they can be analysed as a single item. Lemmatisation is the algorithmic process of determining the lemma for a given word.It requires the knowledge of the grammar of a language.

4. Part of Speech (POS) tagging: POS tagging captures the structure of the sentences. The part of speech explains how a word is used in a sentence. There are eight main parts of speech - nouns, pronouns, adjectives, verbs, adverbs, prepositions, conjunctions and interjections. POS tagging is a supervised learning solution that uses features like the previous word, next word, is first letter capitalized etc. NLTK has a function to get pos tags and it works after tokenization process.

1.5 Characteristics of raw data and feature engineering on raw data

Feature engineering transforms raw data into something usable that can used for machine learning. This problem is a mix of categorical feature and cardinal (continuous) features along with text. The problem is tested with categorical feature and continuous features along with basic natural language processing analysis.The next few plots show the probability density distribution of some features.

1.5.1 Review length

The average review length may be an important indication of reviewers with questionable intentions since about 80% of spammers have no reviews longer than 135 words while more than 92% of reliable reviewers have an average review length of greater than 200 words. The distribution shows that fake reviewers write very short reviews as well as some longer reviews too. Hence, Figure 1.3 shows a bimodal distribution in comparison to real profiles. The mean of probability density distribution of real reviewers is less than fake reviewers. About 10% of the fake review lengths are greater than 1000 characters.

1.5.2 Review length without stop word

The mean of real and fake reviewers are the same around 400 characters. Also about 10% of the fake review lengths are greater than 1000 characters. Real reviewers use more stop words to express emotions. Since fake reviewers use a more direct approach so do not use much stop words.

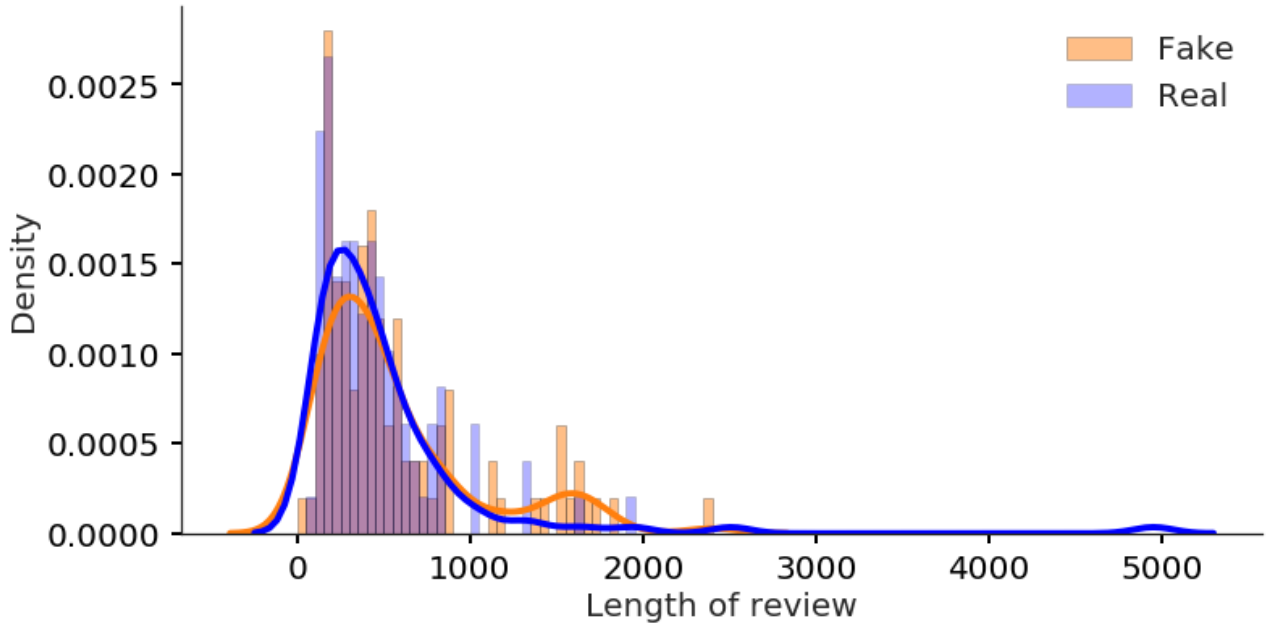

Figure 1.3: Probability distribution of normalized no.of words of review. Fake profile has a binomial distribution.

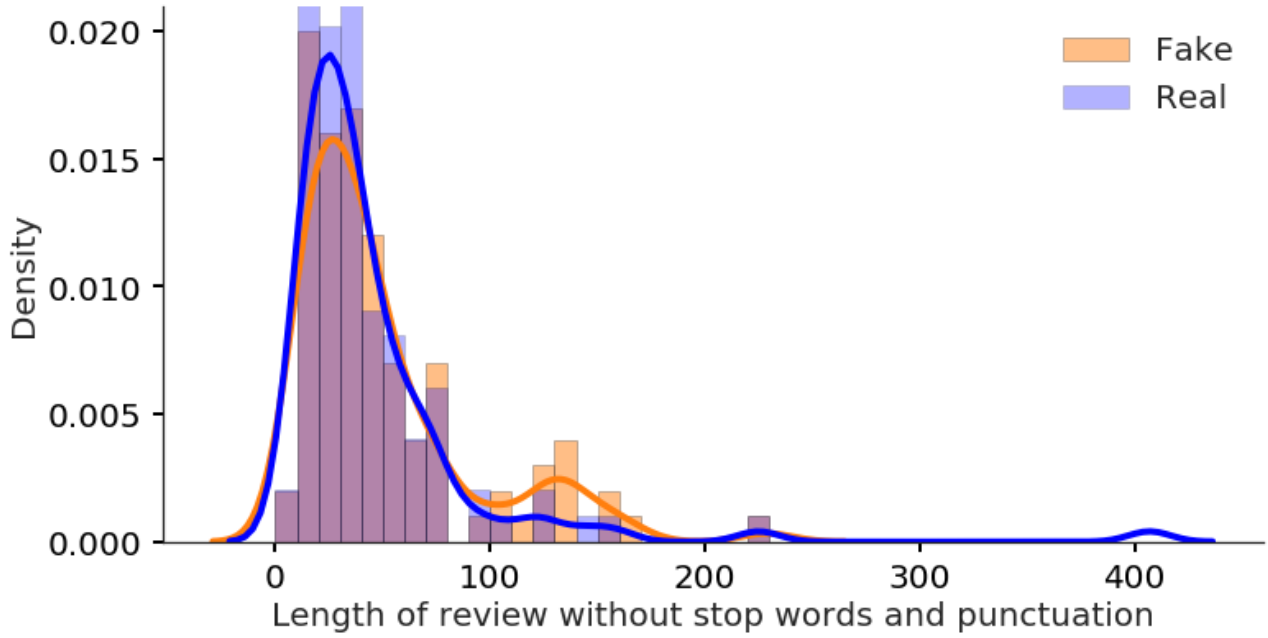

Figure 1.4: Probability distribution of normalized no.of words of review without stop words. Real profile uses more stop words.

1.5.3 Number of nouns and verbs in reviews

When reviewers want to sound sincere, but are not, they use more first-person pronouns like 'I' and 'me'. Genuine reviews focus more heavily on describing situations with nouns, while fake reviews replace these with verbs. This action is supposed to make the reviews sound more convincing, but it ends up doing the opposite. The mean number of nouns is 20 for real and 30 for fake which is not very significant difference. In Figure 1.6 there is a bimodal distribution of verbs for fake reviewers about 15% of the reviewers use 60 verbs.

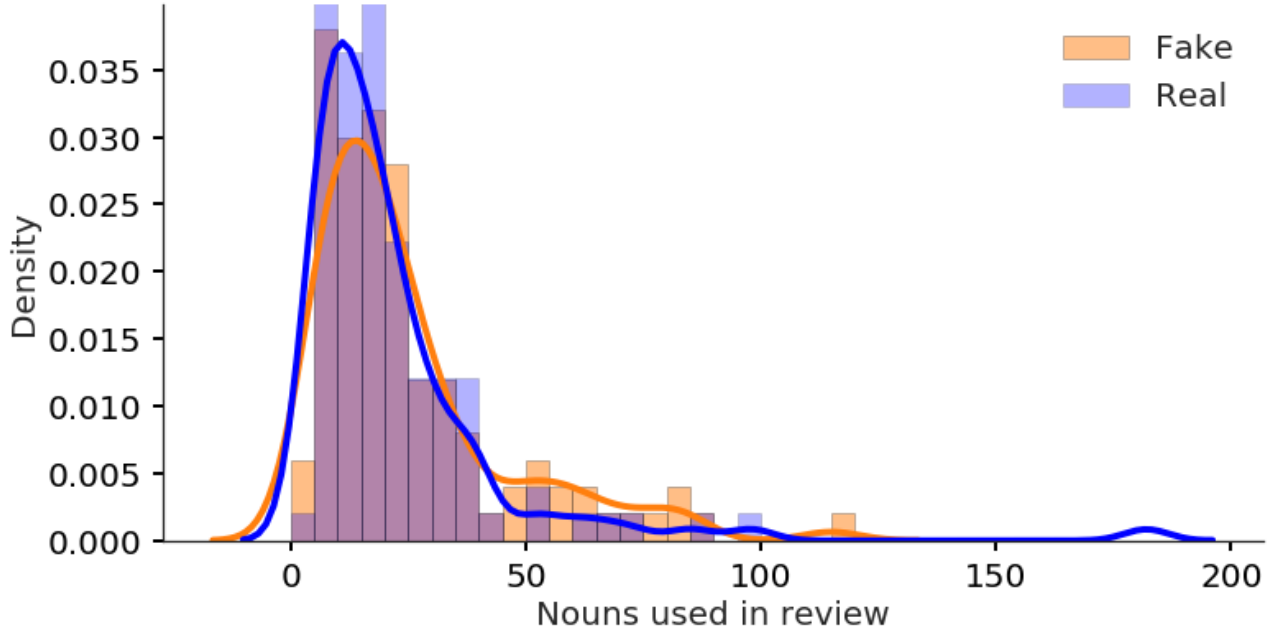

Figure 1.5: Probability distribution of normalized no.of nouns in review.

1.5.4 Number of friends and fans

This dataset is not very complete and hence shows some fake reviewers are having more fans than real reviewers. But in general fake reviewers have a low social profile.

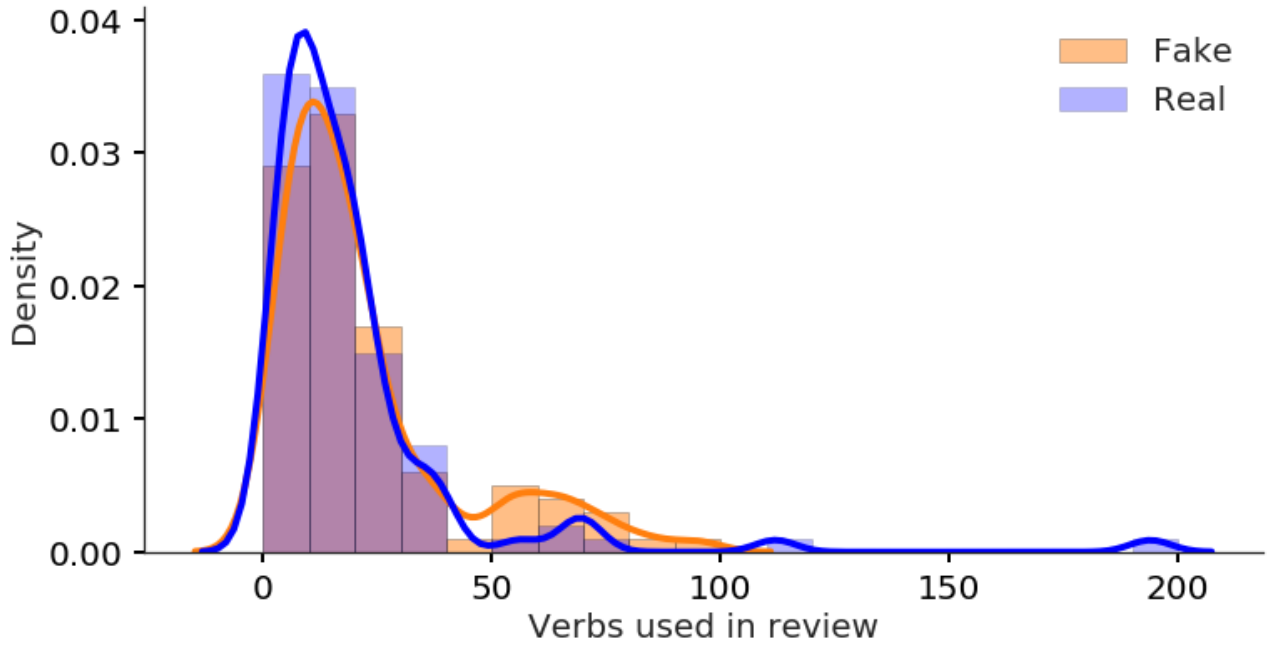

Figure 1.6: Probability distribution of normalized no.of verbs in review. Fake reviewers has binomial distribution in using verbs.

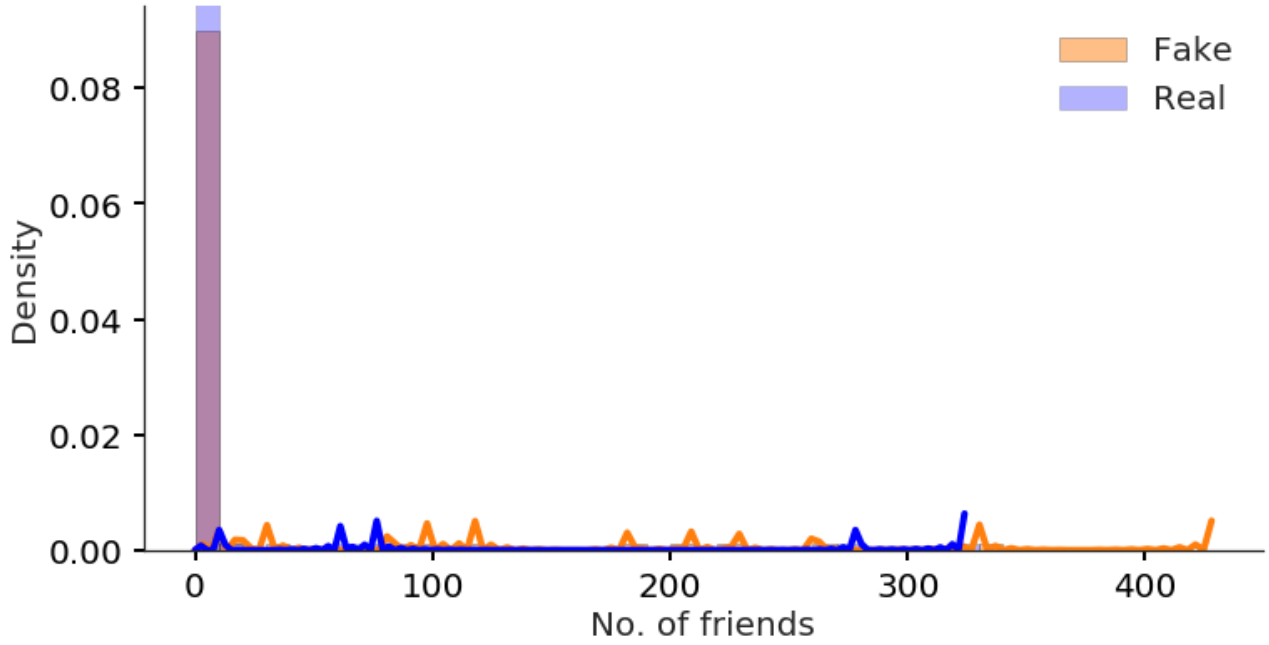

Figure 1.7: Probability distribution of normalized no.of friends. No pattern is noted.

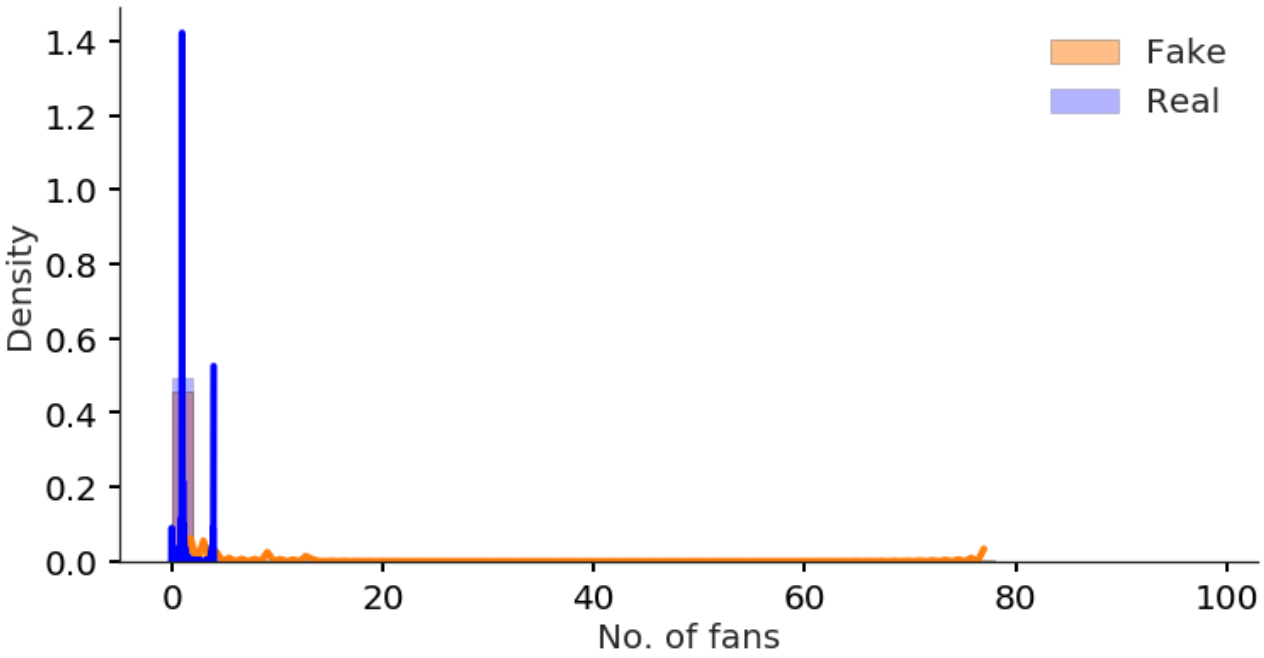

Figure 1.8: Probability distribution of normalized no.of fans. No pattern is noted.

1.5.5 Number of reviews and useful

The number of times reviews have been marked useful is shown in the Figure 1.9. Fake reviewers write mostly one review compared to real reviewers who write more than one review. Also reviews from real reviewers are marked useful more often compared to fake. The labels used in the graph are described below.

1. label=0 : Only one review.
2. label=1 : More than 1 and upto 20 reviews.
3. label=2 : More than 20 and upto 50 reviews.
4. label=3 : More than 50 and upto 100 reviews.
5. label=4 : More than 100 reviews.

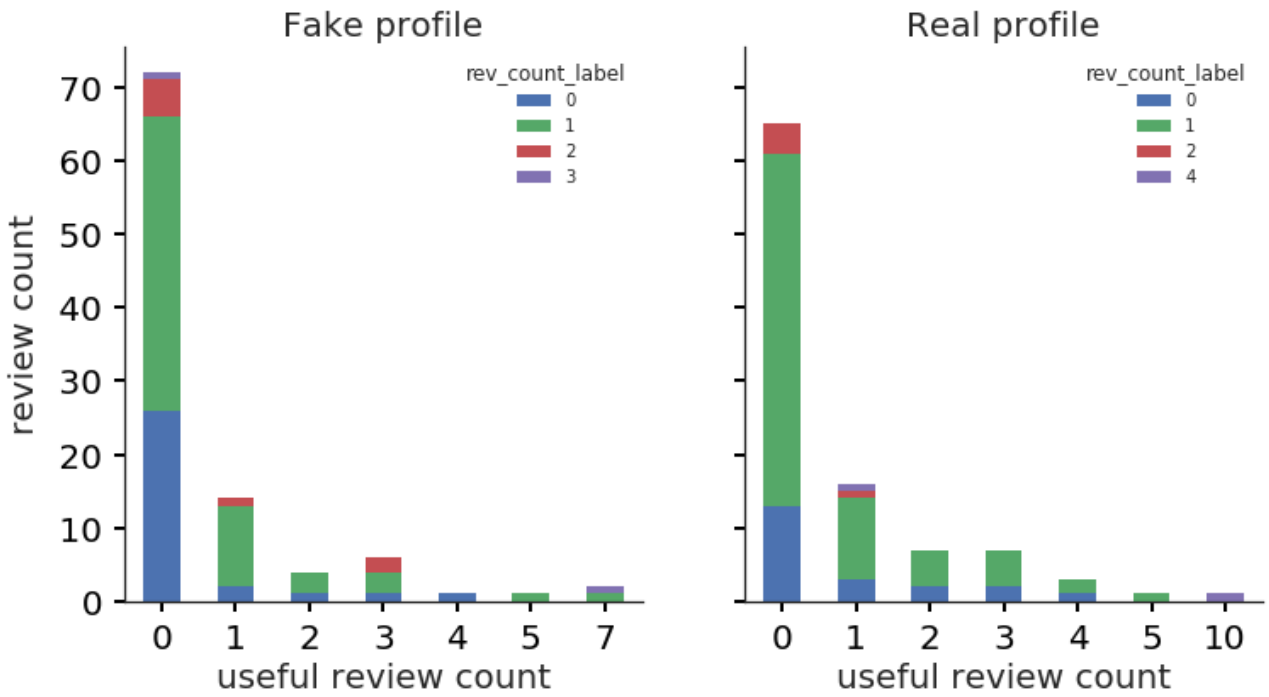

Figure 1.9: Cross tabulation showing the frequency of occurrence of number of reviews based on labels and its usefulness.

1.5.6 Distribution of actual business stars and stars rating given by reviewer

In case of a real profile 3.5, 4.0, 4.5 and 5 star businesses received ratings more positive ratings than 1 and 2 stars. However, in fake profile 3.5, 4.0 and 4.5 businesses are rated 1 stars by more than 7 fake reviewers. Also fake reviews tend to give more 1 star ratings than real reviewers.

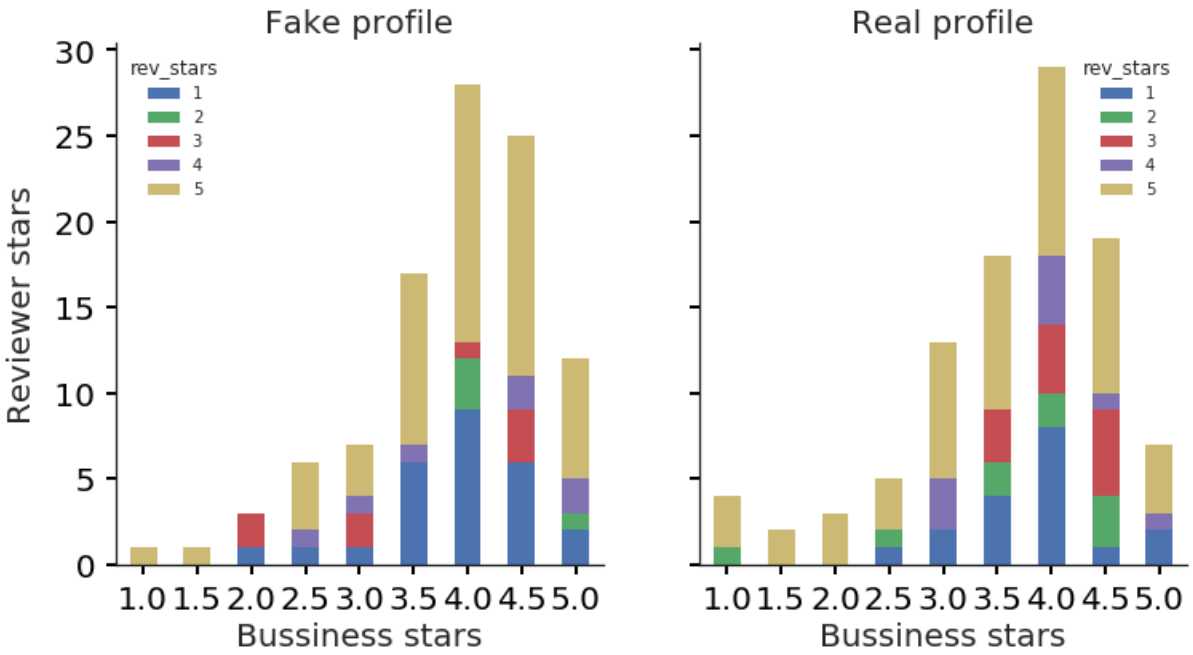

Figure 1.10: Cross tabulation showing the relation between actual stars the business has and the star rating given by the reviewers.

Chapter 2

Machine Learning

The following machine learning algorithms are used in this study.

2.1 Machine learning methods

2.1.1 Logistic Regression

The logistic regression model arises from the desire to model the posterior probabilities of the K classes via linear functions in x, while at the same time ensuring that they sum to one and remain in $[0, 1]$. The model has the form,

$$log\frac{Pr(G = K - 1|X = x)}{Pr(G = K|X = x)} = \beta_{(K-1)0} + \beta_{(K-1)}^{T}x. \tag{2.1}$$

The model is specified in terms of $K - 1$ logit transformations (reflecting the constraint that the probabilities sum to one). Although the model uses the last class as the denominator in the odds-ratios, the choice of denominator is arbitrary in that the estimates are equivariant under this choice. Since $Pr(G|X)$ completely specifies the conditional distribution, the multinomial distribution is appropriate.

2.1.2 Naive Bayes

Naive Bayes is a very simple classification algorithm that makes some strong assumptions about the independence of each input variable.There are two types of quantities that need to be calculated from the dataset for the naive Bayes model: Class Probabilities and Conditional Probabilities. The class probabilities for classes 0 and 1 are

$$P(class = 1) = count(class = 1)/(count(class = 0) + count(class = 1))$$
$$P(class = 0) = count(class = 0)/(count(class = 0) + count(class = 1)). \quad (2.2)$$

The conditional probabilities are the probability of each input value given each class value. To make predictions using Bayes Theorem.

$$P(h|d) = (P(d|h) * P(h))/P(d), \quad (2.3)$$

where: P(h—d) is the probability of hypothesis h given the data d. This is called the posterior probability. P(d—h) is the probability of data d given that the hypothesis h was true. P(h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h. P(d) is the probability of the data (regardless of the hypothesis).

2.1.3 Ensemble method

The idea of ensemble learning is to build a prediction model by combining the strengths of a collection of simpler base models. Gradient Bagging and random forests are ensemble methods for classification, where a committee of trees each cast a vote for the predicted class. Stacking is a novel approach to combining the strengths of a number of fitted models. Ensemble learning can be broken down into two tasks: developing a population of base learners from the training data, and then combining them to form the composite predictor.

Gradient Boosting

This is an ensemble technique in which the predictors are not made independently, but sequentially. In this technique the subsequent predictors learn from the mistakes of the previous predictors. Therefore, the observations have an unequal probability of appearing in subsequent models and ones with the highest error appear most (the observations are not chosen based on the bootstrap process, but based on the error). The predictors can be chosen from a range of models like decision trees, regressors, classifiers etc. Because new predictors are learning from mistakes committed by previous predictors, it takes less time/iterations to reach close to actual predictions. But we have to choose the stopping criteria carefully or it could lead to overfitting on training data.

Random Forest

The essential idea in bagging is to average many noisy but approximately unbiased models, and hence reduce the variance. Trees are ideal candidates for bagging, since they can capture complex interaction structures in the data, and if grown sufficiently deep, have relatively low bias. Since trees are notoriously noisy, they benefit greatly from the averaging. Moreover, since each tree generated in bagging is identically distributed, the expectation of an average of B such trees is the same as the expectation of any one of them. The random forest algorithm is defined as an average of B identically distributed random variables, each with variance σ^2, has variance $\frac{1}{B}\sigma^2$. If the variables are simply identically distributed, but not necessarily independent) with positive pairwise correlation ρ, the variance of the average is $\rho\sigma + \frac{1-\rho}{B}\sigma^2$. The idea in random forests is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much. This is achieved in the tree-growing process through random selection of the input variables.

Extra-Tree

Extra-Trees algorithm builds an ensemble of unpruned decision or regression trees according to the classical top-down procedure. Its two main differences with other treebased ensemble methods are that it splits nodes by choosing cut-points fully at random and that it uses the whole learning sample (rather than a bootstrap replica) to grow the trees. The predictions of the trees are aggregated to yield the final prediction, by majority vote in classification problems and arithmetic average in regression problems.

2.2 Feature Selection

The model has 43 hyperparameters to be optimized. A feature selection method is used as shown in Figure 2.1 to find the most important features. The top 20 features in terms of their feature importances are shown in the Figure 2.1. Out of these top 20 features, 13 have information about the review text, 2 are information on business and 5 have information on reviewer characteristics.

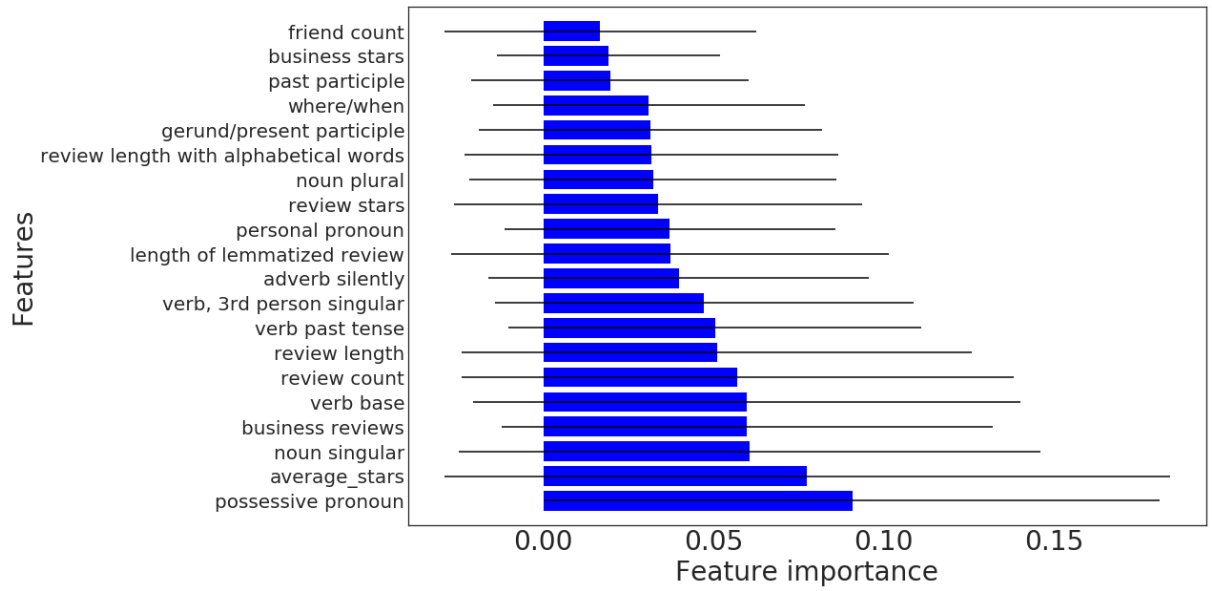

Figure 2.1: Features used in constructing machine learning model.

2.3 Modeling and evaluation metric

Once the data is pre-processed, they are fed to the classification algorithm to build the model. In order to evaluate the performance of the model, the model is tested on the test dataset. Before making predictions on test dataset, we use the exact same pre-processing steps that we used for training dataset and apply them on the test dataset. Python's scikit learn library are used for the machine learning approaches. The pipeline functions is also used to combine all the steps, i.e. pre-processing and classifier learning steps into one. The 200 reviews are divided into 50% of fake and real reviews. The test-training split is made $20\% - 80\%$. So a total of 160 rows of data are used for training set and 40 rows of data used for test set. The fit of the model is tested based on confusion matrix.

2.3.1 Confusion matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or classifier) on a set of test data for which the true values are known. There are two possible predicted classes: yes and no.The most basic terms are whole numbers (not rates) and are follows.

1. true positives (TP): These are cases in which are predicted yes and are yes.
2. true negatives (TN): These are cases in which are predicted no and are no.
3. false positives (FP): These are cases in which are predicted yes but are no.(Also known as a Type I error.
4. false negatives (FN): These are cases in which are predicted no but are yes. (Also known as a Type II error.)

This is a list of rates that are often computed from a confusion matrix for a binary classifier.

1. Accuracy: Overall, how often is the classifier correct? $\frac{(TP+TN)}{total}$.

2. Precision: When it predicts yes, how often is it correct? $\frac{TP}{TP+FP}$.

3. Recall: The ability of a model to find all the relevant cases within a dataset. $\frac{TP}{TP+FN}$.

4. F-Score: This is a weighted average of the true positive rate (recall) and precision. $2 * \frac{Precision*Recall}{Precision+Recall}$.

Chapter 3

Results

3.1 Parameter tuning in machine learning models

This section discusses choosing the best parameter for the machine learning models based on accuracy score.

3.1.1 Inverse of regularization: logistic regression

An inverse of regularization is used to improve the generalization performance, i.e., the performance on new, unseen data. In more specific terms, regularization is increasing bias if our model suffers from (high) varying (i.e., it overfits the training data). On the other hand, too much bias will result in underfitting. The Figure 3.1 shows the accuracy score for a set of inverse of regularization values. 1, 10 and 100 have almost same accuracy score so a value of 100 is chosen to avoid underfitting.

3.1.2 Number of steps: gradient boosting

The number of boosting stages to perform. Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance. Figure 3.2 shows that 30 or 40 steps gives the maximum accuracy.

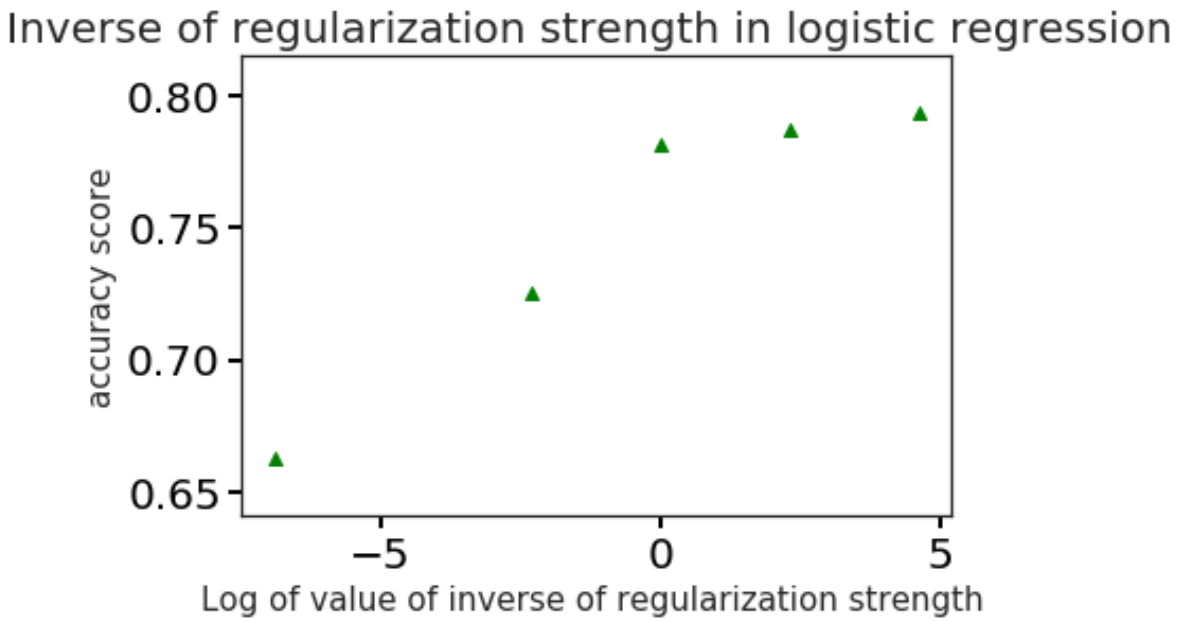

Figure 3.1: Selecting of inverse of regularization in logistic regression. 100 has the best accuracy.

3.1.3 Number of trees: random forest

The number of trees in the forest. Figure 3.3 shows that 70 or 100 trees gives maximum accuracy.

3.1.4 Number of trees: extra tree

The number of trees in the forest. Figure 3.4 shows that 70 gives the best estimate of maximum accuracy.

3.2 Model Comparisons

Logistic Regression, Gaussian Naive Bayes, Random Forest, Gradient Boosting and Extra Randomized Trees classifiers to build a model to predict Fake reviewers. Results are shown for both the training set and testing set. The results of various evaluation metrics scores are shown in Table 3.1 for all models.

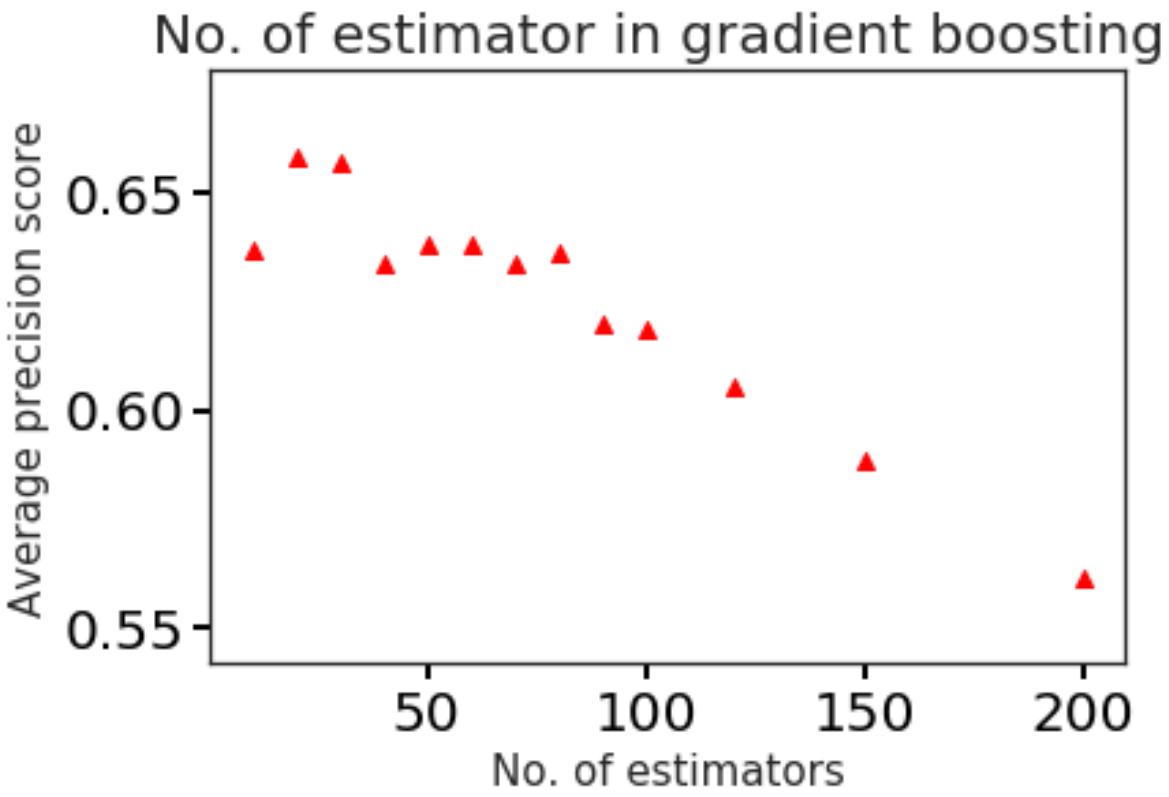

Figure 3.2: Selecting the number of steps in gradient boosting. 30/40 has the best accuracy.

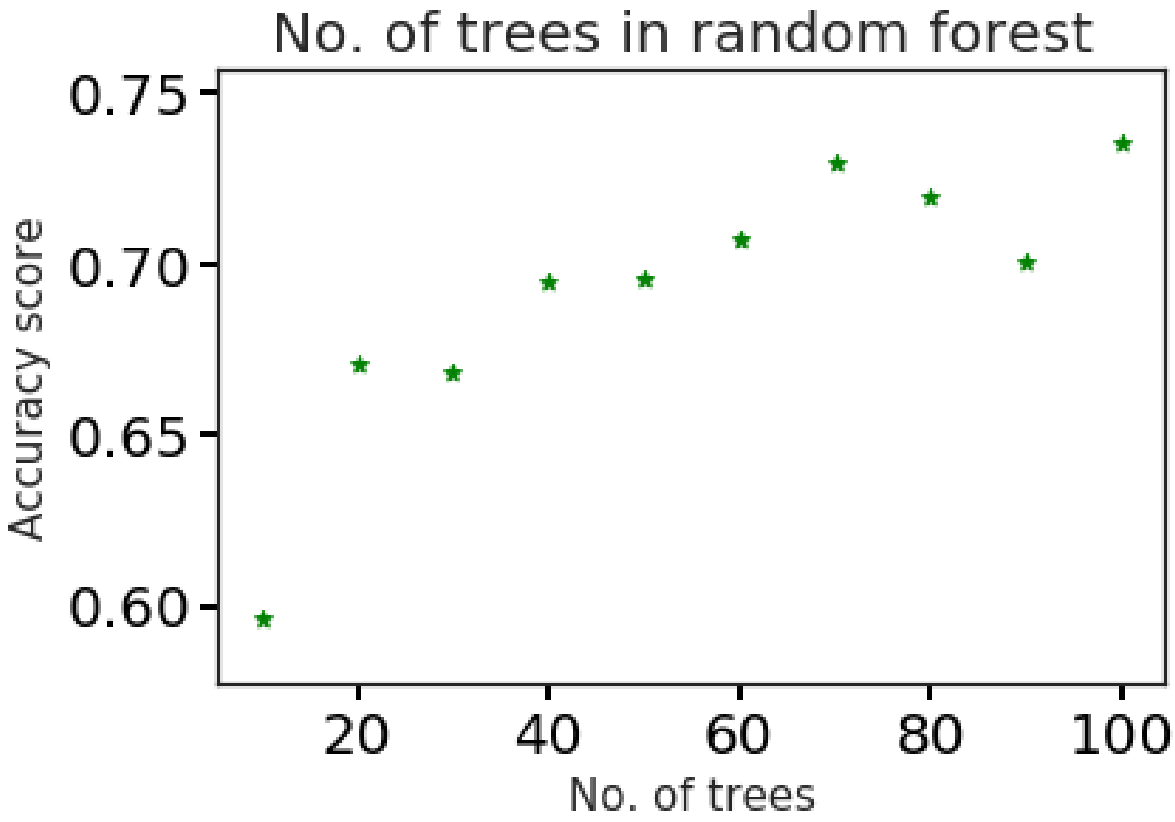

Figure 3.3: Selecting the number of trees in random forest. 70 trees give the best accuracy.

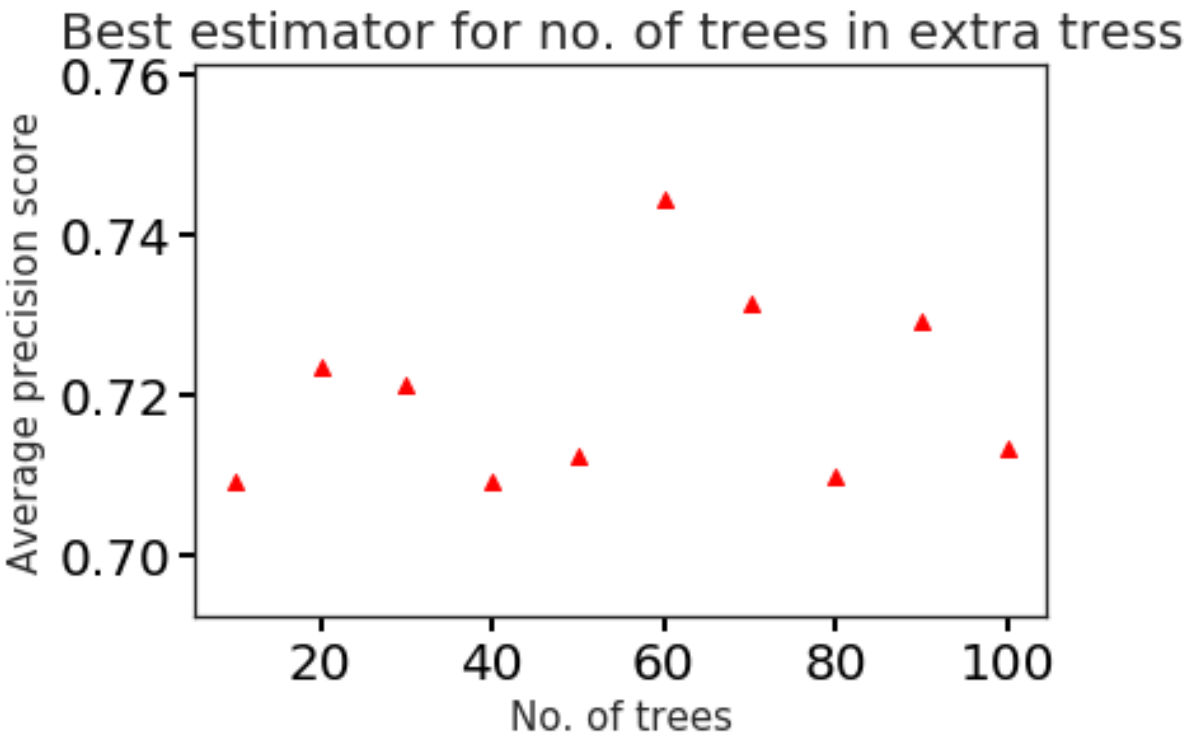

Figure 3.4: Selecting the number of trees in extra tree. 70 trees give the best accuracy.

Algorithm	Accuracy of training set	Accuracy of test set	Precision of training set	Precision of test set
Logistic Regression	0.68	0.70	0.68	0.79
Naive Bayes	0.56	0.5	0.59	0.5
Gradient Boosting	0.69	0.63	0.70	0.67
Random Forest	0.94	0.73	0.95	0.73
Extra Tree	0.97	0.68	0.97	0.67

Table 3.1: Comparing all models for train-test set. Logistic regression (green) is the best model that predicts with an accuracy of 70% and precision of 79%.

3.3 Model and Recommendations

This section discusses the best model that can be used to predict fake reviewers. In statistics, a fit refers to how well a target function is approximated. In supervised machine learning the unknown underlying mapping function is approximated from the output variables of the training set. The target function is not always known so calculating the residual errors is not always the case in machine learning. In supervised learning approximations are made from samples of noisy training data.

Two scenarios appear (1) Overfitting and (2) Underfitting. Overfitting refers to a model that models the training data too well. Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. Decision trees are a nonparametric machine learning algorithm that is very flexible and is subject to overfitting training data. This problem can be addressed by pruning a tree after it has learned in order to remove some of the detail it has picked up.

Underfitting refers to a model that can neither model the training data nor generalize to new data. An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data. A good fit in Machine Learning happens when the model is at the sweet spot between underfitting and overfitting which is very difficult to do in practice.

Table 1 shows accuracy and precision for the five different models used in the study. The extra tree and random forest perform best on the training set 97% and 94% respectively. Logistic regression has an accuracy of 68%, whereas gradient boosting and Naive Bayes doesn't perform well. The accuracy and prediction on test set has a different story. In test set data the best accuracy is predicted by random forest which is 73% followed by logistic regression of 70%. However, it is to be noted that random forest suffers from overfitting i.e a vast difference in accuracy between test and training set. The decision trees have trained on noise so the accuracy has decreased from 94% on training set to 73% on test set. Overfitting is also noted in extra tree where the accuracy has decreased from 97% on training set to 68% on test set. The logistic regression model on the other hand has almost similar accuracies of 68% and 70% on the train and test set respectively. The logistic regression also have high precision of 79% i.e. it can predict the positives in the model quite well. In this particular problem logistic regression is a good fit model performing with an accuracy of 70% on test set.

3.4 Conclusion

In this project, The review text of users were merged with business reviewed by the user. 200 review samples were labeled as fake and real. A quick summary of the exploratory data analysis reveals that 1. Fake reviewers use more verbs and nouns than real user. 2. Fake reviewers use less stop words.i.e express less emotions. 3. Fake reviewers give more 1 star reviews than real users. 4. Reviews of real users are more useful. 5. There is a disparity in

star rating on business and star rating by fake users.

After exploring all datasets, we used five different supervised classification algorithms (Logistic Regression, Gaussian Naive Bayes, Random Forest, Gradient Boosting and Extremely Randomized Trees) are used to train the predictive model by using 80% of the whole data. The remaining 20% was used to evaluate the model i.e. test data. Overall logistic regression performed well on both test and train data set with 70% accuracy. Random forest and extra tree performed well on training dataset but suffered from overfitting on test data set.

However, it is to be noted that there are some limitations in the current model such as lack of complete data because only 200 samples were labeled for fake and real. It is also difficult to identify fake and real charecteretstics. The model can be further improved in future with more sampled dataset and some reviews that are already marked as fake and real.