A Semantic and Classification Approach for Yelp's Star-ratings

Nicholas Smeele, 500772

1. Introduction

Social networks have changed the customer-business relationship as the performance of businesses has become more dependent on how customers are interacting and feeling about the business on online social platforms. These platforms enable individuals to share opinions and experiences with a greater social reach which can result in social influence effects, i.e. information cascades (Khan, M.R., 2017). Yelp is one such platform and information source that helps customers locate local businesses based on social networking functionalities such as reviews and star-ratings. Many individuals use Yelp to plan a night out and studies have found that Yelp restaurant reviews affects customers' food choice decision-making; a one-star increase on average led to a 5-9 percent increase in revenue of independent restaurants (Luca, M., 2011). The reviews are, therefore, not only valuable to customers but are providing valuable insights for restaurant owners as well. Local restaurants can identify what elements their customers liked or what needs to be improved in their service and food offering. However, star-ratings provide an overall indication of whether customers were satisfied with the restaurant and do not indicate what features stimulated them to give the specific rating. This is a challenge since reviews are often technologically poor; business owners often have no choice but to browse through massive amounts of text to find interesting information.

This paper analyses text reviews for the restaurant category *diners* on Yelp to uncover latent topics, i.e. identifying hidden diner features that best describes a set of reviews, and classify the reviews in star-rating sentiment classes based on the identified features. However, a similar analysis can be easily done for any other restaurant category provided sufficient data is present. Research on uncovering and identifying the effect of latent topics in reviews on star-rating sentiment help determine important diner restaurant features. Hence, local diner owners obtain insights into what stimulates customers to give a specific rating. Therefore, this study is dedicated to the following research question: *"What diner factors are most impactful on the predicted sentiment in star-rating on Yelp?"* To answer the research question, this paper addresses the collected dataset and the required preprocessing steps followed by the methods for extracting information from the text reviews and discussion with the results. Finally, the conclusion and discussion for further research is addressed.

2. Data

The dataset for this study is provided by Yelp under its Dataset challenge initiative [1] that consists of a subset of businesses, reviews and users separated datasets. As the basic aim of this study is to identify latent topics in reviews and relate these features to the sentiment in star-ratings for diners, the business and review dataset has been selected to be used which consists of 42,153 local businesses and 1,125,458 reviews. In addition to the reviews, the data contains business attributes such as opening and closing hours, parking availability, ambience characteristics, business categories, location, and many more. Though the dataset is quite extensive with 105 variables in the business and 10 variables in the review dataset. Therefore, the two datasets are merged together based on business IDs and all irrelevant variables are removed. Hence, the merged dataset contains 1,113,601 unique reviews for the 42,153 local businesses where the remaining 11,857 reviews were unrelated to any local business and are removed.

The variables that are considered in the merged dataset are: (i) business IDs and (ii) user reviews in text, (iii) star-ratings in numeric values, (iv) date of a review post, (v) business category that categorised local businesses in main and sub-categories, (vi) total number of reviews for each business, and (vii) whether a business is open or closed as a categorical variable. However, the variables *star-ratings* and *reviews* are the main variables for this study whereas the other variables are used for further subsetting. For the purpose of this study, all businesses categorised as diners and were labelled as 'open' are considered in the merged

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

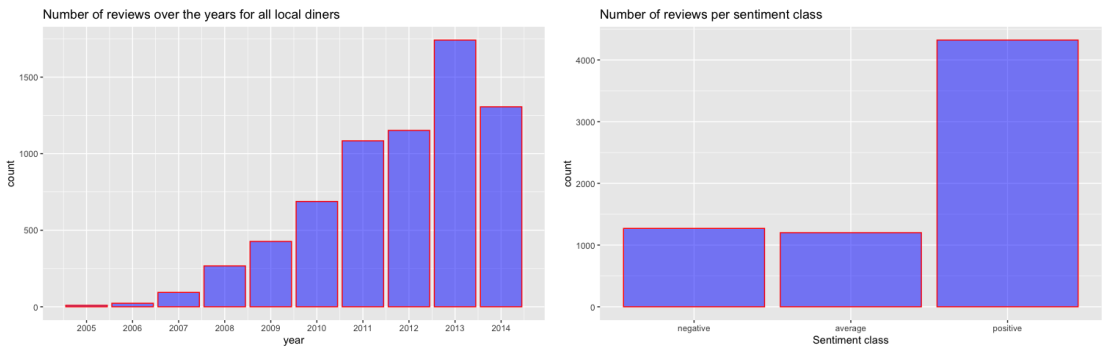

Figure 1: Distribution of reviews over years (left) and sentiment classes (right)

dataset, i.e. businesses that do not fit these conditions were removed. Moreover, diners that received less than or equal to 20 reviews in total were removed since the first set of reviews are most likely to overestimate the eventual rating which is also called the warm-start bias (Potamias, M., 2012). As the purpose of this study is to classify reviews in sentiment rating classes based on latent topics, the sentiment is derived from the *star-rating* variable which is factored and releveled from a one- to five-star range to three levels: (i) one- to two-stars leveled as a *negative* rating, (ii) three-stars as *average* rating and (iii) four- to five-stars as a *positive* rating. Figure 1 shows the distribution of the reviews over the sentiment categories which indicates that the dataset is imbalanced, i.e. the *positive* class is dominant over the other classes. Hence, after a training set containing two-thirds of the entire dataset is created, the resampling method *downsampling* is used on the training set to avoid unreliable and biased classification results. By downsampling the training set, the dominated *positive* class is reduced in observations to ensure that all classes are equally weighted during training. The test set, however, is untouched, contains the remaining one-third of the dataset and is used to evaluate the classification results. Furthermore, Figure 1 provides the diners' review distribution between 2005 and 2014 which are all considered in this study. On this basis, the dataset that is used contains 6,793 unique reviews with each related to a local diner and two variables: (i) user reviews in text and (ii) sentiment categorised into negative, average and positive rating classes.

Since the reviews are provided in the natural human format of sentences that cannot be used directly in the models for the analysis, an n-gram linguistic vocabulary is created. The words in each review are separated into single tokens so-called *unigram* words after which all upper-case characters are set to lower-case characters and all non-ASCII characters, punctuations, stopwords, unnecessary whitespaces and numbers are removed so that a *bag-of-words* was generated. Moreover, each word token is broken down to their root word by using Stemming. Words with a common root often share a similar meaning for which these tokens are grouped into one token to reduce the size of the created vocabulary (Anandarajan, M. et al., 2019). Furthermore, word tokens that do not appear in more than one percent of the reviews, i.e. documents, are removed. By removing less meaningful words, the complexity and noise in the text is reduced which stimulates the analysis to focus on meaningful words instead. Finally, the words tokens are transformed to a vector space in a Document-Term Matrix (DTM) on the basis of the words' frequency, i.e. Term frequency, which is used as input data for the first stage of this study.

3. Methods

The aim in this study is to uncover latent topics in sets of text reviews and classify each review document in sentiment rating classes based on the identified latent features. The first stage in this research is to identify and extract latent topics by using the Latent Dirichlet Allocation (LDA) model. This algorithm uses the entire Document-Term Matrix (DTM) to create a probability distribution on a latent, low-dimensional topic space for each review document which are combined into a new dataset with each row corresponding to a specific review and designated sentiment rating class (Blei, D.M. et al., 2003).

Based on the newly obtained dataset, the second stage implements two machine learning algorithms: (i) Naive Bayes classification and (ii) Random Forest classification. The Naive Bayes (NB) algorithm is the baseline model in this study since it is simple, computationally inexpensive and it has been shown to be

effective in previous text categorization studies (Pang, B. et al., 2002). The Random Forest (RF) algorithm is compared against the baseline model which is a *black-box* model that is able to learn more complex relationships among variables. Moreover, it is less complex to tune and hardly overfits the training data compared to Boosting trees which must be tuned extensively to avoid overfitting and increase the prediction performance (Hastie, T. et al., 2017). Both classifiers are trained on a downsampled training set of the newly created dataset, as suggested in section 2, and evaluated on a testing set based on three evaluation measures (see section 3.4). The best performing classifier is interpreted in the third and last stage by using Permutation Feature Importance (PFI) that measures the influence of each extracted latent feature on the reviews' predicted classes (Fisher, A. et al., 2019).

3.1 Latent Dirichlet Allocation (LDA) Model

The LDA model is an unsupervised and generative method for topic modeling that uses the DTM to estimate the Dirichlet priors between the document-topic (DT) and word-topic (WT) distributions (Blei, D.M. et al., 2003). This method allows documents and words to be explained by unobserved variables that might be clustered into similar groups. Thus, documents may be viewed as a distribution of various topics whereas each topic is a mixture of various words that are assigned via LDA. More formally, the probability that a document is assigned to a topic $z_j \in \{1, ..., k\}$ and the related words $\mathbf{w} = \{w_1, ..., w_n\}$ can be given as:

$$p(\mathbf{w}) = \int_{\theta} (\prod_{n=1}^{N} \sum_{z_j=1}^{k} p(w_n|z_j; \beta) p(z_j|\theta)) p(\theta; \alpha) d\theta$$

Where $p(\theta; \alpha)$ is Dirichlet prior with θ as topic distribution for the documents and α as the parameter to control the document-topic density, $p(z_j|\theta)$ is the probability that the document is assigned to topic (z_j), and $p(w_n|z_j; \beta)$ is the probability that word (w_n) is assigned to (z_j) with β to control the word-topic density. To estimate the Dirichlet prior, the Markov Chain Monte Carlo (MCMC) algorithm Gibbs sampling is used. Studies suggested that between 1,000-5,000 iterations are needed for reasonable accuracy (Raftery, A.E. & Lewis, S., 1991). In this paper, 1,000 Gibbs sampling iterations are used for computational time reasons. Moreover, a burn-in period with 100 iterations is initialized to give the Gibbs sampler time to converge and avoid any misleading estimated joint distribution (Gelfand, A.E., 2000).

There are a set of hyperparameters that affect the Dirichlet prior. The two mixing parameters α and β affect the DT and WT densities with $\alpha = 0$ and $\beta = 0$ implying a concentrated DT and WT distribution whereas $\alpha \to \infty$ and $\beta \to \infty$ creates a wider distribution. The LDA model also assumes that the number of K topics are known beforehand. In practice, α and β are chosen in an ad-hoc manner even though it impacts the outcome (George, C.P. & Doss, H., 2018). However, studies have shown that $\alpha = 50/T$ and $\beta = 200/W$ are reasonable values with T is equal to K topics and W is the number of total $\mathbf{w}$ (Nguyen, D.Q. et al., 2013). To find the optimal parameter values, the number of K topics is determined first by using $\alpha = 0.1$ and $\beta = 0.05$, which are the default values, and iteratively create LDA models with one to twenty topics. The model with the highest coherence measure, i.e. the degree of semantic similarity or similar meaning between words in topics, is selected. To verify the robustness of α and β, the process is repeated with the default α and β, $\alpha = 50/T$ and $\beta = 200/W$, and two arbitrary values while keeping K constant where the parameter combination is selected that maximizes the estimated coherence measure. Hence, the LDA model with the selected K topics and best hyperparameter values is used to extract the probability distribution of the latent features for each review document from the entire DTM.

3.2 Naive Bayes (NB) Classification

The probability distributions of the latent features for all documents are used to classify each review in sentiment rating classes. One approach for this classification study is to use the Naive Bayes (NB) algorithm which is the baseline model. This method relies on Bayesian probability theory where the estimated likelihood of a document's class is based on conditional probabilities and priors (James, G. et al., 2013). In essence, the relation between a given topic feature z_j and class c_n is observed using Bayes' theorem:

$$p(c_n|z_j) = \frac{p(z_j|c_n)p(c_n)}{p(z_j)}$$

Where $p(c_n|z_j)$ is the posterior probability that measures the likelihood of c_n given z_j, $p(z_j|c_n)$ is the conditional probability that assumes the effect of the value of z_j on a given c_n is independent of the value of other K topics, $p(c_n)$ and $p(z_j)$ are the prior probabilities which provides the likelihood of occurrence based on prior information. The theorem assumes class-conditional independence that not always holds. However, studies have shown that the method still performs fairly well (Huang, J. et al., 2003). Thus, the NB classifier computes the probability of document's class c_n, given the priors provided by $z_j \in \{1, ..., k\}$, as the product of probabilities of each z_j conditioned on c_n, the prior of c_n and scaling factor $1/N$ where the document is assigned to the class with the maximum likelihood estimation.

$$p(c_n|Z) = \frac{1}{N} p(c_n) \prod_{j=1}^{n} p(z_j|c_n)$$

The prior probabilities may be estimated from the training set. Where a class's prior is straightforward to estimate, the features' priors must assume a probability distribution (Lowd, D. & Domingos, P., 2005). For discrete features, Multinomial (MD) and Bernoulli (BD) distributions may be assumed whereas MD relies on frequency and BD on binary data. The topic features, however, are continuous in their probabilistic values for which the Gaussian (GD) distribution is used as it relies on continuous data (Lowd, D. & Domingos, P., 2005). The strict assumption of GD is that the data must be normal distributed. Therefore, the non-parametric approach GD kernel density estimation is used to make the model more robust in domains that violate the assumption (John, G.H. & Langley, P., 1995). This approach estimates the GD of the underlying data by weighting observations differently depending on their distance from evaluation points across the smoothed distribution (Hastie, T. et al., 2017).

3.3 Random Forest (RF) Classification

The second classification approach is to create a Random Forest (RF) classifier that is built upon decision-trees and aggregates as well as decorrelates many trees to improve the predictive power since single trees are non-robust and unstable with low predictive power (James, G. et al., 2013). The decision-trees in the RF model are created by using the recursive binary splitting approach. This is a top-down greedy approach that splits the predictor space into subregions by selecting the predictor variable with the purest node partition, i.e. purest subregion partition. This splitting procedure is repeated until the purity of the subregions are maximized (James, G. et al., 2013). n this sense, purity measures how dominant a class of observations is within the created subregions. To select the predictor variable with the purest node partition per splitting iteration, two splitting criteria or so-called purity measures are considered.

Gini Index (GI) This measures the total variance across the K classes and subtracts the sum of squared proportions of training observations in each child node per prediction class from one. If the proportions in each child node are close to zero or one, the measure provides a small value indicating that the partition results in pure nodes, i.e. observations of the same class (Hastie, T. et al., 2017).

Entropy This measure provides similar results as GI where it takes small values for pure partition results. The computation, however, is different since it multiplies the sum of proportions of training observations and the logarithmic proportions in each child node per prediction with negative one (Hastie, T. et al., 2017).

Both measures provide similar results where GI is less computationally intensive since it does not consider logarithmic functions. Therefore, the GI is used as a splitting criterion. Moreover, decision-trees consider a stopping criterion to avoid fully grown and overly complex models that overfit the training data, i.e. high variance predictions. In general, by adding bias to reduce the variance, the complexity can be reduced which is known as post-pruning (Hastie, T. et al., 2017). However, the RF model becomes more effective with high variance decision-trees for which each tree is fully grown and not pruned in this study.

The RF algorithm utilizes the ensemble method *bagging* to reduce the variance in the model by repeatedly taking equally sized bootstrap samples, with replacement, from the original dataset; where a decision-tree is created on each sample that contains approximately two-thirds of the original dataset (Hastie, T. et al., 2017). The remaining one-third is used for testing, i.e. out-of-bag (OOB) data; meaning all decision-trees that have not seen the predicted observation in training are used to predict the observation's class by taking

the majority vote of the aggregated predictions. To determine the best number of bootstrap samples (B), the OOB error rate is used which stabilizes at some point as B increases (Hastie, T. et al., 2017). In other words, the OOB error rate does not improve significantly by creating more samples and trees after the stabilization point is reached. Hence, B is set at the stabilization point for computational time reasons.

To reduce the overall variance further, RF decorrelates each independent tree by randomly selecting a subset of m predictors from all p predictors as split candidates per tree-building iteration. This avoids that each decision-tree becomes similar to each other as it selects the predictor variable with purest node partition in the same iteration order. In practice, the rule of thumb for the number of randomly selected predictors per iteration is $m \approx \sqrt{p}$ (James, G. et al., 2013). However, in this study, this parameter is tuned by fitting and comparing RF models with a range of m-values after B is selected based on the OOB error. The parameter value that minimizes the OOB error is selected to be used in the best RF classifier.

Since the OOB error rate is based on a subset of trees, the RF classifier's overall performance is evaluated on the designated test set, which contains one-third of the complete dataset, based on three measures (see section 3.4). This would be a more accurate evaluation as all trees are used to compute the testing error rate. Thus, the RF classifier is fitted and bootstrap sampling is applied on the training set.

3.4 Performance Evaluation

The two classification approaches are evaluated and compared on the basis of the prediction performance on data that was not seen by both algorithms during training, i.e. testing set. The classifier with the best model-fit is selected for the remainder of this study. Since the machine learning algorithms are used as classification models, class-specific measures are used which can help determine the class- and overall performance of each classifier. The class-specific measures that are used in this study are:

Precision This measures how many observations are correctly predicted (x_i) compared to the total number of predicted observations (P), and does not consider the actual number of observations in the class. It is also known as the *prediction accuracy rate* (Anandarajan, M. et al., 2019).

$$Precision_i = Accuracy_i = \frac{x_i}{P}$$

Recall This measures how many observations are correctly predicted (x_i) compared to the total number of observations that actually belong to that class (C) (Anandarajan, M. et al., 2019).

$$Recall_i = \frac{x_i}{C}$$

F-measure This computes the balance between precision and recall where the maximum value is 1 and provides a goodness-of-fit assessment for the classifiers (Anandarajan, M. et al., 2019).

$$F_i = 2 * \frac{Precision_i * Recall_i}{Precision_i + Recall_i}$$

3.5 Permutation Feature Importance (PFI)

The selected classifier with the best model-fit is interpreted by the model-agnostic method permutation feature importance (PFI). This identifies the most impactful restaurant factors on the predicted sentiment in star-rating for each review document. In essence, PFI measures the changes in prediction error after the features' values are permuted; i.e. breaking the relationship between each predictor and actual outcome (Fisher, A. et al., 2019). This paper repeats the permutation procedure of each feature twenty times to create a 95% confidence interval and obtain more stable results. Thereafter, the classification error on the predicted classes of the permuted feature is computed and compared to the original error of the non-permuted data.

$$FI_j = \frac{error_{perm}}{error_{orig}}$$

When the error ratio is higher than one, the feature is important for the classifier's predictions since the permuted error is higher than the original error. On the contrary, the feature is unimportant when the model error is unchanged after permutation; i.e. the error ratio is lower or equal than one (Breiman, L., 2001). By comparing each feature on its importance, the most impactful restaurant factors can be identified. To compute PFI, the test data is used which provides the actual prediction effects. The training data would give misleading results since the classifier has used the training set during the learning procedure. Thus, the PFI would be not realistic compared to the PFI results based on the test set (Fisher, A. et al., 2019).

4. Results

Before the classification performances can be evaluated, the latent features must be extracted from the text reviews and some model-building decisions must be made. The first step is to obtain the probability distribution of the latent features for each review via the best LDA model for which the optimal number of K topics, α and β must be determined. To find the best K topics, twenty models are created based on the Document-Term Matrix (DTM) of all review documents. Each model differs in the number of K topics; starting from one to twenty topics while the default $\alpha = 0.1$ and $\beta = 0.05$ are used. From this procedure, each LDA model obtained a coherence measure. Figure 2 shows that the model containing 17 topics has the highest measure with 0.097; meaning that this model has the largest degree of semantic similarity between the words in the latent topics. Based on this result, the LDA model with $K = 17$ is selected and used to tune the other two parameters to derive the best model.

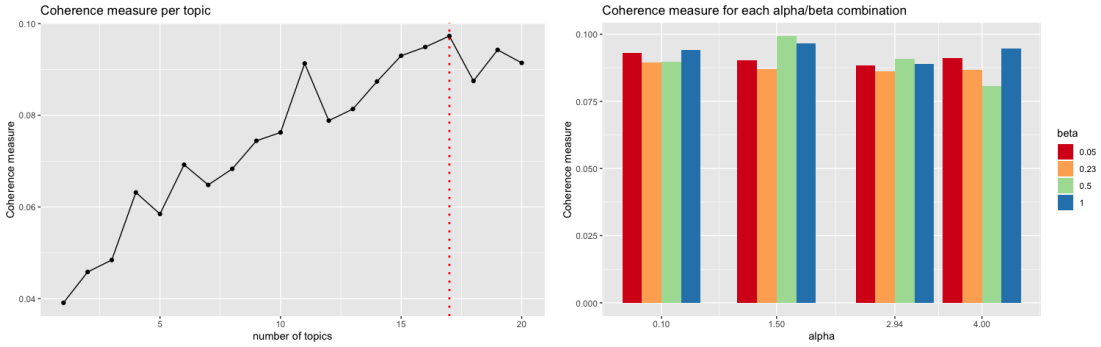

Figure 2: Coherence measures over each K topics (left) and hyperparameter combinations (right)

To verify the robustness of α and β, a search grid was initialized containing the default α and β, $\alpha = 50/T$ and $\beta = 200/W$, and two arbitrary values for both parameters: $\alpha = \{0.1, 1.5, 2.94, 4\}$ and $\beta = \{0.05, 0.23, 0.5, 1\}$. This resulted in sixteen value combinations where a LDA model is created based on each value pair while keeping $K = 17$ constant. Figure 2 shows the coherence measure for each α and β combination. On this basis, the best model is selected with $K = 17$, $\alpha = 1.5$ and $\beta = 0.5$ since it obtains the maximum coherence measure.

The best LDA model, however, is fitted based on the DTM to obtain the probability distribution of the latent topics for each review which are combined into a new dataset with each row corresponding to a specific review and its sentiment rating class. Table 1 provides an overview of the extracted topics with each containing the top-10 words with the highest occurrence probability. To use this dataset for the remaining study, the topics must be defined in distinctive features. Hence, some topics are merged since their semantics are overlapping. Thus, the 17 extracted topics resulted in 14 distinctive features that are created and named as: (i) 'Breakfast menu' defined topic 1 and 4, (ii) 'Service' created by topic 2 and 6, (iii) 'Diner menu' by merging topic 3 and 5, (iv) 'Opening hours' by topic 7, (v) 'Overall quality' by topic 8, (vi) 'Food quality' by topic 9, (vii) 'Lunch menu' by topic 10, (viii) 'Price quality' by topic 11, (ix) 'Food offer' by topic 12, (x) 'Dessert menu' by topic 13, (xi) 'Visit frequency' by topic 14, (xii) 'Location' by topic 15, (xiii) 'Ambience' by topic 16, and (xiv) 'Guest experience' by topic 17.

Table 1: Top-10 words over each (raw) extracted latent topic

t_1	**t_2**	**t_3**	**t_4**	**t_5**	**t_6**	**t_7**	**t_8**	**t_9**
breakfast	just	chees	breakfast	steak	us	vega	get	like
biscuit	review	hot	egg	special	order	night	im	just
gravi	ask	onion	pancak	potato	wait	eat	can	realli
portion	said	dog	toast	order	tabl	cafe	ive	bad
good	want	sauc	bacon	meal	came	food	dont	wasnt
bear	know	fri	coffe	hard	minut	hour	go	pretti
place	dont	grill	french	get	waitress	late	come	better
huge	one	side	waffl	cook	took	stay	alway	tast
hash	say	like	hash	salad	seat	denni	like	noth
egg	order	top	blueberri	deal	got	open	time	much
t_10	**t_11**	**t_12**	**t_13**	**t_14**	**t_15**	**t_16**	**t_17**	
burger	food	chicken	menu	time	place	diner	great	
fri	good	order	pie	back	go	food	place	
good	servic	salad	one	will	like	place	love	
sandwich	price	restaur	cream	tri	lot	feel	friend	
chicken	star	greek	option	go	peopl	look	food	
got	place	fresh	dessert	first	one	like	alway	
realli	great	delici	delici	year	around	old	servic	
tri	reason	tri	meal	visit	just	friend	staff	
pretti	fast	dish	ice	last	area	littl	amaz	
lunch	qualiti	soup	dinner	went	walk	local	awesom	

The second step is to create two classification models based on the features probability distributions for all documents. The first classifier that is fitted is the baseline model, NB classifier, where Table 2 shows the prediction performance. By using the test set, it can be evaluated that the precision rates are 66% for the *negative* class, 59% for *average* class and 66% for *positive* class. The classifier does not seem to be biased due to the imbalanced data since downsampling is applied on the training set. Moreover, the overall F-measure, or so-called goodness-of-fit, is 63% which is computed by aggregating the precision and recall rates to derive the F-measure per class, and compute the average over all F-measures. On this basis, the NB classifier seems to be a decent model to use in the remainder of this study. However, this needs to be verified by comparing with the RF classifier's performance.

Table 2: Classification performance for the NB and RF classifiers

Naive Bayes	Negative	Average	Positive
Precision rate	0.66	0.59	0.66
Recall rate	0.59	0.54	0.79
F-measure	0.62	0.56	0.72
Random Forests	Negative	Average	Positive
Precision rate	0.61	0.50	0.73
Recall rate	0.61	0.53	0.70
F-measure	0.61	0.51	0.71

To derive the best RF classifier, the number of (B) bootstrap samples is determined first in which the selected m predictors are set to $\sqrt{17} \approx 4$. Figure 3 shows that the OOB error approximately stabilizes at

$B = 1,000$ where the error is reduced from 59% to 38%. Thus, the best RF model is derived with 1,000 bootstrap samples. Moreover, to select the best number of m predictors, Figure 3 shows the OOB error for a set of RF models with each different values of m while $B = 1,000$ is constant. The error is minimized at $m = 4$ which is equivalent to the rule of thumb value. Therefore, $m = 4$ is initialized to decorrelate all trees and minimize the overall variance. Hence, the best RF classifier is fitted with $B = 1,000$ and $m = 4$.

The prediction performance of the derived RF model is shown in Table 2 which is computed on the test set. In terms of the precision rates, the classifier predicts 61% of the *negative* class, 50% of the *average* class and 73% of the *positive* class correctly among the total number of predictions. The overall goodness-of-fit or F-measure is 61% which is 2% lower compared with the NB classifier. Even though both classifiers perform equally well, the NB classifier is the simplest and yet more accurate in comparison with the RF classifier for which the NB model is used in the remainder of the analysis.

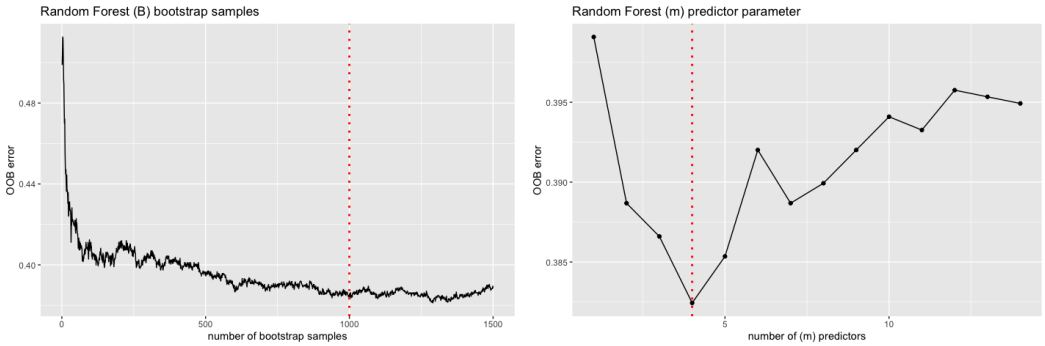

Figure 3: The OOB error over the number of B (left) and m predictors (right)

As determined, the NB classifier is selected to be used to analyse the most impactful diner factors on the predicted sentiment in Yelp's star-rating. Figure 4 shows the importance of features per class which observes that the *negative* ratings are mostly affected by the diner's level of service with the highest average error ratio of 1.18 followed by the food quality with 1.08 and breakfast menu with 1.07. Even though the location seems to be less important than food quality and breakfast menu on average, it can be indicated that it is overall more important since its 95% confidence interval strictly contains positive error ratios compared to food quality and breakfast menu. The other features are on average and overall more or less equally important. For *average* ratings, it seems that the error ratios among the features do not differ a lot. On average, food quality obtains the highest error ratio with 1.07 but contains negative values in its interval. Opening hours obtain the second highest error ratio with 1.06 and its confidence interval remains positive. Hence, the opening hours is the most important and stable feature compared to the others in the *average* class. Lastly, for *positive* ratings, food quality has the highest average and overall error ratio where its average error ratio is 1.28. Guest experience, level of service and opening hours are the other features that seem to be important due to their positive error ratios. All other features are more or less unimportant for the *positive* class prediction since the average error ratios are close to 1 and intervals contain negative values.

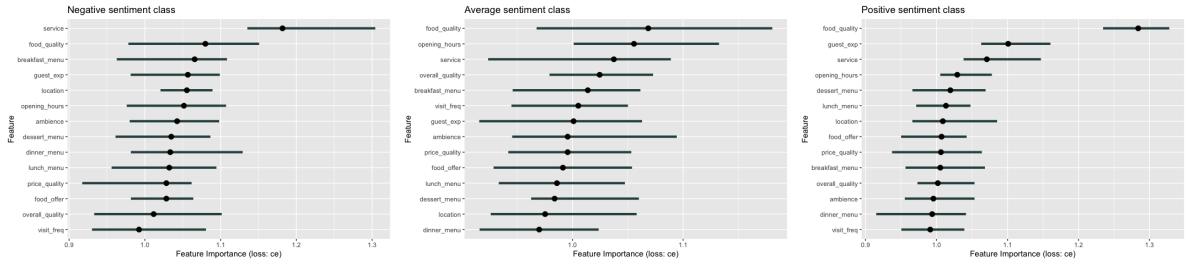

Figure 4: Permuted feature importance per sentiment class

5. Conclusion and Discussion

The purpose of this paper was to uncover latent features in sets of reviews from Yelp related to local diners and classify reviews in sentiment rating classes to provide an answer for the formulated research question: "*What diner factors are most impactful on the predicted sentiment in star-rating on Yelp?*" This study is performed by creating a Latent Dirichlet Allocation (LDA) model in combination with two classification models, Naive Bayes (NB) and Random Forest (RF), where the best classifier is selected to be interpreted by using Permutation Feature Importance (PFI).

To summarize, the topic extraction procedure resulted in 14 uncovered and distinctive features where the NB classifier was used to identify the most impactful features on class predictions. The results have shown that the diner's level of service is the most important feature for the predicted *negative* sentiment in star-ratings followed by the diner's location. Moreover, food quality and the breakfast menu are also important to some extent but these results are not fully stable. The most important feature for the predicted *positive* sentiment class, on the other hand, is the diner's food quality followed by the guest experience, level of service and opening hours which are all stable in their results. Thus, the predicted *negative* and *positive* sentiment in star-ratings for diners are both influenced by the level of service. Even though the magnitude of the influence cannot be derived from these results, by using common knowledge, it could mean that it is more likely that customers give *negative* star-ratings when experiencing low level of service and *positive* star-ratings when the service exceeds their expectations. The diner's food quality could have the same effect since it is important for both the predicted *negative* and *positive* sentiment as well. However, for this feature, the variable importance for the *negative* class is not stable for which it is not fully reliable. For the *average* sentiment class, the error ratio does not differ a lot in magnitude among all features where the opening hours is the only feature that is important to some extent and is stable. While food quality is on average more important, this feature is not stable. These results seem to be reasonable since, in general, individuals provide *negative* or *positive* ratings based on their expectations and experiences; meaning, by common knowledge, they are willing to pay more but expect higher food quality compared to home cooked meals and some level of service in return. This is also inline with the *average* ratings and the importance of the features; meaning that when all expectations are met without unexpected surprises, customers are less likely to give *negative* nor *positive* ratings.

In conclusion, the latent topic extraction and classification models can be a reasonable framework to predict and assess important factors for restaurants, and not only for diners, on the sentiment in star-ratings. Yelp is able to provide valuable insights to local diners and other restaurant owners in what features drives the customers negative or positive rating on the platform. This helps the business owners to be aware of the factors that are important for receiving higher ratings and simultaneously increase their revenue. However, these findings must be taken with some consideration since the dataset at hand is collected between 2005 and 2014 which might be outdated and some subjective model-building decisions were made such as the merging of the extracted topics to obtain distinctive features. For further studies, it might be interesting to explore more advanced classification algorithms, such as neural networks, when a larger number of latent topics are extracted to uncover more complex relationships and to explore the magnitude of factors' effect on the sentiment classes.

References

Anandarajan, M., Hill, C., & Nolan, N. (2019). *Practical Text Analytics* [ISBN: 978-3-319-95663-3]. Springer.

Blei, D.M., Ng, A.Y., & Jordan, M.I. (2003). *Latent Dirichlet Allocation*. Journal of Machine Learning Research 3, 993-1022. http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

Breiman, L. (2001). *Random Forests*. Machine Learning, Vol. 45. doi:10.1023/A:1010933404324

Fisher, A., Rudin, C., & Dominici, F. (2019). *All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously*. Journal of Machine Learning Research, Vol. 20, No. 177. arXiv:1801.01489

Gelfand, A.E. (2000). *Gibbs Sampling*. Journal of the American Statistical Association, Vol. 95, No. 452. doi:10.2307/2669775

George, C.P., & Doss, H. (2018). *Principled Selection of Hyperparameters in the Latent Dirichlet Allocation Model*. Journal of Machine Learning Research, Vol. 18, No. 162. http://jmlr.org/papers/v18/15-595.html

Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The elements of Statistical Learning* [ISBN: 978-0-387-84857-0]. Springer.

Huang, J., Lu, J., & Ling, C.X. (2003). *Comparing Naive Bayes, Decision Trees, and SVM with AUC and Accuracy*. IEEE International Conference on Data Mining, Vol. 3. doi:10.1109/ICDM.2003.1250975

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning* [ISBN: 978-1-461-47137-0]. Springer.

John, G.H., & Langley, P. (1995). *Estimation Continuous Distributions in Bayesian Classifiers*. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. arXiv:1302.4964

Khan, M.R. (2017). *Cascading Behavior in Yelp Reviews*. Proceedings of ACM Conference (Conference'17). arXiv:1712.00903

Lowd, D., & Domingos, P. (2005). *Naive Bayes Models for Probability Estimation*. Proceedings of the 22nd International Conference on Machine Learning. http://aiweb.cs.washington.edu/ai/nbe/nbe_icml.pdf

Luca, M. (2011). *Reviews, Reputation, and Revenue: The Case of Yelp.com*. Harvard Business School. https://www.hbs.edu/faculty/Publication%20Files/12-016_a7e4a5a2-03f9-490d-b093-8f951238dba2.pdf

Nguyen, D.Q., Billingsley, R., Du, L., & Johnson, M. (2013). *Improving Topic Models with Latent Feature Word Representations*. Association for Computational Linguistics, Vol. 3. arXiv:1810.06306

Pang, B., Lee, L., & Vaithyanathan, S. (2002). *Thumbs up? Sentiment Classification using Machine Learning Techniques*. In ACL-EMNLP. arXiv:cs/0205070

Potamias, M. (2012). *The warm-start bias of Yelp ratings*. Groupon Research. arXiv:1202.5713

Raftery, A.E., & Lewis, S. (1991). *How Many Iterations in the Gibbs Sampler*. University of Washington. https://pdfs.semanticscholar.org/0daf/54c4b59fd2c362de822de0ffdab84f49c6fd.pdf