Classifying Food and Beverage Clients

Andrés Potapczynski, Gaurav Chawla, Jason Kuo, Nanshan Li

May 13, 2019

Abstract

We present a novel machine-learning classification process for food service establishments based on their webpage contents. This process involves parsing the webpage, defining a vocabulary for the task, filtering out outlier HTMLs, engineering features that capture the semantics of the industry and applying a different classifier depending on the type of analysis.

Keywords— webscraping, NLP, feature reduction, multi-label classification

1 Introduction

In this project, we constructed a series of HTML classifiers to determine the food and beverage category and sub-category of different clients based on its web presence. The categories are based on establishment type: Restaurant, Grocery / Supermarket, Bar or Liquor Store and the sub-categories are based on different criteria for each: Cuisine Type for Restaurants, Organic vs Not Organic for Grocery / Supermarket, and drink category (beer, wine, cocktail) for both Bar and Liquor Stores. To construct these classifiers we manage close to 1 Million unstructured webpages from close to 200K clients. Additionally, we employed a novel approach to separate the informative HTMLs based on the construction of several meta-features, data visualizations and an anomaly detector. Moreover, we constructed a dense matrix representation from the contents of these webpages that preserved the semantic nature of the data but that avoided the high-dimensionality problems of the usual one-hot encoding or words. Finally, we tested different classifiers for each type of category and sub-category each with specific metrics depending on the task at hand (multi-class, binary classification, multi-label or multiple regression). We build these ML models with the goal of improving and scaling the classification process for our sponsor Neoway.

Neoway delivers insights to customers based on firmographic data publicly available from different sources on the web. It is especially relevant for Neoway to extract useful data features for the food and beverages industry. The data scientists at Neoway have worked extensively on this type of task but mostly based on subjective text mining methods, such as building custom lists of words to perform regex comparisons. We are training different machine learning algorithms to automate and generalize the task of determining the segments of different establishments supplied by that industry, based on website scrapped textual data and segments labels provided by Neoway. For example, we may be able to determine that a given establishment is of the type 'restaurant' and cuisine 'Mexican', based on its website that has a predominance of words such as 'taco' and 'tortilla'. The aim is that the relations between words and segments are captured naturally by a classification model, instead of being built in the usual ad hoc manner discussed above.

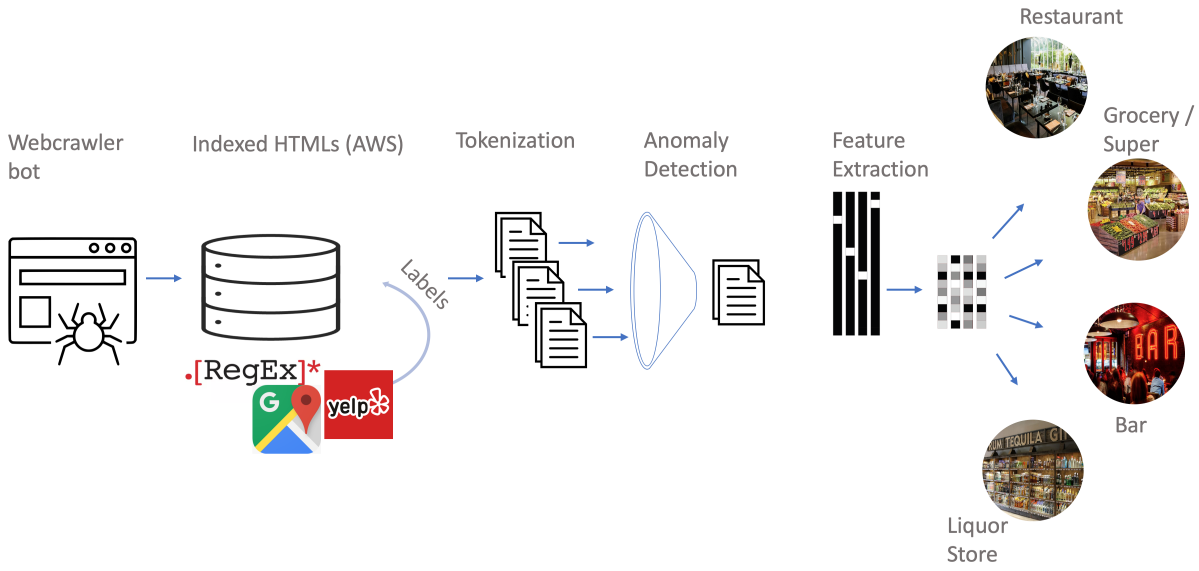

Figure 1: Project Overview

2 Methodology

2.1 Overview

We established a pipeline to utilize the crawled HTML text data to train four classification models to provide Neoway contextual labels to segment their client's webpages. Below we will highlight the pipeline steps in two parts - feature engineering the raw input and creating our ML models for labeling food-service entities.

2.2 Data

The process that generated the data for the project is the following. First, a web crawling bot gathered and indexed the HTMLs from various webpages that were linked to a client. For example, a client can be WholeFoods and then, the bot starting possibly at www.wholefoods.com, crawled the sub-urls linked in the home page and saved that information in the cloud (AWS bucket). After this, some of the clients in the cloud where classified into the categories and sub-categories based on either metadata from sources like Google and Yelp or by the use of RegExs based on keywords depending the category and subcategory. For example, some restaurant's cuisine type information used Google Maps whereas the organic classification of a supermarket was dependent on keywords like: fresh, organic, etc.

3 Feature Engineering

3.1 Anomaly Detector

After exploring the HTML data scraped by a web crawler targeting food-related websites, we created an Anomaly detector to reduce the amount of unusable data to pass into the model. We determined several factors that could make a HTML page's data unusable, which includes stub webpages (sections of websites with too few content like an "About Us" page), mostly Javascript/animation driven of which our

text scraping will not be able to parse, or contains too few food-related words when compared against a dictionary of common ingredients.

In addition, we employed an Isolation Forest to filter lengthy pages which would be difficult to process or too unusual from the rest of the data set. Unlike other outlier detection techniques which attempt to profile the normal data points to construct outlier bounds, Isolation Forest explicitly looks to detect how anomalous a data point is. The algorithm works with the premise that outliers can be individually partitioned out with fewer splits of a decision tree than a normal observation that is sitting well within the rest of the data population.

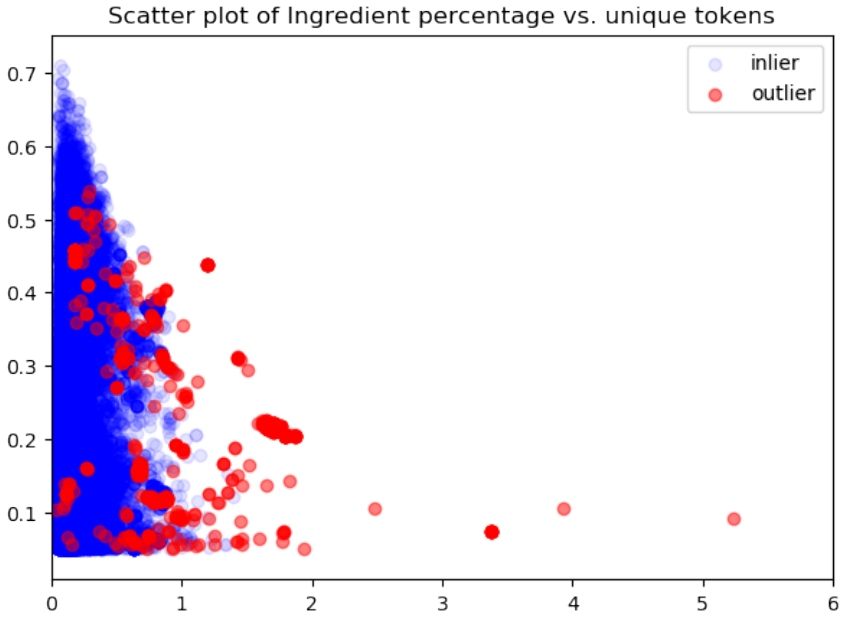

Figure 2: Isolation Forrest finds anomalies considering all meta-features

3.2 Tokenization

We utilized the Spacy package to tokenize documents and allocate part-of-speech (POS) tags as well as dependency markers for each token. We thus implemented a filter for the desired words, based on tags allocated by the Spacy model. The filter kept tokens that were tagged as adjectives, nouns and pronouns, while stripping determiners, personal and possessive pronouns, adverbs and comparative adjectives.

The tokens were lemmatized using WordNetLemmatizer in NLTK instead of using the Spacy model, as we wanted the stem of each word independent of its context and POS tag in the document. All words were converted to lowercase in our tokenization process to remove sentence structure.

We employed term-frequency, inverse document frequency (TF-IDF) as a technique to trim down the feature word list further. The TF-IDF metric identifies the relative importance of a word in a document in a larger corpus. By adjusting the min and max document frequency bounds, we can remove the most commonly used words and most rare used words. We wish to remove the most common words as these are most likely stop words across the documents in addition to the most rare words as they will be tougher to generalize in our model. Based on extensive analysis of our corpus, we set the frequency bounds at 1% and 65% after reviewing the word list.

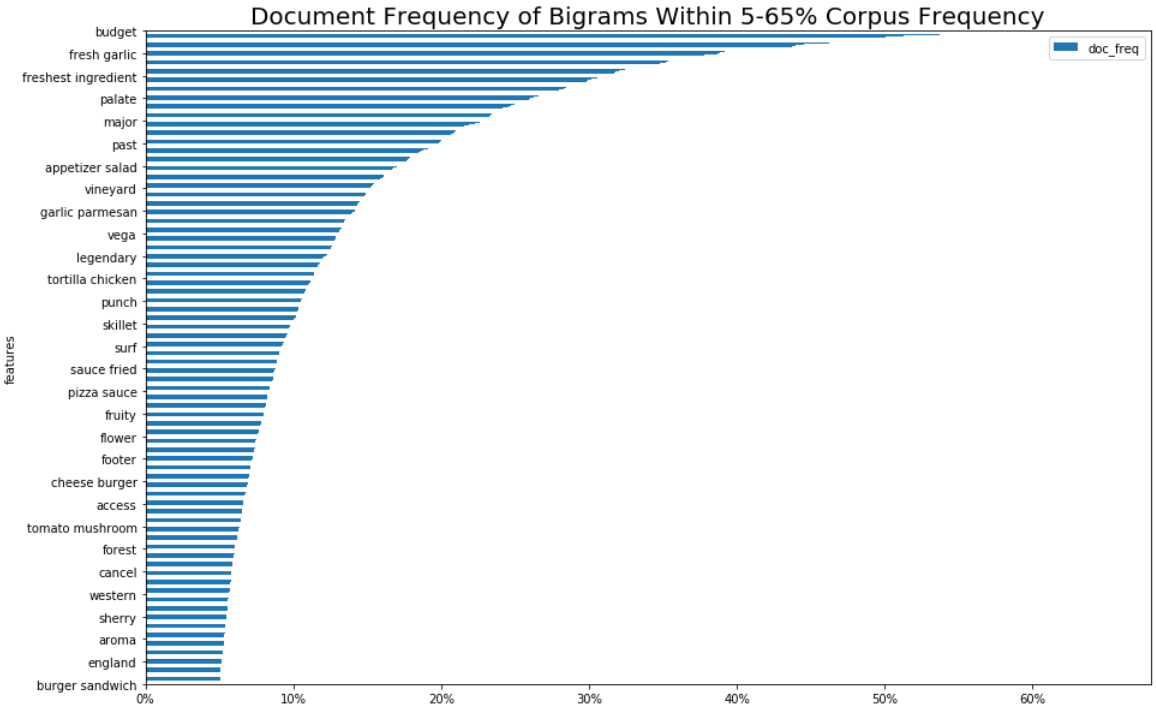

Figure 3: Example of an Document Frequency of Corpus Words

We identified the following corpus-specific stop words by their associated document frequency: menu (93%), home (75%), contact (73%), special (71%), hour (69%), food (68%), restaurant (67%). Similarly we identified the corpus-specific rare words (some sample words all under 1% frequency): salami linguica, orange blossom, beef sauteed, meatball sauce, cater, focaccia bread, wine reduction, croquette, pesto cream. We removed the words from both of these lists from our final feature set.

3.3 Feature Engineering

We looked to improve upon the TF-IDF approach of using the word tokens directly in the classifier models through various feature engineering techniques. The goals was to improve robustness of the model since the tokens were only a sparse representation of the HTML pages of which alternative techniques can allow for page representations that share characteristics among similar web pages. Moreover, by extracting dense features out of the sparse one-hot encodings on the words, we were able to reduce the dimensionality of the problem by at least 90%. This is quite advantageous for the classification task since it allowed us to iterate through several alternatives fast.

3.3.1 LDA

Summarizing the pages as a collection of sub-topics was one approach explored. Representing the HTML pages as documents, we employed Latent Dirichlet allocation (LDA) on corpus of data [1]. Intuitively, LDA utilizes Dirichlet distributions to represent a collection of topics within a document and the words associated with each topic. An example of the topics uncovered are below:

Topic #1: roll rice gallery online menu order shrimp tuna spicy soup

Topic #2: chicken rice bean cheese beef tortilla sauce taco shrimp mexican

Topic #3: chili website grill restaurants privacy information copyright policy reserved

Topic #4: restaurant mexican menu food review cuisine location home good

Topic #5: menu grill home restaurant food burger beach contact great

LDA is an unsupervised technique, but having labels at our disposal, it was natural to contrast the relationship between the LDA topics and the cuisine labels. An interesting example is below. We see that the NeowayIDs that are part of the American cuisine on average are related to certain topics more closely.

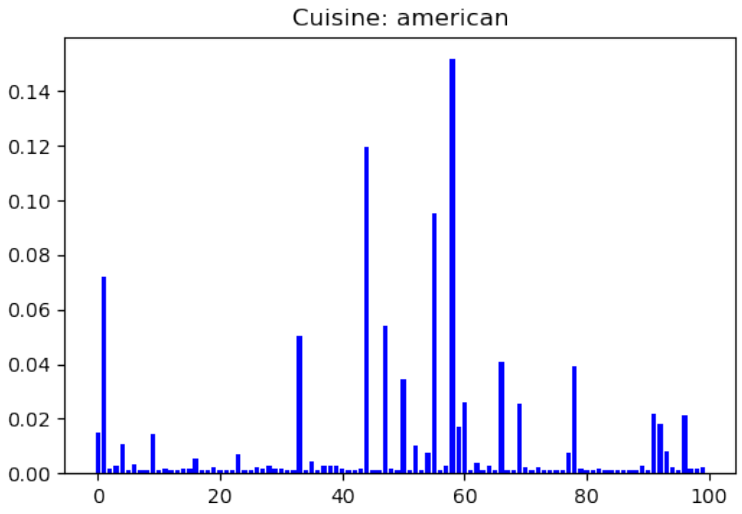

Figure 4: Example of the mean American distribution over topics in LDA

3.3.2 NMF

An alternative to LDA in generating topics is non-negative matrix factorization (NMF). The approach factorizes the matrix components of TF-IDF embeddings into a lower dimension vectors by gradually reducing

the loss introduced by the decomposition of the original embedding. We have found both methods perform similar but noticed certain labeled cuisines were more easily identified by the topics in the LDA method than the NMF method. We were exploring methods such as topic coherence to more quantitatively assess the number of topics to use and performance against similar models.

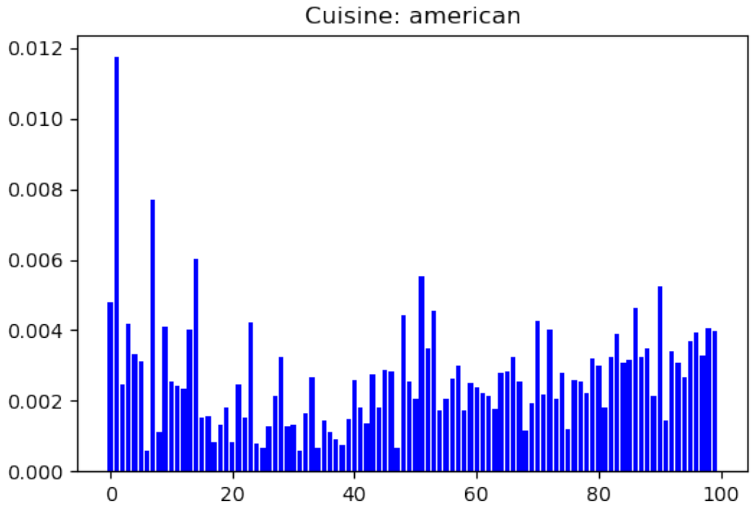

Figure 5: Example of the mean American distribution over topics in NMF

In contrast to LDA we see how the NeowayIDs that are part of the American cuisine are related to many topics. Thus, it becomes hard to interpret the topics learned by NMF as cuisine topics.

3.3.3 Doc2Vec

LDA and NMF are topic models based on the simple bag-of-words representation of the data. This representation does not take into consideration the *context* into which each of the word occurs. Where, in this case, the *context* is defined by the other words that are part of the HTML. A popular method for capturing the relationship between words that are likely to co-occur together is *word2vec* [2]. This model, intuitively, will generate a vector space that will place common occurring words together. This is desirable, since words like *cheese* and *pizza* will be mapped closed together - which did not occur for the TF-IDF representation. Yet, having a vector representation of words is not directly useful for classifying HTMLs into a given establishment or subtype. However, the idea behind *word2vec* can be extended to not only embed related words together but also documents. Hence, if we considered each NeowayID as a document, then *doc2vec* [3] can help us map each ID into an embedding that takes into consideration that words that these HTMLs have in common.

In general, the *doc2vec* algorithm generates this semantic embedding by maximizing the average log probability that a word occurs given the surrounding words in that document (ngram) and the document tag. Mathematically,

$$\max \frac{1}{T} \sum_t \log p(w_t | w_{t-k}, \ldots, w_{t+k}, tag)$$

where w_j is the jth word in the ngram of size $2k$ and tag represents the document tag from where the ngram was taken from. This is done via a simple neural network that has the following architecture:

We applied *doc2vec* to the NeowayIDs in the sample where the notion of document was defined by the categories present in each analysis. For example, for the *establishment classification* task there were four types of documents based on each establishment type (namely, restaurants, bars, liquor stores and supermarket / grocery). Or for the *cuisine classification*, where each cuisine determined a different document. To show

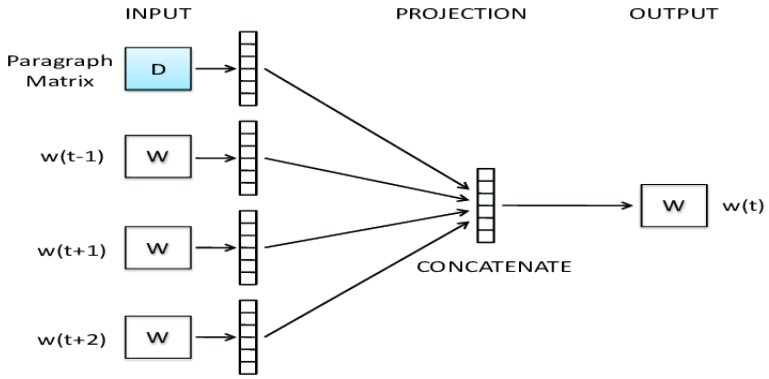

Figure 6: Doc2Vec Architecture

what relationships *doc2vec* uncovered for this last task we plotted the center of each cuisine determine by taking the mean of all the NeowayIDs that belong to that document.

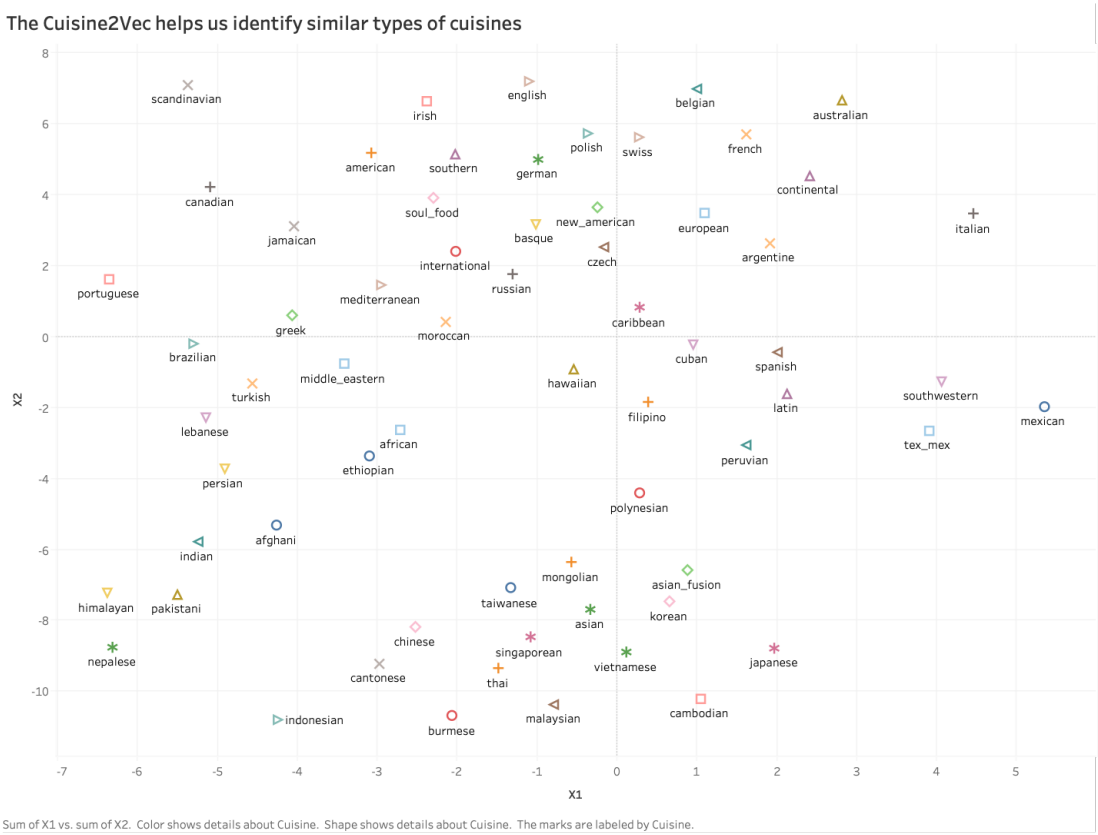

Figure 7: Cuisine Centroids

As we can see above, *doc2vec* is able to uncover which cuisines are closely related to others. This previous plot can help us create broader groupings of cuisines.

4 Classification

The process can be visualized in Figure 8 below:

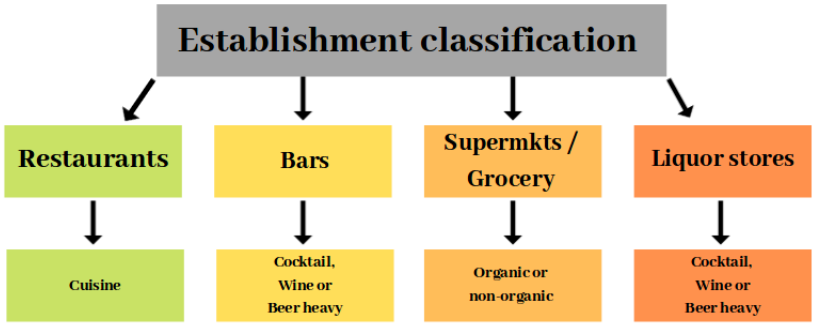

Figure 8: Summary of classification process

This is a two-step classification problem, with the first step being the classification of establishments into Restaurants, Supermarkets / Grocery Stores, Bars and Liquor Stores. After the establishment type is determined in the first step of the classification, the data is then fed into one of three classifiers based on the determined establishment. These are:

- Cuisine Classification (for Restaurants)
- Organic / Non-organic Classification (for Grocery Stores)
- Drink-type Classification (for Bars and Liquor Stores)

We were provided with a set of close to 200K labelled data, with which we used to evaluate the accuracy of our models. We filtered almost half of the data due to inconsistencies, such as bars with cuisines, restaurants with bar-type classification to form our data set and multiple Neoway ids for the same establishment.

4.1 Establishment Classification

The first task in the pipeline is to determined the establishment type of a given webpage. This problem is a multi-class classification problem. We trained a Multinomial Naive-Bayes classifier for this task. The features created after word tokenization were used as an input matrix for the Multinomial Naive-Bayes Classifier.

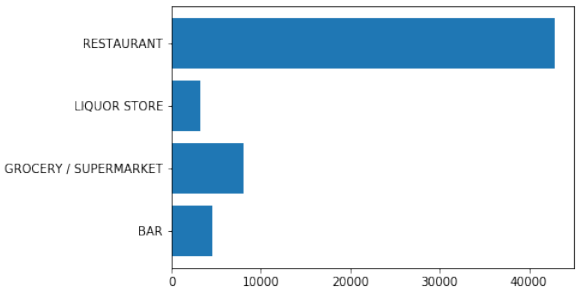

Figure 9: Count by Establishment Type

Establishment Type	Words (coefficient)
Restaurant	onion (5.38), tomato (5.34), salad (5.3), order (5.19), location (5.14), pizza (5.14), sauce (5.06), restaurant (4.99), cheese (4.94), chicken (4.87)
Grocery / Supermarket	brand (4.9), saving (4.88), recipe (4.77), department (4.71), pharmacy (4.65), product (4.54), weekly (4.51), grocery (4.42), coupon (4.35), store (3.69)
Bar	chicken (5.40), drink (5.34), facebook (5.32), music (5.31), grill (5.21) sport (5.16), night (5.10), beer (5.06), cheese (5.04), event (4.82)
Liquor Store	product (5.38), whiskey (5.30), event (5.19), bottle (5.16), tasting (5.04), store (4.43), spirit (4.32), beer (4.18), liquor (3.86), wine (3.59)

As it can be seen above, we have a strong majority class. To address this we stratified out train and test set splits. The confusion matrix below in Figure 10 shows that most of the predictions fall on the diagonal. There is a higher bias towards Bars being classified as Restaurants. The error rate between Bars and Restaurants can be attributed to words like 'Chicken' and 'Onion' that are among the highest contributors for both the categories.

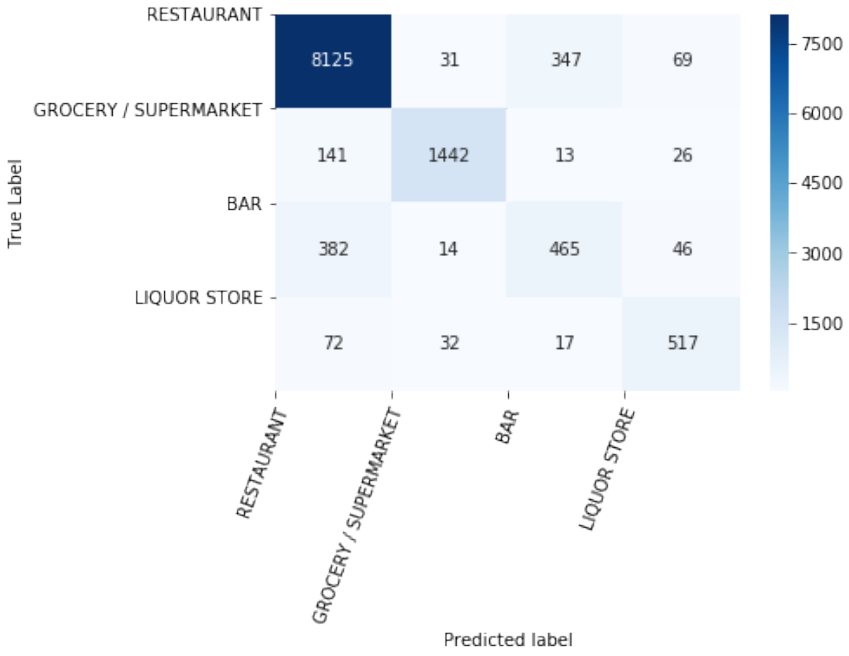

Figure 10: Confusion Matrix - Establishment Classification Prediction

Overall F1 Score: 90 %		
Establishment	**Precision**	**Recall**
RESTAURANT	93 %	95 %
GROCERY / SUPERMARKET	95 %	89 %
BAR	55 %	51 %
LIQUOR STORE	79 %	81 %

The next table shows the 10 most significant words that help classify each type of the establishment.

4.2 Cuisine Classification

The labelled dataset has a total of 99 possible cuisines, and these labels are not mutually exclusive. About half of the labelled restaurants only have 1 cuisine label (34615 / 64739) and thus we decided to treat this as a multi-label classification problem, where labels are not mutually exclusive. Below is a sample of the possible cuisine labels from the dataset:

afghani, african, american, argentine, asian, asian-fusion, australian, bakery, basque, bbq, belgian, brazilian, breakfast-brunch, brew-pub, buffet, burgers, burmese, cajun-creole, cambodian, canadian, cantonese, caribbean, chicken, chinese...

4.2.1 Limitations of data set & possible orthogonality

There are multiple types of labels within the data set. Upon exploration, it was found that the cuisine labels could be classified into 4 main types:

1. Regional cuisines, describing the place of origin of the food (E.g. Afghani, African, American)
2. Food type, describing food items and ingredients (E.g. burgers, chicken, crepes, fish & chips)
3. Dietary restrictions (E.g. kosher, gluten-free, halal, vegetarian)
4. Restaurant Type (E.g. bakery, breakfast & brunch, brew pub, buffet)

The types of cuisine labels can be seen as orthogonal to one another (e.g. pizza can be American or Italian style, and there can be vegetarian or halal options at both types of restaurants). Out of the 99 different cuisines in the data set, 66 were regional, 20 described the foods sold at the restaurants, 6 indicated options for special dietary needs and 7 were restaurant types. The full list can be found in Appendix. We decided to focus on regional cuisine labels are that was the most common way to classify a restaurant.

4.2.2 Exploration of regional cuisine labels

Looking at the regional cuisine labels, some labels suggested a possible hierarchical structure. There are regional labels such as Asian food for food from Asia (a continent), national labels such as Chinese food for food from China (a country within Asia) and sub-national labels such as Cantonese (a region in China). However, some exploratory analysis indicate that such a hierarchy is not present within the labelled data, as there are less than a third of restaurants labelled Chinese labelled Asian as well (see Figure 11 below).

After exploring the co-occurrences of labels through distance matrices (using Jaccard's Index) and looking at possible nesting structures, there was no easy way to create a hierarchy within the cuisines. As such, we treated the classification task as a one-step multi-label problem.

4.2.3 Classification models: An Overview

We explored 3 main approaches to multi-label classification in this task. Binary Relevance, Classifier Chains and ML-KNN.

In Binary Relevance, an ensemble of binary classifiers, one for each possible label, is trained. The occurrence of each cuisine is assumed to be independent and uncorrelated with the occurrence of other cuisines, which might not entirely be the case.

In Classifier Chains, there is also an ensemble of binary classifiers, except that the output of each classifier is added to the input of the next classifier in order to take into account possible correlations between labels. One limitation of this is that the order of the classifiers are randomly generated, and thus an ensemble of

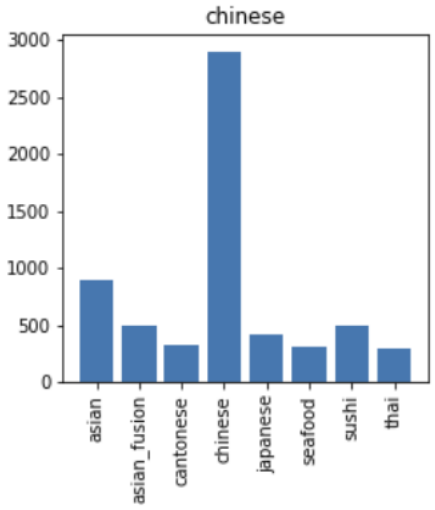

Figure 11: No label hierarchy present in data set

classifier chains need to be trained in order to provide a better result. There are a possible n! classifier chains for a multi-label problem of n labels.

Lastly, we also tried multi-label k-nearest neighbors (ML-KNN), where the k-nearest neighbors of each data point and a combination of MAP estimated is used to determine the multiple labels attached to the point.

We did a preliminary test of all three methods by classifying the 20 cuisines in the data set with greater than mean number of observations.

Among the 3 methods, ML-KNN performed the worst across all metrics, presumably due to the fact that the hierarchical structure is not well defined in the labels of the data set. An ensemble of 10 binary relevance classifiers performed better than an ensemble of 10 classifier chains when the same binary classifier is passed, and thus we decided to take the approach of finding the best performing binary classifier to pass to Binary Relevance.

We did not explore the usage of Label Powerset, where each combination of labels in the dataset was treated as a unique classification result, as we did not want to create additional distinctions between similar cuisines due to the lack of a consistent hierarchy in the regional labels.

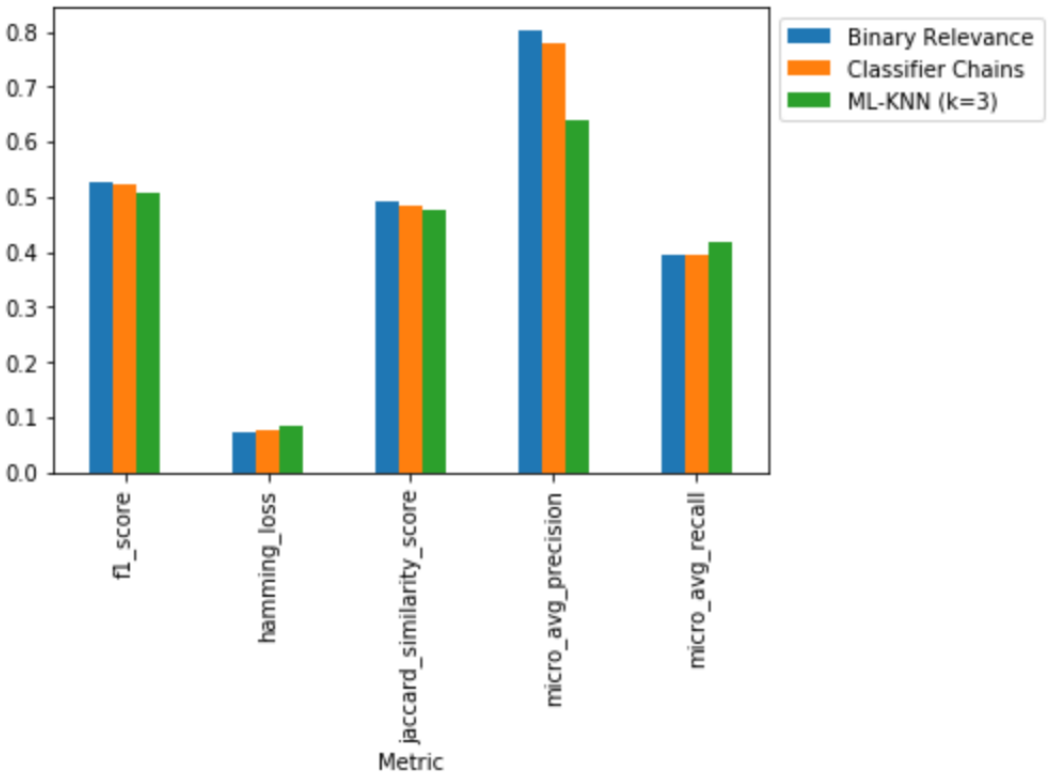

Figure 12: Performance of all 3 on top 20 cuisines

4.2.4 Evaluation Metrics

Since our data set is imbalanced and it is a multi-label classification problem, we have chosen a few evaluation metrics that will provide more insight as to the actual performance of the model. They are the F1 score, hamming loss and Jaccard similarity score.

The F1 score is derived from 2 other metrics: the micro-average precision and the micro-average recall. The micro-average precision describes the proportion of positive identifications that are actually correct, while the micro-average recall describes the proportion of actual positive values that are identified correctly. The formula for both metrics are presented below.

$$\text{Micro-average Precision}(D) = \frac{\sum_{c_i \in C} TPs(c_i)}{\sum_{c_i \in C} TPs(c_i) + FPs(c_i)} \quad \text{Micro-average Recall}(D) = \frac{\sum_{c_i \in C} TPs(c_i)}{\sum_{c_i \in C} TPs(c_i) + FNs(c_i)}$$

The F1 score can be interpreted as a weighted average of the precision and recall, where the relative contribution of precision and recall to the F1 score are equal. The F1 score ranges from 0 to 1, where 1 indicates the best possible performance. The F1 score is a measure over the complete data set and its classes, since the micro-averaged precision and recall are used. The formula for the F1 score is:

$$F_1 = \frac{2(\text{precision} \cdot \text{recall})}{\text{precision} + \text{recall}}$$

The Hamming loss describes the fraction of labels that are incorrectly predicted. A lower loss indicates a better performance of the model. Hamming loss $= \frac{1}{NL} \sum_{i=1}^{N} \sum_{j=1}^{L} \text{XOR}(y_{i,j}, z_{i,j})$

where $y_{i,j}$ is the target and $z_{i,j}$ the prediction.

The Jaccard similarity score is a measure of similarity between two data sets. In this case, it measures the similarity between the actual and predicted labels. For each data point, M_{11} is the number of cuisines

that were predicted and actually positive. A higher Jaccard similarity score indicates better performance. Figure x below shows the coefficients used in the computation of the index.

		A	
		0	1
B	0	M_{00}	M_{10}
	1	M_{01}	M_{11}

Figure 13: Coefficients used in the computation of Jaccard similarity score

Jaccard Similarity Score = $\frac{M_{11}}{M_{01}+M_{10}+M_{11}}$

4.2.5 Model results

We did a preliminary run of several models on the 66 regional cuisines using the Binary Relevance wrapper (logistic regression, logistic regression CV, decision tree, random forest, KNN, CatBoost (a categorical boosting model) [4]) and a multi-layer perceptron. Figure 14 shows the results from the run.

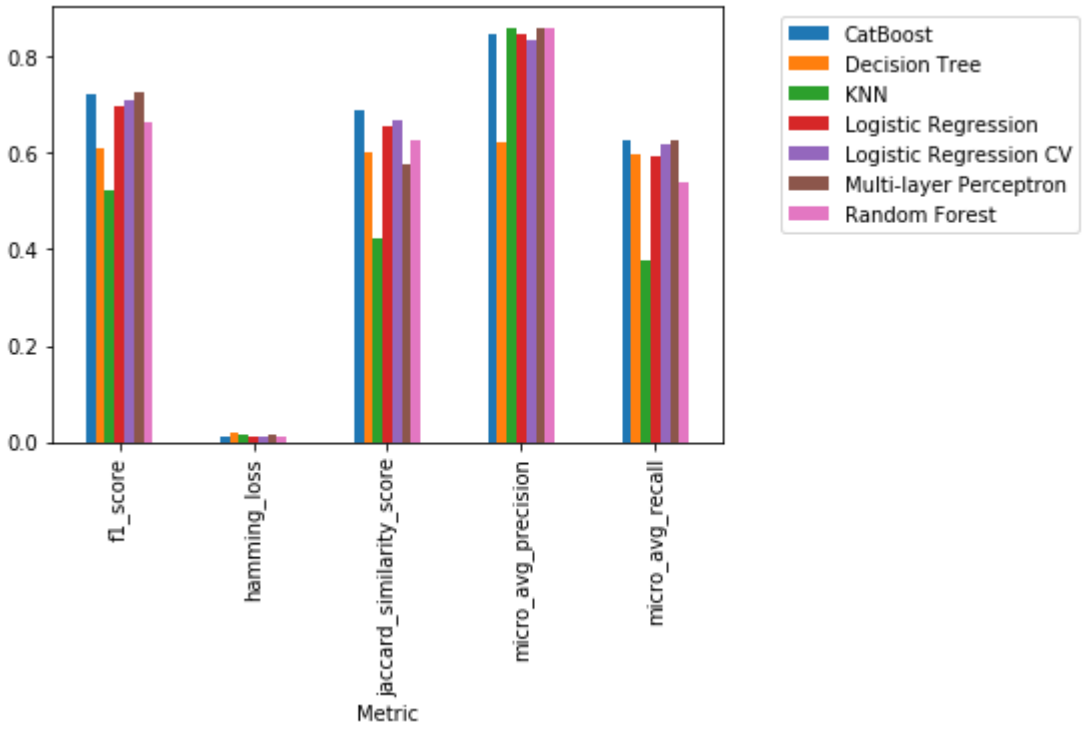

Figure 14: Results from preliminary run of models

Among all 7 models, the linear Logistic Regression model and CatBoost performs the best while multi-layer perceptron is close behind. CatBoost, while showing strong results, did not meaningfully improve upon the simpler Logistic Regression model. Although the multi-layer perceptron model learnt the majority classes really well, better than the other classifiers, it did not provide predictions for the minority classes.

We are investigating techniques to deal with unbalanced classes which may help improve the predictions of these cuisines. The Random Forest provides a significant performance boost over the Decision Tree classifier.

We also visualized the impact of using different derived features on the same Logistic Regression with Binary Relevance classifier on 66 regional cuisines. The features are LDA (100 topics), NMF (10 dimensions), Neoid2vec (100 dimensions), a horizontal stack of Neoid2vec and the 100 LDA topics, as well as the original one-hot-encoded features. Figure 15 below shows the results from this run.

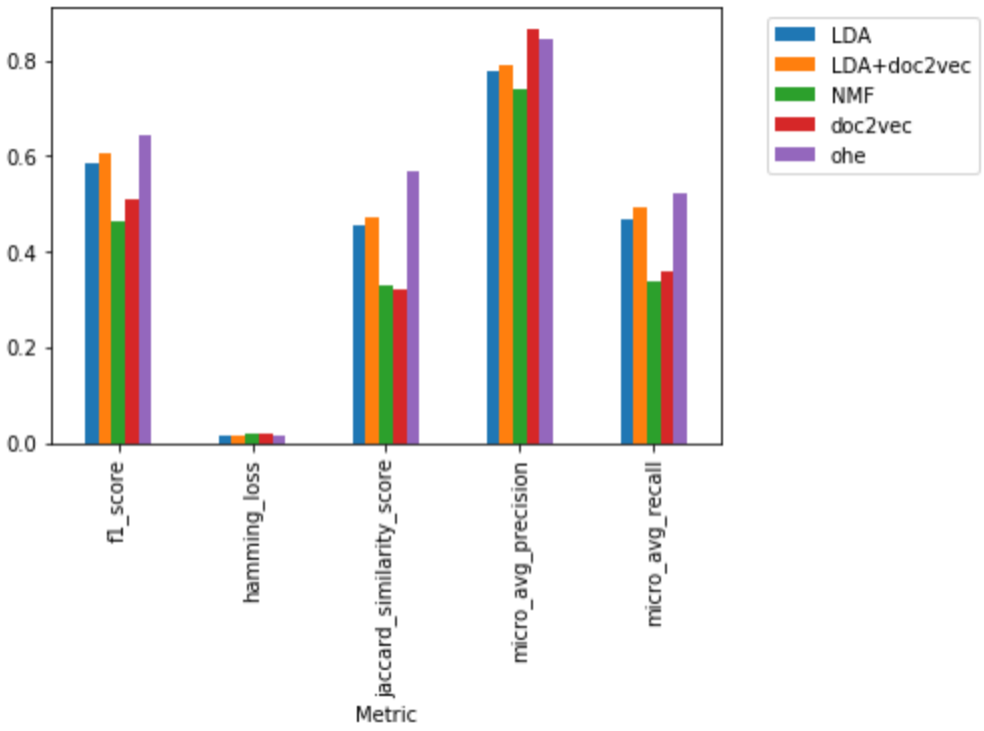

Figure 15: Performance of linear model using different features

Among all 5 feature types used, the one hot encoded features performed the best under the linear Logistic Regression Binary Relevance model. The NMF features performed the worst, probably due to the low dimensionality in the features. While LDA performed the best out of all 3 reduced features, adding doc2vec to the LDA topics also provided a boost in performance.

We also tested all the derived features on a non-linear model (Decision Trees) and obtained the following results in Figure 16.

Using a non-linear model seemed to improve on the relative performance of the doc2vec features. However, overall, the linear model did better than the decision trees, and the one-hot encoded features remained the best-performing out of all the features attempted.

Another point to consider is the dimensionality reduction achieved by using the LDA and doc2vec features. Instead of training the model on 1374 unique words (features), exponentially faster training times can be obtained by using the 100 LDA topics, the doc2vec vectors with 100 dimensions or the resultant 200 features when both of them are horizontally stacked. This would prove useful in utilizing more complex models with boosting (e.g. CatBoost, XGBoost).

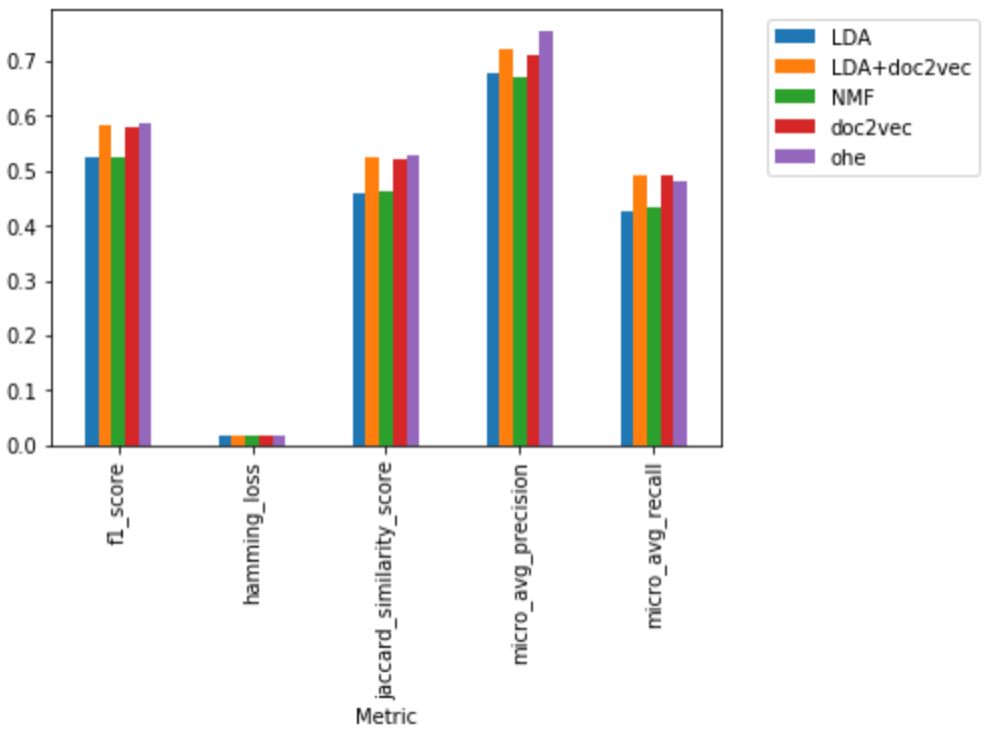

Figure 16: Performance of non-linear model using different features

4.3 Grocery / Supermarket Organic Classification

Once it has been determined that a given NeowayID's establishment is of the supermarket or grocery type, then the pipeline checks if this client contains (or focuses) on organic products. From all our classification tasks, this is the simplest one since we have two labels and thus we can model this as a binary classification problem. Yet, we also face an imbalance problem were the minority class only represents 11% of the sample.

We tested different models for the data matrix generated by *doc2vec*. For each model, we tried a set of hyperparameters values. From this effort we selected: (1) Ridge Logistic Regression with penalty equals to 1, (2) Decision Trees with a maximum depth of 10, (3) k-Nearest-Neighbors for 20 neighbors and (4) a Neural Network with the following architecture 64 ReLu - 128 ReLu - 64 ReLu - 2 Sigmoid. For each of the models we computed the test accuracy, precision and recall. Yet, we created a set of revisited metrics were we analyzed whether a misclassified NeowayID was actually organic or not (this was a tractable effort since the number of test observations was around 140-150). We did this only for our best performing model. Once we corrected for this *bad labels* we recomputed the metrics with a considerable improvement as seen below.

Different Model Test Results				
Models	Accuracy	Precision	Recall	Accuracy (R)
Logistic (1)	90.02%	14.34%	57.81%	
Dec Trees (10)	89.90%	37.21%	51.89%	
KNN (20)	92.08%	35.66%	74.80%	
NN (64-128-64)	92.32%	42.64%	74.83%	**97.42**%

4.4 Type of Beverage Classification

The premise for Beverage Classification is to be able to quantify a given bar's preference towards serving three types of beverages (Beer, Wine and Cocktail). For this task, we were provided the counts of the occurrences of each type of drink in the webpage. We normalized these counts across the three types of beverages and them that as target variables for Ridge Regression Models.

The normalization was done across the types of beverages for a given bar. That is, given Bar-A (Beer: 10, Wine: 20, Cocktail: 30) and Bar-B (Beer: 1, Wine: 2, Cocktail: 3) the normalization would assign (Beer: 1/6, Wine: 2/6, Cocktail: 3/6) to both these bars since they both have same proportion of menu allocated to the three types of beverages. We chose not to normalize across all the Bars since the Neoway web crawlers do not assign weights to the beverages across the bars. That is, if Bar-A has Beer:1 and Bar-B has Beer:10 that does not imply that Bar-B serves more Beer than Bar-A.

For the three types of beverages three independent ridge regression models with different ridge parameters (alpha) were used. The Ridge parameters were calculated after a grid-search on the training data using repeated K-Fold cross validation. The root mean square (RMSE) value was calculated from a Test data set which was not used to train the model. The graphs below show in the order of significance the words that contribute towards the type of beverage served in a bar. Looking at the words on the far right of the X-axis one can say with confidence that correct words are being picked for the corresponding Beverage.

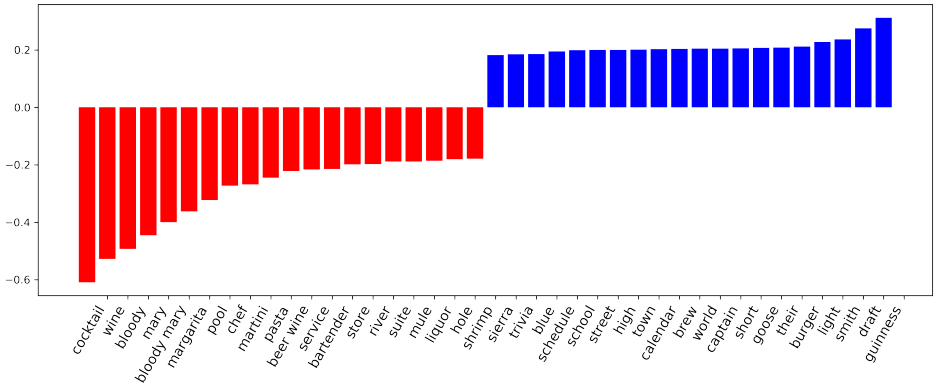

Figure 17: Significant Words for Beer

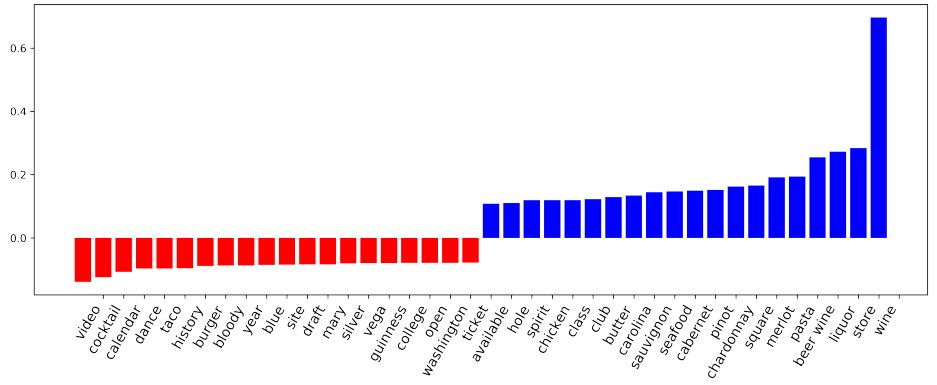

Figure 18: Significant Words for Wine

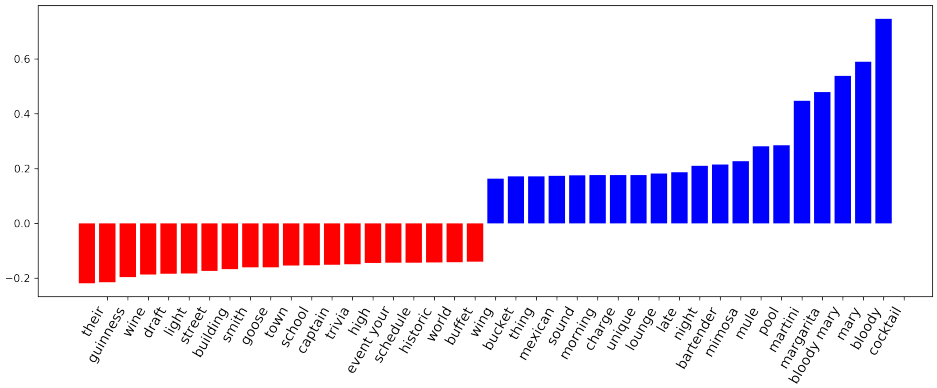

Figure 19: Significant Words for Cocktail

We analyzed the performance of the model on the test data set and the prediction agrees with most of the weights that were assigned in the labelled data. We also saw some instances where the predicted model was performing better than the existing process in place at Neoway. Below we show the prediction versus the labels of two bars: Cecilia Bar (with NeowayID: 052f927e-8208-4329-8c65-59c8a8b80fac) and Windward Tavern (with NeowayID: 09442e7a-07cf-49b7-b589-7114f7d2e521). The main webpage of each bar can be seen in the Appendix.

Bar	**Beer**	**Wine**	**Cocktail**	**Beer (pred)**	**Wine (pred)**	**Cocktail (pred)**
Cecilias	0.64	0	0.34	0.34	0.12	0.54
Windward Tavern	0.2	0.2	0.6	0.54	0.24	0.19

5 Conclusion

Our new pipeline provides several benefits over existing regular expression/manual labeling approaches. Our method derives domain knowledge from the data rather than requiring expert feedback, it is scalable in terms of new data updates and modular to accept different model approaches and, most importantly, it is robust to human error and noise in the webscraping dataset. Moreover, our pipeline presents a principled way to extract useful information out of an HTML and to separate them based on their information content. Additionally, the semantic dense representation of the data that we employ is able to reduce the computational overhead to 75-100x the cost of using a sparse one-hot encoding representation. Finally, our pipeline derives acceptable performance in each classification task.

6 Recommendations & Next Steps

We have a few suggestions to make additional improvements to the pipeline. First, we suggest that Neoway stores the tokenized data in a NoSQL DB in order to accelerate the model prototyping and facilitate the access to the data. Also, in terms of the representation of the data, we suggest the inclusion of state-of-the-art approaches such as ELMo or Bert and to adjust the number of topics in the feature reduction steps which could provide a performance that rivals or outperforms the one-hot-encoding while minimizing dimensionality overhead. Second, we recognize that the provided labeled data has no hierarchy. By providing an orthogonal label framework like partitions for geographic region/restaurant format/food types/dietary restrictions the spacial understanding of cuisines can improve the classifier approaches used. In this respect, the semantic presentation of the data based on Doc2Vec or the previously suggested can help derive natural clusters of cuisines. Finally, implementing a parameter tuning framework such as GridSearchCV from the scikit-learn package can further improve the classifier models at the cost of additional training time.

7 Acknowledgements

This research was supported by the Data Science Institute at Columbia University and Neoway. We would like to specifically thank Sining Chen from Columbia for always providing insightful comments and feedback on the quality of our work. Also Felipe Penha, Gabriel Alvim and Manoel Vilela from Neoway for providing guidance, resources and a constant flow of ideas.

References

[1] Blei D. M., Ng Y. A. and Jordan M. I. Latent Dirichlet Allocation In *Journal Of Machine Learning Research 3 (2003)*, pages 993-1022

[2] Mikolov T., Sutskever I., Chen K., Corrado G. and Dean J. Distributed Representations of Words and Phrases and their Compositionality In *Advances on Neural Information Processing Systems*, 2013c

[3] Mikolov T. and Le Q., Distributed Representations of Sentences and Documents In *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 2014. JMLR: W & CP volume 32, 2014.

[4] Anna V. D., Vasily E., Andrey G., CatBoost: gradient boosting with categorical features support In *Workshop on ML Systems at NIPS 2017*, California, USA, 2017.

A Bar Classification - Webpages

Figure 20: Cecilia Bar - Home page (focus on cocktail)

We see that in the given data labelled data set Cecilia Bar is focused more on beer and less on cocktails. The Ridge Regression model on the other hand predicts that the Bar is focused on cocktails. Upon inspecting the website and home page of the bar it is apparent that Cecilia is a cocktail bar and night club.

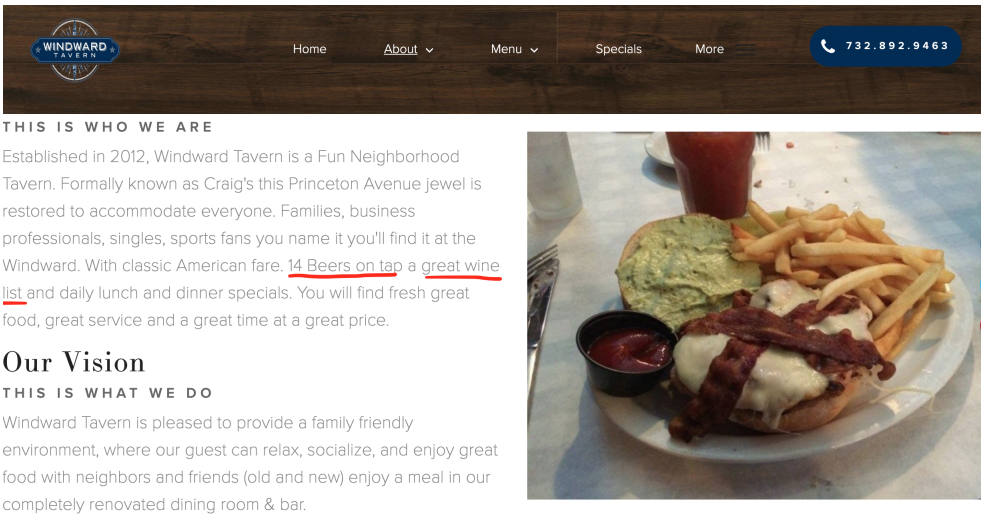

Figure 21: Windward Tavern - Home page (mentions Beer and Wine only)

We see that in the given data labelled data set Windward Tavern is focused more on cocktails and the ridge regression model on the other hand predicts that the bar is focused on beer and wine. Upon inspecting the website and home page of the bar it is apparent that Windward Tavern is a family Bar/restaurant with primarily beer and wine on the menu.