# Sentiment Analysis on Customer Reviews: Selection of the Best Feature Engineering and Machine Learning Models for Sentiment Prediction Problem

Benjamin Kan
*School of Continuing Studies*
*York University*
Toronto, Canada

Lingling Zhang
*School of Continuing Studies*
*York University*
Toronto, Canada

Sophie Lee
*School of Continuing Studies*
*York University*
Toronto, Canada

Amit Asghar
*School of Continuing Studies*
*York University*
Toronto, Canada

## Abstract

*This empirical research aimed to find the best feature engineering and machine learning algorithm in predicting sentiment from customer reviews. Combinations of four feature engineering methods with several of the most popular algorithms were tested; ensemble learning techniques as well as neural network were also tested and the results were then compared with base models. Results showed that the base models have already achieved superior predictive power. For the ensemble models, we found that stacking using Decision Tree and Logistic Regression as base models and KNN as meta model produces the best result. Multiple strategies for converting customer ratings to the customer sentiment target column were also investigated and discussed.*

## 1. Introduction

With the booming of online comments, the ability of predicting customer sentiment based on customer reviews becomes a prompt and practical need for enterprises and organizations of various sizes. This research aims to investigate the best machine learning algorithm in predicting sentiment from customer reviews.

A large empirical dataset from online review platform Yelp [1] was used. A total of four most popular feature engineering techniques - Bag of Word (BOW), Term Frequency–Inverse Document Frequency (TF-IDF), N-grams and Word2Vec were used.

Algorithms including Logistic Regression, Support Vector Machines (SVM) and Decision Tree were used. Different ensemble learning methods (Bagging, Random Forest, Boosting and Stacking) as well as neural network library, i.e. Keras, were further investigated so as to determine whether they could enhance performance.

## 2. Methodology

## 2.1 Data Collection and Preparation

The original dataset was obtained from Yelp (https://www.yelp.com/dataset). It consists of the following sub-datasets in json format:

- **business.json**: Contains business data including location data, attributes, and categories.
- **review.json**: Contains full review text data including the user_id that wrote the review and the business_id the review is written for.
- **user.json**: User data including the user's friend mapping and all the metadata associated with the user.
- **checkin.json**: Checkins on a business.
- **tip.json**: Tips written by a user on a business. Tips are shorter than reviews and tend to convey quick suggestions.

In this project, we mainly used the information in the business and review files.

The original review dataset has more than 6 million reviews. Due to the sheer volume of the data, we sampled our dataset to contain reviews for restaurants that are advertised to be specialized in Chinese cuisine. In order to select the reviews pertaining to Chinese restaurants, we performed the following filtering routines:

- Using the business dataset, obtain a list of business IDs which are Chinese restaurants by searching for keyword "Chinese" in the business categories column.
- Under the review dataset, select the relevant reviews based on the list of business IDs

In total, we found that there are 4,468 restaurants specialized in Chinese cuisine with 227,132 customer reviews.

## 2.2 Data Exploration

### 2.2.1 Initial Exploration of Sentiment Distribution

There was no target column of the sentiment in the customer review data. However, there was a "stars" column which was the rating of the business given by the customers. We would need to find a best way to create the target "sentiment" column in later process.

In the initial data exploration process, we used the following logic to create a sentiment column based on the user review ratings:

- Positive: Rating > 3
- Neutral: Rating = 3
- Negative: Rating < 3

We computed and visualized the total number of positive, neutral and negative reviews based on above sentiment label creation process:

Total negative reviews: 56841
Total positive reviews: 134161
Total neutral reviews: 36130

Ratio among the positive, negative, neutral reviews was about 50:26:24.

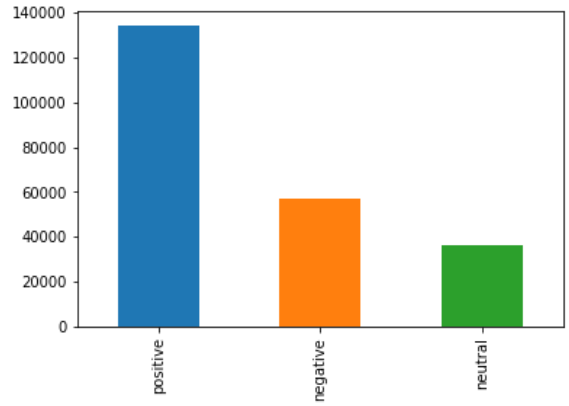


**Figure 1. Sentiment Distribution**

### 2.3.2 Top 20 Restaurants Receiving Most Positive Reviews

We then explored the top 20 restaurants which received the most positive reviews. The restaurant receiving the most positive reviews is located in Las Vegas.

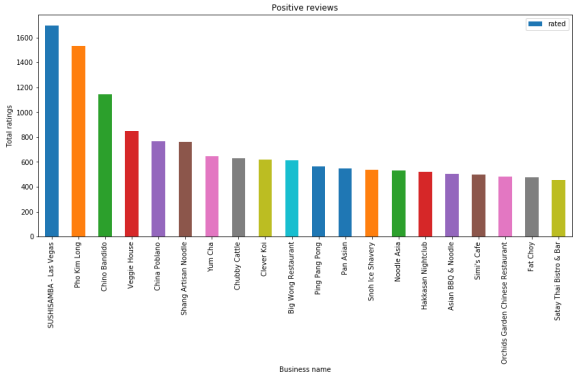


**Figure 2. Top 20 Restaurants (Positive Reviews)**

### 2.2.3 Top 20 Restaurants Receiving Most Negative Reviews

Similarly, we explored the top 20 restaurants who receive the most negative reviews. One of the interesting observations is that several top 20 most

positively reviewed restaurants are also on the list of the top 20 most negatively reviewed restaurants. This may indicate the diverse taste and preference among the customers in the Chinese cuisine.

For example, Pho Kim Long, located Las Vegas, has made to both the Top 20 positive and negative reviews. The high volume of reviews can be attributed to the fact that the restaurant is open 24 hours a day, 7 days a week. This restaurant is frequented by tourists as we looked up their Yelp site: https://www.yelp.ca/biz/pho-kim-long-las-vegas

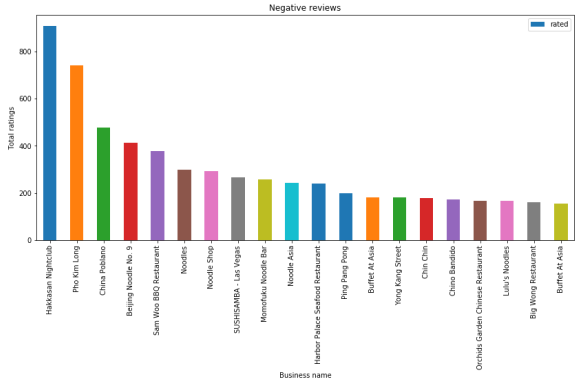


**Figure 3. Top 20 Restaurants (Negative Reviews)**

## 2.2.4 Restaurant Locations

Top three locations that received the highest number of reviews in Chinese restaurants are Las Vegas, Toronto and Phoenix.

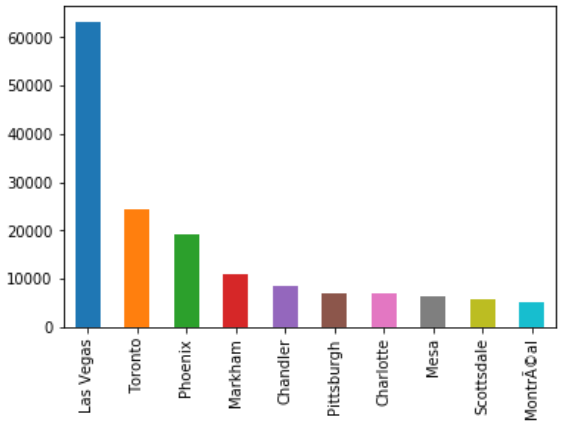


**Figure 4. Top Restaurant Locations**

## 2.2.5 Word Cloud of Customer Reviews

We counted the word frequency the top most positively and negatively reviewed restaurants respectively and visualized the word frequency in word cloud visualizations.

### 2.2.5.1. Positive Reviews to SUSHISAMBA - Las Vegas

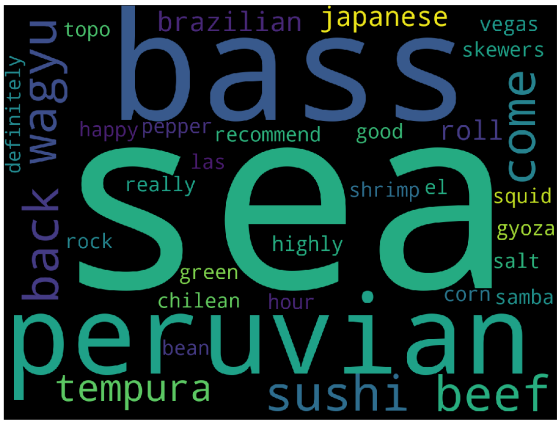


**Figure 5. Word Cloud for SUSHISAMBA - Las Vegas (Positive)**

----- 10 most common 2-word phrase -----
sea bass: 435
sushi samba: 372
peruvian corn: 155
las vegas: 136
happy hour: 126
highly recommend: 117
sushi rolls: 113
wagyu beef: 113
come back: 112
really good: 106

----- 10 most common 3-word phrase -----
sea bass skewers: 77
chilean sea bass: 75
rock shrimp tempura: 46
salt pepper squid: 41
wagyu beef gyoza: 39
japanese brazilian peruvian: 39
el topo roll: 36
green bean tempura: 35
japanese peruvian brazilian: 32
definitely come back: 31

**2.2.5.2. Negative Reviews to Pho Kim Long**

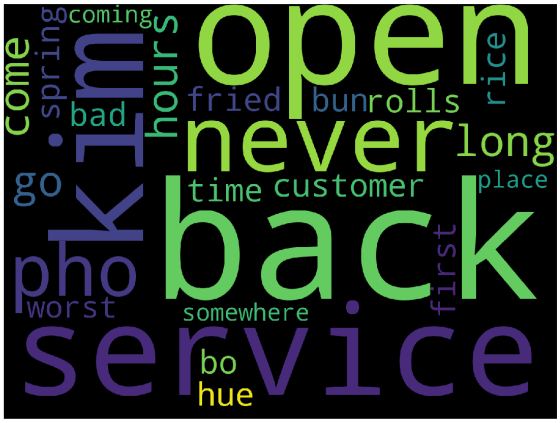


**Figure 6. Word Cloud for Pho Kim Long (Negative)**

----- 10 most common 2-word phrase -----
kim long: 127
pho kim: 123
open 24: 76
late night: 70
24 hours: 66
spring rolls: 63
egg rolls: 61
pho place: 44
best pho: 42
24 7: 36

----- 10 most common 3-word phrase -----
pho kim long: 116
open 24 hours: 47
open 24 7: 22
bun bo hue: 16
place open 24: 12
fried egg rolls: 10
bo luc lac: 9
kim long pho: 8
best pho town: 7
late night pho: 7

While we are not surprised to see that the most frequently used words in both cases are related to food, we observed several most used adjectives that would indicate the sentiment level. For example, under the positively reviewed word cloud, we can see words like "best", while we can see "never" and "worst" in the negatively reviewed word cloud.

## 2.4 Data Processing and Cleaning

### 2.4.1 Sentiment Label Creation

As there was no manually coded "target" column of sentiment, we needed to generate the target column from either the "stars"column from the original dataset or by creating a "stars_norm" column (normalization based on "stars" column per user).

Due to a lack of standard rule of transforming rating to sentiment target, we explored seven approaches to create the sentiment column. The dataset was then trained using Bag of Words (BOW) as the feature engineering and Logistic Regression and SVM as the prediction models. The results are summarized in Table 1.

We decided to choose the seventh method (stars > 3, positive; stars < 3, negative) listed in Table 1 for creating the target column because this method yields the best result.

We considered the treatment for the 3-star reviews. The neutral rating could cause issues with the sentiment predictions because the algorithms might not be able to determine whether certain reviews are neither positive nor negative. In addition, from the business' perspective, when a user decided to write a review, he/she would mostly express his/her positive or negative sentiment only so we reckon that there is no such thing as "neutral sentiment".

Therefore, we decided to remove the 3-star reviews in model training in order to build models that can predict positive and negative sentiment more accurately. Rather, we would like to re-classify these 3-star reviews by feeding the reviews into one of the trained models to predict the "true" sentiments based on the reviews' texts, semantics and contexts.

**Table 1. Different Target Generation Methods - Impact on Model Performance**

| Index | Rule of Creating Target Column | Accuracy (LR on BOW) | Accuracy (SVM on BOW) |
|---|---|---|---|
| 1 | stars_norm > 0, positive; star_norm = 0, neutral; stars_norm <0 negative | 0.6017 | 0.6068 |
| 2 | stars_norm >= 0, positive; stars_norm <0, negative | 0.791 | 0.796 |
| 3 | stars_norm > 0, positive; stars_norm <=0, negative | 0.7654 | 0.7714 |
| 4 | stars > 3, positive; star = 3, neutral; star < 3, negative | 0.8106 | 0.8139 |
| 5 | stars > =3, positive; star < 3, negative | 0.9077 | 0.9115 |
| 6 | stars > 3, positive; star <= 3,negative | 0.8722 | 0.8759 |
| 7 | stars > 3, positive; stars < 3, negative | 0.944 | 0.9465 |

### 2.4.2 Data Processing and Cleaning

We implemented a language detection routine to eliminate any reviews that are not in English. We found that 99.5% of the reviews are in English. The remaining 0.5% of the reviews were removed.

### 2.4.3 Texts Cleaning

We deployed the following texts cleaning steps:

- Convert all the uppercase characters to lowercase
- Remove punctuations
- Remove numbers
- Remove stopwords
- Lemmatization

## 2.5 Feature Engineering, Model Selection and Evaluation

### 2.5.1 Feature Engineering

In this research project, we attempted four different feature engineering methods: BOW, TF-IDF and N-Grams and Word2Vec.

- Bag-of-Words (BOW) - The BOW feature extraction model simply counts the word frequency and occurrence from the corpus without concerning about the grammar, word orders and sequences.
- N-grams - N-grams is the extension of the BOW model which takes in a collection of word tokens from the document. N denotes the number of words in a collection (e.g. uni-gram is 1 word which is BOW, bi-gram is 2 words, tri-gram is 3 words and so on)
- Term Frequency–Inverse Document Frequency (TF-IDF) - It is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling.
- Word2Vec - Word2Vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space [4].

When comparing the performance of ensemble learning models versus the base models, we chose TF-IDF as the feature engineering as the baseline, as this method is particularly useful for sentiment analysis. TF-IDF measures document relevance instead of word frequency. It favors words that are distinct but appear frequently in the document.

### 2.5.2 Model Selection

The candidate models that are suitable for sentiment analysis are classification models such as Logistic Regression (LR) and Support Vector Machine (SVM). More advanced modeling technique such as ensemble learning and neural network would also be used to enhance the prediction performance.

### 2.5.3 Model Evaluation

In order to validate the model accuracy, we also implemented the K-fold cross validation.

### 2.5.4 Project Design

We split the research project into two phases:

**Phase 1** was to implement a combination of FE techniques BOW, TF-IDF and N-Grams and Logistic Regression and SVM algorithms. We then selected the FE and model combination that gives the best prediction performance by analyzing the evaluation metrics.

**Phase 2** was to implement ensemble learning to enhance the model performance and reduce variance. We also implemented a supervised deep learning library - Keras on a relatively more sophisticated FE, namely Word2Vec.

Table 2 and 3 summarize the FE and the algorithms that we implemented in Phase 1 and 2 respectively.

**Table 2. Phase 1: Feature Engineering and Model Trials**

| Trial Index | FE Method | Model |
|---|---|---|
| 1 | BOW | LR |
| 2 | BOW | SVM |
| 3 | TF-IDF | LR |
| 4 | TF-IDF | SVM |
| 5 | N-Grams | LR |
| 6 | N-Grams | SVM |

**Table 3. Phase 2: Feature Engineering, Ensemble Learning and Deep Learning**

**Bagging**

| Trial Index | FE Method | Model |
|---|---|---|
| 7 | TF-IDF | LR |
| 8 | TF-IDF | SVM |

**Random Forest**

| Trial Index | FE Method | Model |
|---|---|---|
| 9 | TF-IDF | Random Forest |

**Boosting**

| Trial Index | FE Method | Model |
|---|---|---|
| 10 | TF-IDF | SVM + Adaboost |

**Stacking**

| Trial Index | FE Method | Model |
|---|---|---|
| 11 | TF-IDF | SVM + Random Forest + KNN (NB as meta-classifier) |
| 12 | TF-IDF | SVM + RandomForest + KNN (LR as meta-classifier) |
| 13,14 | TF-IDF | DecisonTree + Logistic Regression (KNN as meta-classifier) |

**Neutral Network**

| Trial Index | FE Method | Model |
|---|---|---|
| 15 | Word2Vec | DNN |

# 3. Results

## 3.1 Phase 1 - Prediction Using 3 FEs and 2 Models

The prediction results of Phase 1 are recorded in the table below. The prediction results are expressed in Recall and Precision metrics.

**Table 4. Phase 1 Results**

| Trial Index | FE + Model | Precision (Positive Negative) | Recall (Positive Negative) |
|---|---|---|---|
| 1 | BOW + LR | 0.943 (0.95 0.92) | 0.944 (0.97 0.89) |
| 2 | BOW + SVM | 0.9462 (0.96 0.93) | 0.9465 (0.97 0.89) |
| 3 | TF-IDF + LR | 0.949 (0.95 0.95) | 0.9491 (0.98 0.88) |
| 4 | TF-IDF + SVM | 0.9413 (0.94 0.95) | 0.9407 (0.98 0.84) |
| 5 | N-Grams + LR | 0.9552 (0.96 0.94) | 0.9554 (0.97 0.91) |
| 6 | N-Grams + SVM | 0.953 (0.96 0.93) | 0.9532 (0.97 0.91) |

We employed K-fold validation to evaluate the models. For simplicity, we selected Trial 3 (TF-IDF + LR) to perform the K-fold validation. We chose K to be 5. Below table summarizes the results.

**Table 5. K-fold Cross Validation Results (TF-IDF + Logistic Regression)**

| Fold # | Precision | Recall |
|---|---|---|
| 1 | 0.9479 | 0.9479 |
| 2 | 0.9508 | 0.951 |
| 3 | 0.9519 | 0.952 |
| 4 | 0.9492 | 0.9494 |
| 5 | 0.9505 | 0.9506 |
| Mean | 0.95006 | 0.95018 |
| SD | 0.00154 | 0.00158 |

The results above show that the model is adequately generalized to handle new test data as the results are in line with the results shown in Trial 3.

## 3.2. Phase 2 - Ensemble and Supervised Deep Learning

### 3.2.1 Bagging

Below table lists the results of LR and SVM model and these two base models plus Bagging. The Standard Deviation slightly reduced for SVM when applying Bagging.

**Table 6. Bagging**

| Trial Index | FE Method + Model | Bagging | Accuracy (Standard Deviation) |
|---|---|---|---|
| 7 | TF-IDF +LR | N | 0.949583 (+/- 0.000899) |
| 7 | TF-IDF | Y | 0.944072 (+/- |

| | | | |
|---|---|---|---|
| | +LR | | 0.001152) |
| 8 | TF-IDF + SVM | N | 0.937064 (+/- 0.001371) |
| 8 | TF-IDF + SVM | Y | 0.925212 (+/- 0.001333) |

### 3.2.2 Random Forest

Comparing the result of Random Forest (RF) with Decision Tree (DT), we can find RF model outperformed the DT model, however, it underperforms all of the six non-tree based models (Trial 1 - 6).

**Table 7. Random Forest**

| Trial Index | FE Method + Model | Precision (Positive Negative) | Recall (Positive Negative) |
|---|---|---|---|
| 9 | TF-IDF + Decision Tree | 0.8559 (0.89 0.76) | 0.8567 (0.90 0.75)) |
| 9 | TF-IDF + Random Forest | 0.8761 (0.88 0.88) | 0.8758 (0.96 0.68) |

### 3.2.3 Boosting

We ran the Adaboost on the TF-IDF + SVM base model. We found that when evaluated by precision, predictions of the negative sentiments improved at the expense of the positive sentiments.

**Table 8. Boosting**

| Trial Index | FE Method + Model | Precision (Positive Negative) | Recall (Positive Negative) |
|---|---|---|---|
| 4 (Base line) | TF-IDF + SVM | 0.9413 (0.94 0.95) | 0.9407 (0.98 0.84) |
| 10 | TF-IDF + SVM (Adaboost) | 0.9328 (0.95 0.89) | 0.9329 (0.95 0.88) |

### 3.2.4 Stacking

In Trial 11 and Trial 12, we used advanced and diversified base models and simple meta models, as suggested by Kaggle top winner Marios Michailidis [3]. However, the results turned out to be not as satisfactory as anticipated. The models were particularly not able to predict negative posts correctly (Table 8).

In Trial 13, we used two simpler base models and KNN as meta model, the prediction of negative posts increased, and the overall result marginally outperformed base models SVM and Logistic Regression (Trial 4 and Trial 3). In this trial we used max_depth =1 as the parameter of Decision Tree.

In Trial 14, we used the same combination as Trial 13 except for increasing the depth of the Decision Tree. The performance was worse than that of Trial 13.

**Table 9. Stacking (All Using TF-IDF as Feature Engineering)**

| Trial Index | FE Method + Model | Precision (Positive Negative) | Recall (Positive Negative) |
|---|---|---|---|
| 4 (Baseline) | TF-IDF + SVM | 0.9413 (0.94 0.95) | 0.9407 (0.98 0.84) |
| 11 | SVM + Random Forest + KNN (NB as meta-classifier) | 0.7971 (0.82 0.75) | 0.8036 (0.93 0.51) |
| 12 | SVM + RandomForest + KNN | 0.7971 (0.82 0.75) | 0.8036 (0.93 0.51) |

| | | | |
|---|---|---|---|
| | (LR as meta-classifier) | | |
| 13 | DecisonTree (depth=1) + Logistic Regression (KNN as meta-classifier) | 0.9495 (0.95 0.95) | 0.9496 (0.98 0.88) |
| 14 | DecisonTree + Logistic Regression (KNN as meta-classifier) | 0.8528 (0.89 0.76) | 0.8537 (0.90 0.74) |

### 3.2.5 Deep Learning

The prediction results of Word2Vec + DNN are inferior to the performance of the base models, especially for predicting negative posts.

**Table 11. Deep Learning**

| Trial Index | FE Method + Model | Precision (Positive Negative) | Recall (Positive Negative) |
|---|---|---|---|
| 15 | Word2Vec + DNN | 0.7357 (0.78 0.62) | 0.7512 (0.89 0.42) |

## 3.3. Model Implementation - Determining "True" Sentiment of 3-Star Rated Reviews

We fed the 3-star reviews to the TF-IDF + LR model, Table 11 shows the predictions of the "true" sentiment. The predicted proportion of positive vs negative of the 3-star reviews is almost identical to the proportion of positive vs negative in original data (Figure 1).

**Table 11. "True" Sentiment for Neutral Reviews**

| Sentiment | Count | Percent |
|---|---|---|
| Positive | 24,737 | 69% |
| Negative | 11,209 | 31% |

# 4. Discussions

**Base Models**. All six base models have superior predictive performance with the precision and recall metrics being around 94% - 95%.

**Bagging.** Applying Bagging on SVM slightly lowered variance comparing to single mode. As a special type of Bagging, Random Forest produces better performance than Decision Tree (Table 7).

**Boosting.** The overall precision and recall metrics did not improve after applying AdaBoost to SVM. We observed that the prediction capability on the negative reviews improved slightly at the expense of the prediction power for the positive reviews when evaluating using recall (Table 8).

**Stacking.** We found that all 4 stacking attempts did not further improve the base models, where three of the four stacking models produce worse prediction metrics, while one model only matches the performance of the base models. We observed that stacking advanced models would not improve the model performance.

**Deep Learning.** The prediction results of neural network techniques was not satisfactory. Future works may consider to stack neural networks with other models.

**Sentiment Prediction on 3-Star Reviews.** The prediction results on 3-star reviews proved our observation and assumption in sentiment label generation process: 3-star reviews are a mixture of positive and negative posts. Therefore removing 3-star reviews in the initial model building process is a validated decision. In the future work, we may consider adding the 3-star reviews with predicted sentiment labels back to build another model.

## 5. Conclusion and Future Work

**Conclusion**. In this project, we used an unstructured dataset which consists of over 227,000 Yelp reviews on Chinese restaurants. In the first part of the project, we performed data exploration, data cleaning and language detection; implemented 4 feature engineering routines (BOW, TF-IDF and Bags-of-N-Grams, Word2Vec) and 2 algorithms (Logistic Regression and SVM). We firstly removed the 3-star reviews as part of the data cleaning exercise and feed these neutral reviews into one of the models to determine the true sentiment (positive or negative). In all cases, we achieved satisfactory prediction results with high precision and recall scores in the range of 94%-95%. All 6 base models (3 FEs + 2 models) generate similar results.

We followed up with the implementation of the cross validation routines to evaluate the models and we found that the models were generalized enough to describe dynamics.

In the second part of the project, we implemented ensemble routines: Bagging, Boosting and Stacking. Our Boosting and Stacking trials were not able to make an obvious improvement to model performance. That may be due to the limited number of trials and our lack of experience implementing ensemble routines.

We also attempted word embeddings (Word2Vec) feature engineering and a neutral network model. Similar to the results of ensemble models, the results were inferior than the base models.

**Future Work.** Our research provides practical insights for data scientists in enterprises or organizations who want to apply machine learning techniques on customer reviews to predict customer sentiment.

However, it is important to note that our research has several limitations and future work could enhance our research. First, our research is conducted based on data collected from one platform - Yelp, which is most popular in North America. We also just selected reviews on Chinese restaurants in the English language only. Therefore, it is recommended that future work could enlarge the data variety.

Secondly, we only tried 4 stacking combinations due to limited computational resources, and they were all one layer ensemble learning. Future work could expand the investigation and try multi-layer stacking.

Thirdly, we only tried 4 feature engineering methods, and when we compared the performance of ensemble models V.S base models, we used TF-IDF only. Future works could try other feature engineering methods.

## 6. References


[1] Yelp Inc, “Yelp”, retrieved from https://www.yelp.com/, January 2019.

[2] Stack Exchange, “Bagging, boosting and stacking in machine learning”, retrieved from https://stats.stackexchange.com/questions/18891/bagging-boosting-and-stacking-in-machine-learning, January 2019.

[3] Michailidis, M., “How to Win a Data Science Competition: Learn from Top Kagglers - Ensembling - Stacking”, retrieved from https://www.coursera.org/lecture/competitive-data-science/stacking-Qdtt6, January 2019.

[4] Mikolov, Tomas; et al. "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781