Text & Web Analytics Syndicate Project Application Report

Syndicate 1 – Mathew, Shuting, Yijie, Nick, Bill

Introduction

With the rise of food discovery platforms like Yelp and Zomato, there is a large amount of information being shared customer sentiment and perspectives on the various restaurants all over the world. The food service industry is a competitive landscape with low barriers to entry. Restaurants need to stay at the top condition in all aspects – including food quality, service, and ambience. Otherwise, word of mouth will spread, and can swiftly damage a restaurant's reputation.

Methods

The Task

There is an opportunity to turn the plethora of restaurant text reviews into actionable insights in a scalable manner, using this application. This application generates insights regarding the aspect-based performance for individual restaurants. The motivating idea is to be able to identify customer pain points, and improvements for restaurant managers, customers, and other stakeholders. Food discovery platforms who possess review data can share insights, and interesting sentiments to restaurants on their platform – to monetise or share as goodwill.

Dataset and Experimental Design

A dataset containing reviews for a certain restaurant with pre-labelled classes for polarity and the aspect was used. A total of 1406 labelled rows were used for training, and a holdout dataset with 282 unlabelled rows was used as the basis for the insights outlined further. The value of this application is to be able to predict the labels and insights for the aspect-less reviews on sites like Yelp, without the costly manual labelling associated with doing this on a large scale.

The methodology for this application involves a two-stage approach. Two distinct models are used to classify the category and the sentiment polarity, respectively – allowing for more customisability and performance tuning for each model. Detailed model parameters and accuracy scores are outlined in the optional appendix.

An 80%/20% train-test split was used to ensure sufficient training data while not overfitting. 6 model types were chosen and was evaluated against the common target throughout iterations of hyperparameter and features experiment, each iteration varies the target and compares the result with previous performance holding other model configurations constant.

Methodology – Stage 1: Text Classification

This stage extracts the aspects "food", "service", "ambiance" and "price" from the dataset, and the approach involves the process of features experiment, model selection & evaluation.

Tools Used, Feature Cleaning & Transformation:

As the feature matrix for NLP tasks is sparse, feature engineering is necessary to help the models to focus on more essential input features and improve performance. Restaurant reviews are cleaned using regex commands. Letters were lowercased for consistency, and non-alphabetic letters or words from NLTK's list of English stop words are removed (likely noisy features). While a term-document matrix may suffice, a TF-IDF transformation is better for this context as it deprioritises

the importance of some general words that occur commonly in all documents. Here, text is represented as a feature matrix for the models.

As an example, the TF-IDF score for the term "Tiramisu" that appears 4 times under the food document and also appears in another document is calculated as (using the sklearn TfidfVectorizer):

TF (w) = No. of times term 'w' occurs in a document = 4
IDF(w) = ln[N/DF(W)] + 1 =ln(4/2)+1
TF-IDF(w)=4*(ln(4/2)+1)=6.78

A word that is found in a smaller amount of topics would receive a higher weight, as it provides strong signals of topic it belongs to (e.g. tiramisu signals food). A parameter space of unigrams, bigrams, trigrams, and 4-5 grams, maximum allowable features (1000,1500,2000), and minimum document frequency (1 or 2) have been compared – with unigrams and bigrams and 1500 features attaining best performance (f1-score).

Additional features:

A TF-IDF transformation is also used here to create a feature matrix. Topic-specific features were created to enhance performance. Dictionaries corresponding to food, service, ambiance, and price were also created. For example, the word *costly* can be a very strong signal to the aspect of price, and the word *soup* has a strong relation to food. The additional four feature dimensions capture the signals of these words and put 1 (different values were explored, but a value of 1 provided the greatest resulting accuracy. The prediction of the food and service category would be better, roughly 5% gain in f1-score). Modifying the quality of the words of dictionaries based on our domain understanding in the context of restaurants also greatly improved the model.

Model Creation and Hyperparameter Tuning:

The following machine learning models are trained – the MultinomialNaïveBayes (MNB), LinearSVC (LSVC), LogisticRegression (LR), DecisionTree Classifier, KNeighborsClassifier, and the XGBClassifier. Relevant hyperparameters were identified for each model type, with a range of possible values specified as a search space. To tune the hyperparameters, we used a mix of Cross-validated grid search, Cross-validated halving grid search, and Cross-validated halving random search – repeatedly narrowing down target values from the initial search space to find an optimum. In this stage, when evaluated for f1-score, the MNB and LSVC perform better than other models, though different models have different strength in predicting different categories.

Evaluation and discussion:

We focused on maximizing the weighted f1-score. This is due to a target class imbalance (food and service have more observations, but also being the most important) in the dataset – and in this application, both precision and recall are important. The f1-score (representing the harmonic mean of both metrics) was used to provide a well-balanced objective. Weighted f1-score ($Weighted\ \mathrm{F1\_score} = \frac{1}{\mathrm{N}}\sum_{i=0}^{N} F1_{score_i} * w_i$) which places higher weights on these two important categories.

Precision, recall and f-1 score is summarised below, and calculated by the formulas:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\mathrm{F}_1 = 2 * \frac{precision * recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Linear SVC	precision	recall	f1-score	support
ambience	0.731707	0.652174	0.689655	46
food	0.768657	0.927928	0.840816	111
price	0.815789	0.688889	0.746988	45
service	0.898551	0.775	0.832215	80
accuracy	0.801418	0.801418	0.801418	0.801418
macro avg	0.803676	0.760998	0.777419	282
weighted avg	0.807	0.801418	0.798746	282

The LSVC model was best in overall performance. A visualization of prediction/actual text categories is provided in appendix plot 1. The recall score for predicting food is particularly high, meaning that our model is sensitive to message of food, which may also be attributed to the signal-rich features from food dictionaries available for prediction. On the other hand, the MNB model has strong precision in predicting service category and rarely makes mistakes when the model detects the service category. In contrast, LSVC model has very similar performance compared to MNB model, though MNB model outperforms in predicting for ambiance. The other models perform relatively poorly compared to these two. Price and ambience are harder to predict, partly because features and concepts for these two aspects are more subtle and occur less frequently (the small support value indicates possible noise in the subset).

Methodology – Stage 2: Sentiment Polarity Classification

I*n the second stage*, we conduct a sentiment analysis from four polarity scores: "negative", "conflicted", "neutral", and "positive". A similar modelling process was used, only with key differences in feature engineering and output class labels.

Tools Used, Feature Cleaning & Transformation:

The cleaning process is similar to stage one. However, punctuation marks may play an important role in sentiment analysis ("!" signals an emotion that leans towards positive or negative rather than neutral). The transformed restaurant reviews also used same TF-IDF approach except that the four documents are now under their respective polarity scores. The parameter set for transformation remains similar, except the maximum allowable features is smaller (1000 compared to 1500) as features like keywords of the polarity are fewer for sentiment polarity than text classification tasks.

Additional features:

Lists of keywords (positive & negative) were maintained and counts of positive and negative words of each text review were added as additional feature dimensions. The count of punctuation marks, and review length are also included as features (may indicate a changed level of sentiment and reduced neutrality).

Modelling, Evaluation and Discussion:

The same set of models have been considered, but the same types of models work best as well. Here, we focused on maximizing both a weighted average, and a macro average f1-score. The weighted average is used as many restaurants would care more on the positive and negative polarity

categories. The macro average score ($\text{Macro F1\_score} = \frac{1}{N}\sum_{i=0}^{N} F1\_score_i$) is also used to maintain balance of performance for conflict or neutral classes as these are smaller categories.

MNB polarity	precision	recall	f1-score	support
conflict	0.25	0.1	0.142857	20
negative	0.589041	0.651515	0.618705	66
neutral	0.4	0.125	0.190476	16
positive	0.790816	0.861111	0.824468	180
accuracy	0.716312	0.716312	0.716312	0.716312
macro avg	0.507464	0.434407	0.444127	282
weighted avg	0.683063	0.716312	0.691998	282

Although LSVC is better at predicting conflict sentiment and the logistic regression has better performance in predicting neutral message, the MNB was chosen for this stage for its performance (highest weighted average score) in predicting positive and negative sentiments. Outlook of MNB model prediction is illustrated in appendix plot 2.

Insights, Conclusions and Further improvement:

		Sentiment Polarity					
		Positive	Negative	Neutral	Conflict	**Total**	% Positive
Aspect	Food	101	24	3	6	134	75%
	Service	42	25	2	0	69	61%
	Ambience	27	12	0	2	41	68%
	Price	26	12	0	0	38	66%
	Total	**196**	**73**	**5**	**8**	**282**	

Looking at predictions made for the holdout dataset, the restaurant has mediocre performance for each aspect. Doing decently on food quality, but less well on service, ambience and price. Customer sentiment on the service and price is divisive, indicating the manager should focus improvements on service and value for money. This example interpretation can be automated and repeated with data from various restaurants, food discovery platforms can process and aggregate their review data with richer information to share with customers and restaurants alike.

Due to time constraints, the Random Forest and XGB models have not been optimized well enough to be serious contenders. Nevertheless, the MNB, LSVC and LR models have achieved satisfactory outcomes for both tasks, and we outline further improve their performance. The dictionary for each of the target classes in both the aspect and sentiment prediction models can be further tailored and enhanced to enable more precise feature engineering. A future extension of this application could incorporate keyword extraction to understand what specific points restaurants can improve upon. This will help managers make even more targeted decisions and improvements. An attempt was made to implement this, but the dataset was not rich enough for such an analysis.

Additionally, incorporating word embedding information, and part-of-speech tagging (locate negating words and record as additional features) in classification and sentiment analysis respectively.

Appendix

Model 1: Text Classification

Hyperparameters Used

Table 1: Multinomial Naive Bayes Hyperparameters and Values (for Methodology 1: Text Classification)

name	**value**
memory	None
steps	('clf', MultinomialNB(alpha=0.1))
verbose	False
clf	MultinomialNB(alpha=0.1)
clf__alpha	0.1
clf__class_prior	None
clf__fit_prior	True

Table 2: SVC Model Hyperparameters and Values (for Methodology 1: Text Classification)

name	**value**
memory	None
steps	('clf', LinearSVC(C=0.64, class_weight='balanced', loss='hinge', max_iter=10000, random_state=99))
verbose	False
clf	LinearSVC(C=0.64, class_weight='balanced', loss='hinge', max_iter=10000, random_state=99)
clf__C	0.64
clf__class_weight	balanced
clf__dual	True
clf__fit_intercept	True
clf__intercept_scaling	1
clf__loss	hinge
clf__max_iter	10000
clf__multi_class	ovr
clf__penalty	l2
clf__random_state	99
clf__tol	0.0001
clf__verbose	0

Additional Breakdown on Performance Information

Table 3: Performance of Naïve Bayes Model (for Methodology 1: Text Classification)

Naïve Bayes	**precision**	**recall**	**f1-score**	**support**
ambience	0.790698	0.73913	0.764045	46
food	0.77037	0.936937	0.845528	111
price	0.810811	0.666667	0.731707	45
service	0.880597	0.7375	0.802721	80
accuracy	0.804965	0.804965	0.804965	0.804965
macro avg	0.813119	0.770059	0.786	282
weighted avg	0.811409	0.804965	0.80193	282

Table 4: Performance of Logistic Regression Model (for Methodology 1: Text Classification)

Logistic Regression	**precision**	**recall**	**f1-score**	**support**
ambience	0.68	0.73913	0.708333	46
food	0.793651	0.900901	0.843882	111
price	0.780488	0.711111	0.744186	45
service	0.892308	0.725	0.8	80
accuracy	0.794326	0.794326	0.794326	0.794326
macro avg	0.786612	0.769036	0.7741	282
weighted avg	0.800999	0.794326	0.793413	282

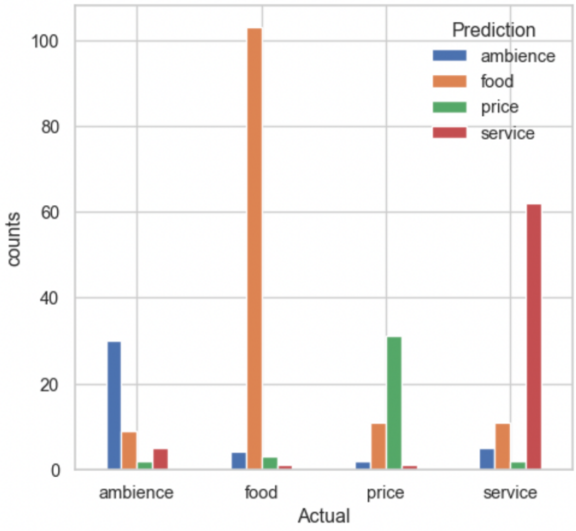
Table 5: Performance of Random Forest Model (for Methodology 1: Text Classification)

Random Forest	**precision**	**recall**	**f1-score**	**support**
ambience	0.647059	0.478261	0.55	46
food	0.690476	0.783784	0.734177	111
price	0.736842	0.622222	0.674699	45
service	0.738095	0.775	0.756098	80
accuracy	0.705674	0.705674	0.705674	0.705674
macro avg	0.703118	0.664817	0.678743	282
weighted avg	0.704302	0.705674	0.700861	282

Table 6: Performance of XGBoost Model (for Methodology 1: Text Classification)

XGBoost	**precision**	**recall**	**f1-score**	**support**
ambience	0.755556	0.73913	0.747253	46
food	0.714286	0.900901	0.796813	111
price	0.818182	0.6	0.692308	45
service	0.859375	0.6875	0.763889	80
accuracy	0.765957	0.765957	0.765957	0.765957
macro avg	0.78685	0.731883	0.750066	282
weighted avg	0.778757	0.765957	0.762712	282

Plot 1: Outlook of MNB model in classifying text categories.



Model 2: Text Sentiment Analysis

Hyperparameters Used

Table 7: Multinomial Naive Bayes Hyperparameters and Values (for Methodology 2: Sentiment Polarity Classification)

name	**value**
cv	5
error_score	nan
estimator__memory	None
estimator__steps	('clf', MultinomialNB())
estimator__verbose	False
estimator__clf	MultinomialNB()
estimator__clf__alpha	1.0
estimator__clf__class_prior	None
estimator__clf__fit_prior	True
estimator	Pipeline(steps=[('clf', MultinomialNB())])
n_jobs	-1
param_grid	nan
pre_dispatch	2*n_jobs
refit	True
return_train_score	False
scoring	f1_weighted
verbose	3

Table 8: SVC Hyperparameters and Values (for Methodology 2: Sentiment Polarity Classification)

name	value
memory	None
steps	('clf', LinearSVC(C=0.16, class_weight='balanced', max_iter=10000, random_state=99))
verbose	FALSE
clf	LinearSVC(C=0.16, class_weight='balanced', max_iter=10000, random_state=99)
clf__C	0.16
clf__class_weight	balanced
clf__dual	TRUE
clf__fit_intercept	TRUE
clf__intercept_scaling	1
clf__loss	squared_hinge
clf__max_iter	10000
clf__multi_class	ovr
clf__penalty	l2
clf__random_state	99
clf__tol	0.0001
clf__verbose	0

Additional Breakdown on Performance Information

Table 9: Performance of Linear SVC Model (for Methodology 2: Sentiment Polarity Classification)

SVC	**precision**	**recall**	**f1-score**	**support**
conflict	0.333333	0.2	0.25	20
negative	0.512195	0.636364	0.567568	66
neutral	0.375	0.375	0.375	16
positive	0.80814	0.772222	0.789773	180
accuracy	0.677305	0.677305	0.677305	0.677305
macro avg	0.507167	0.495896	0.495585	282
weighted avg	0.680626	0.677305	0.675952	282

Table 10: Performance of Logistic Regression Model (for Methodology 2: Sentiment Polarity Classification)

Logistic Regression	**precision**	**recall**	**f1-score**	**support**
conflict	0.185185	0.25	0.212766	20
negative	0.450549	0.621212	0.522293	66
neutral	0.3	0.5625	0.391304	16
positive	0.858209	0.638889	0.732484	180
accuracy	0.602837	0.602837	0.602837	0.602837

macro avg	0.448486	0.51815	0.464712	282
weighted avg	0.683396	0.602837	0.627073	282

Table 11: Performance of Random Forest Model (for Methodology 2: Sentiment Polarity Classification)

Random Forest	**precision**	**recall**	**f1-score**	**support**
conflict	0.25	0.05	0.083333	20
negative	0.453333	0.515152	0.48227	66
neutral	0.666667	0.125	0.210526	16
positive	0.75	0.833333	0.789474	180
accuracy	0.663121	0.663121	0.663121	0.663121
macro avg	0.53	0.380871	0.391401	282
weighted avg	0.640378	0.663121	0.634646	282

Table 12: Performance of XGBoost Model (for Methodology 2: Sentiment Polarity Classification)

XGBoost	**precision**	**recall**	**f1-score**	**support**
conflict	0.285714	0.1	0.148148	20
negative	0.493671	0.590909	0.537931	66
neutral	0.4	0.25	0.307692	16
positive	0.77957	0.805556	0.79235	180
accuracy	0.673759	0.673759	0.673759	0.673759
macro avg	0.489739	0.436616	0.44653	282
weighted avg	0.656096	0.673759	0.659619	282

Plot 2: Outlook of MNB model in analysing text sentiments