Capstone Report

1. Background

Shanghai is one of the four municipalities of the People's Republic of China. It is located on the southern estuary of the Yangtze, with the Huangpu River flowing through it. With a population of 24.28 million as of 2019, it is the most populous urban area in China and the second most populous city proper in the world. Greater Shanghai is a global center for finance, technology, innovation and transportation and the Port of Shanghai is the world's busiest container port.

Shanghai has been described as the "showpiece" of the booming economy of China. Featuring several architecture styles such as Art Deco and shikumen, the city is renowned for its Lujiazui skyline, museums and historic buildings—including the City God Temple, Yu Garden, the China Pavilion and buildings along the Bund. Shanghai is also known for its sugary cuisine, distinctive dialect and vibrant international flair. Every year, the city hosts numerous national and international events, including Shanghai Fashion Week, the Chinese Grand Prix and ChinaJoy.

2. Description of the problem

In this scenario, it is urgent to adopt machine learning tools in order to assist homebuyer’s clientele in Shanghai to make wise and effective decisions. As a result, the business problem we are currently posing is: how could we provide support to homebuyer’s clientele in to purchase a suitable real estate in Shanghai in this uncertain economic and financial scenario?

To solve this business problem, we are going to cluster Shanghai neighborhoods in order to recommend venues and the current average price of real estate where homebuyers can make a real estate investment. We will recommend profitable venues according to amenities and essential facilities surrounding such venues i.e. elementary schools, high schools, hospitals & grocery stores.

3. Data Representation

The computer can’t recognise natural language, and can’t deal with unstructured data such as text directly, so we need to pre-process the textual data at the beginning. Better data pre-processing strategies can produce more useful information and provide more feature engineering options.

3.1 CountVectorizer & Doc2Vec

Bag of Words’ and ‘n-grams’ are two commonly used methods to turn these text content into numerical feature vectors. CountVectorizer allows us to use the BOW approach (Susan, 2017), which just finds words that are in their vocabulary and then assigns them some score. In contrast, Doc2Vec, using the method of n-grams, can capture the relationship between words.

I trained the dataset with base classifier NB (MNB for CountVectorizer while GNB for Doc2Vec) and Logistic Regression to check the performance.

Method	Model	Accuracy
CountVectorizer	MNB	83.90%
Doc2Vec_50	GNB	72.05%
CountVectorizer	LR	84.23%
Doc2Vec_200	LR	83.29%

Table 1 Accuracy compare for CountVectorizer and Doc2Vec fitting in NB and Logistic Regression Model.

3.2 Analysis

Ideally, Doc2vec should perform better in these subjective texts, but actually, its accuracy is lower than that of CountVectorizer. The reason might be that the dataset is not big enough for Doc2Vec to capture word relationship in the embedding space with limited information.

Therefore, I used CountVectorizer, the most straightforward and most intuitive way, to transform the review text into a sparse feature vector with the count of words.

4. Feature Selection

Since the number of columns is more than that of rows in the sparse matrix, feature selection can be used to avoid the risk of over-fitting.

4.1 TF-IDF & SelectKBest

The words recently appear in most documents, such as the word 'is', are not informative. TD-IDF can be used after CountVectorizer to scale down the frequently occurring words in the feature vectors. I use SelectKBest to select k features that have a stronger relationship with specific class.

4.2 Grid search

Grid search with pipeline was implemented to optimise the parameters for these two selection algorithms. To evaluate the accuracy, SGDClassifier is the better choice because we can change the hyperparameter' loss' to simulate the SVM and LogisticRegression model with less run-time.

After search, the best set of parameters is TF-IDF (norm='l2', sublinear_tf=True), SGD (loss='hinge') and use all features. Alternatively, I finally chose the set ranked 3rd, which has used SelectKBest (k=20000, score_func=chi2) and the other parameters are same as that of best one because compared with optimised one, it just decreases the accuracy by 0.00015% while using only ½ features in the following model implementation.

4.3 Analysis

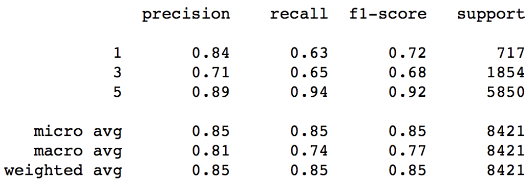
Overall, the improvement is not obvious after TF-IDF and SelectKBest. Partly because the issue is not complex at all. For SGD model, 'hinge' loss is preferred, which implies SVM is better than Logistic Regression after pre-processing since SVM has an advantage in dealing with high dimension dataset and can overcome the situation that number of features is more than the number of instances.

5. Model Evaluation-LinearSVC

A reasonable hypothesis is that the sparse matrix is **linearly separable** so that the linear model will perform better (Jonathan, 2018). Besides, the Grid Search in the last section showed that SVM would generate higher accuracy than LR. Accordingly, I implement **LinearSVC** at first.

5.1 Classification Report & Confusion Matrix

he results of implementing default LinearSVC with the pre-processed dataset are shown as below. The Classification Report (Figure1) and heatmap for Confusion Matrix (Figure 2) illustrate that SVC is most effective for label5, lable1 is the second, and lable3 is the worst.



```
             precision    recall  f1-score   support

          1       0.84      0.63      0.72       717
          3       0.71      0.65      0.68      1854
          5       0.89      0.94      0.92      5850

  micro avg       0.85      0.85      0.85      8421
  macro avg       0.81      0.74      0.77      8421
weighted avg      0.85      0.85      0.85      8421
```

Figure 1 The Classification Report for sparse matrix trained in default LinearSVC Model.

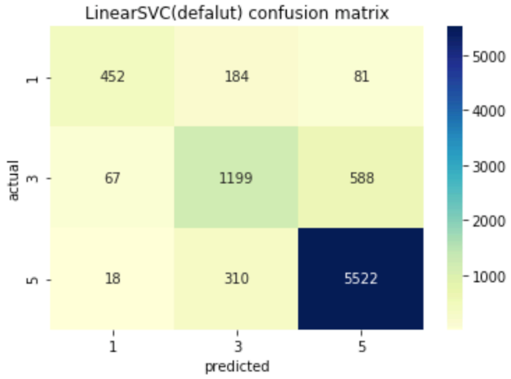


Figure 2 Heatmap of Confusion Matrix for default LinearSVC Model.

One of the reason is that the middle class is most ambiguous and confusing. Apart from this, the uneven distribution of class label may cause label1 performing not as good as label5. But normally, the result is excellent due to the dataset is linearly separable.

5.2 Validation Curve

For LinearSVC, we need to know the size of regularisation, which is the hyperparameter 'C'. Validation curve (Figure3) demonstrate the relationship between accuracy and 'C'.

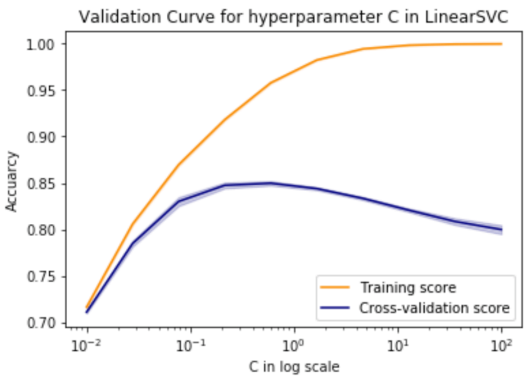


Figure 3 Validation curve for hyperparameter 'C' in LinearSVC

The training score is increasing because when 'C' gets larger, the decision boundary becomes stricter, then has a smaller margin, which contributes to the more accurate classification. However, the validation curve increases at first but decrease later, which implies the over-fitting has occurred.

Meanwhile, the variance increase with the decline of accuracy, which means the generalisation is not good. Because the validation data is unknown to model, and the unstable performance for the unknown data shows the model cannot be widely used. So I chose the 'C' with the highest validation score to improve my model.

5.3 Learning Curve

I generated a learning curve (figure4) to check for underfitting or overfitting (Kent, 2015). In general, the more significant the gap between two curves, the more over-fitting the curve is. It should be noted that the test score has a trend from low accuracy, high variance to high accuracy, low variance, which implies the computer studied well under this model. Moreover, the learning curve can show whether the machine has learnt enough by observing the curves' convergence. For this model, the test score still has growth trend even though it is small.

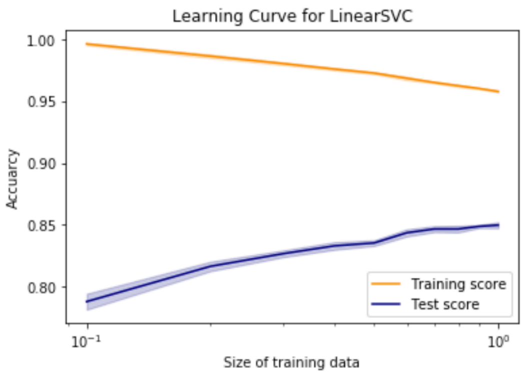


Figure 4 Learning curve for LinearSVC

6. Conclusions

In this report, I outlined the techniques used in feature selection and model evaluation. The result showed that data representation, learner choice and hyperparameter identification have a significant effect on prediction accuracy.

For this project, feature selection is the most critical part. Selecting some bad features may mislead the model. The final submission is the output that all training data with 20k best-selected features fed in LinearSVC and scored an accuracy of 84.99%.

7. References

Carenini, Giuseppe, Raymond T., & Adam P. Interactive multimedia summaries of evaluative text. In Proceedings of Intelligent User Interfaces (IUI). ACM Press, 2006. 124-131.

Jonathan, B. Linear SVC on yelp review dataset, 2018.https://medium.com/@gongon05 /linear-svc-on-yelp-review-dataset-4f93c64d1dd.

Kent, L. & James,R. Prediction of Yelp Ratings Based on Reviewer Comments Segmented by Business Type, 2015.

Mooney,R., Bennett,P. & Roy,L. Book recommending using text categorization with extracted information. In Proc, 1998. Technical Report WS-98-08.

Mukherjee, A., Venkataraman, V., Liu, B. & Glance, N. What Yelp fake review filter might be doing? 7th International AAAI Conference on Weblogs and Social Media, 2013.

Pang, B. & Lillian L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, 2004. 271–278.

Qu, L., Ifrim, G. & Weikum, G. The bag-of-opinions method for review rating prediction from sparse text patterns. In Proceedings of the 23rd International Conference on Computational Linguistics, 2010. 913-921

Rayana, S. & Akoglu, L. Collective opinion spam detection: Bridging review networks and metadata. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015. 985-994.

Susan, L. Scikit-Learn for Text Analysis of Amazon Fine Food Reviews, 2017. https://towardsdatascience.com/scikit-learn-for-text-analysis-of-amazon-fine-food-reviews-ea3b232c2c1b.

Yi, L. & Xiaowei, X. Predicting the Helpfulness of Online Restaurant Reviews Using Different Machine Learning Algorithms: A Case Study of Yelp, 2019.