Text Classification Kaggle Project Report

HUANG Sixuan (A0049228B)

Introduction

This project is a text classification problem for the comments posted on two Q&A websites. All of these comments are classified into 16 different topics.

1. Data Overview

There are 955,454 comments in the training data and 552,735 comments in the test data. No null value is found. The training data are classified into 16 different topics, as summarized in **Table 1** based on word clouds (**Appendix A**) on the training data.

Table 1. Topics of Different Classes

Class	Topic	Class	Topic
0	WordPress App	**8**	Drupal App
1	Travelling	**9**	Games
2	Apple Related	**10**	Movies
3	Android Related	**11**	Blockchain & Bitcoin
4	Astronomy	**12**	Physics
5	Electronics	**13**	Geography
6	Economics	**14**	Math
7	Fictions / Books	**15**	Politics

One important finding on the training data is that it is highly unbalanced. As shown in **Figure 1**, Class 14 has a proportion of 50.7% in the training data, while proportions of other classes are only 0.2% to 8.7%.

It is also noted that there are many special characters, punctuations and non-English languages (Chinese, Japanese, etc.) in the data.

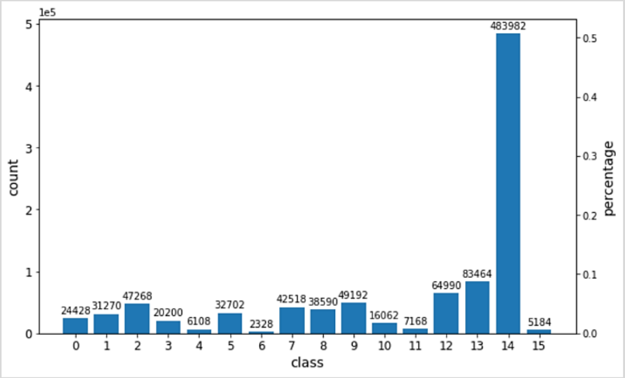


Figure 1. Classes in the Training Data

2. Base Case Description

The base case is intentionally designed to be simple. Leaderboard score of the base case is 0.54964, which is above the benchmark score of 0.52536. Detailed descriptions of the base case are as below.

- No training data sampling is carried out. The data imbalance problem is dealt with at a later stage.
- No text pre-processing is carried out. It is found that text cleaning could undermine some features (special characters etc.) which actually help to identify classes like math and physics etc.
- CountVectorizer is used for text vectorization because it is found to provide better results than other methods including TF-IDF vectorization.
- Multinomial Naïve Bayes model is used since it is proven to be effective and efficient for text multi-classification with a large data scale.
- Parameters are tuned with cross validation (cv=5, ‘accuracy’) and heuristic Bayesian Optimization. min_df=2 and token_pattern=’\b\w\w+\b’ for the CountVectorizer and alpha=0.2395 for the MultinomialNB model are finally obtained.

3. Data Imbalance Problem

Although the base case model does not make perfect predictions, looking into its results can provide important insights on the test dataset. **Figure 2** shows that the test dataset also suffers from severe data imbalance problem.

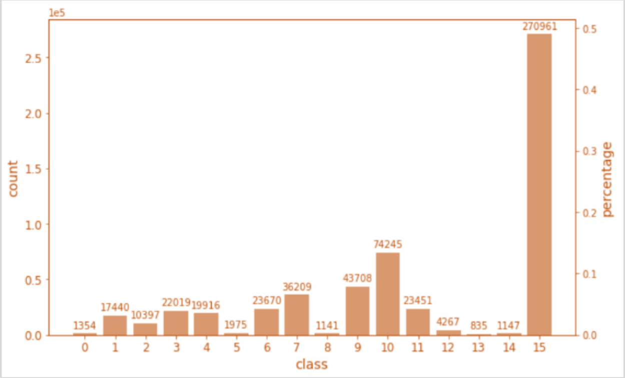


Figure 2. Predicted Classes (Base Case)

It is crucial to keep in mind the significant different class distributions between the training data and the test data. Improper treatment of the training data can easily lead to overfitting problems (classes 14, 13, 8 etc.) or underfitting problems (classes 15, 11, 10 etc.). Classification reports should be carefully analyzed during training, so that improvement measures can be taken for each class accordingly.

To solve the data imbalance problem, both training and test aspects need to be taken care of. For the training data, 100,000 samples are drawn for each class to form a balanced dataset. For Class 14 which originally has more than 100,000 data, sampling without replacement is made. For all other classes, sampling without replacement is carried out.

To overcome the severe training data insufficiency problem for some classes, the 'Pseudo-Labelling' method is used to fill the gap between training and test data. That is, those test data with predicted probability of more than 0.95 are added to the training set for a second round of training. This method sometimes leads to overfitting problems, but it works well for this project, probably because the original training data is too little or with too much noise for some classes.

The two parameters mentioned above, the sampling amount of 100,000 and the probability threshold of 0.95, are not optimized due to the limit of time.

4. Feature Engineering and Selection

Basic text cleaning is carried out to remove '&#xA;', '&#xD;' and white spaces before feature engineering. More than 40 manual features are tried but only 21 of them are kept based on a LGBM model for feature selection. Training and test data are split with a ratio of 0.3 for the test data. Care is taken of to make sure that there is no overlap between the training and the test datasets. Data sampling is done to make both training and test data balanced, therefore 'accuracy' can be used as a proper metric for evaluation during the feature selection process.

Statistical Features

Statistical features are created by counting or computing the density of different types of the words in the text. **Appendix B** can be referred to for details. As shown in the plots, the created features have different statistical characteristics among different classes. This explains why they can help to improve the prediction results in this text classification problem.

LDA-based Features

Latent Dirichlet allocation (LDA) is carried out to get the most frequent word lists for each class. Then the words that are not specific to a topic or are duplicated in various topics are removed from the word lists. Additional words are added to the list based on an overview of the word clouds and the text contents of the training data. The final words lists (**Appendix B**) are then used as class dictionaries for counting topic-related keywords in each of the text samples.

5. Model Ensembling and Final Results

Efforts have been made to try out KNN and SVM models. However, they are very inefficient for the large dataset since they both involve computing a large number of distances. Therefore, Multinomial Naïve Bayes is used as the final model.

Proper model ensembling can help to improve the prediction scores. However, improvement is not achieved after limited attempts in this project (see **Table 2**). This is because the original models have either too different prediction scores or too similar prediction behaviors. For model ensembling to work, we would need to have good models with similar high scores but focus on different aspects, so that they can complement each other's weaknesses.

The final results are summarized in **Table 2**, with the highest Kaggle leaderboard score as 0.65384.

Table 2. Final Analysis Results (Selected)

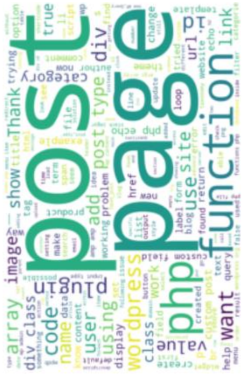
Case	**Base_0**	**Model_1**	**Model_2**
Desc.	See Section 2	Statistical Feat.	LDA-based Feat.
Score	0.54964	0.64244	0.65372
Case	**Model_3**	**Ensemble 1+2**	**Ensemble 0+3**
Desc.	All features	Model 1+2	Model 0+3
Score	**0.65384**	0.65098	0.64264

6. Possible Improvements

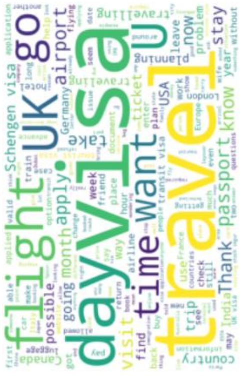
Due to the limit of time, many aspects have not been optimized. Possible improvements include:

- More proper data pre-processing to remove possible noises to get higher quality data.
- More class-specific feature engineering to further improve those poorly predicted classes.
- Better parameter optimization and try out more complex models if time and resource permit.
- Proper design of the running cases to make the final model ensembling more effective.

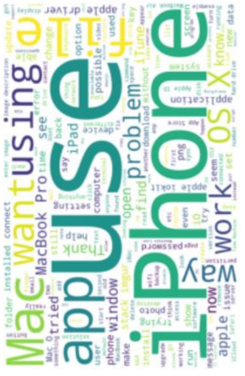
Appendix A: Word Clouds for Different Topics



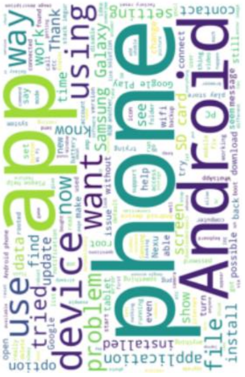
Class 0: WordPress App



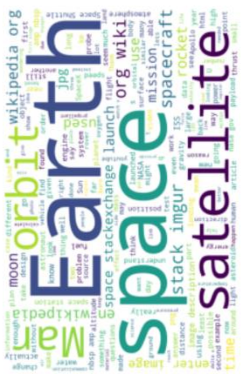
Class 1: Travelling



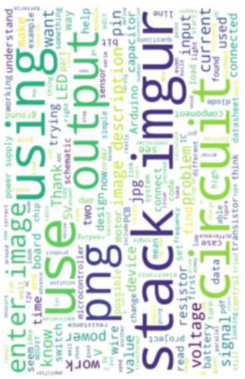
Class 2: Apple Related



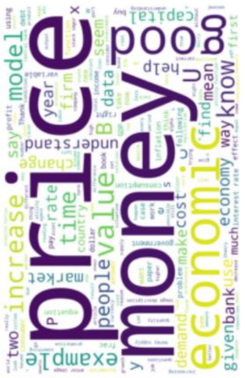
Class 3: Android Related



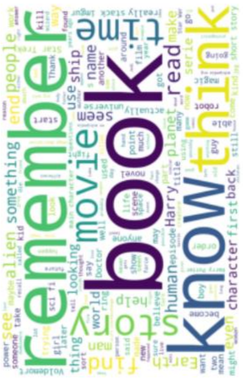
Class 4: Astronomy



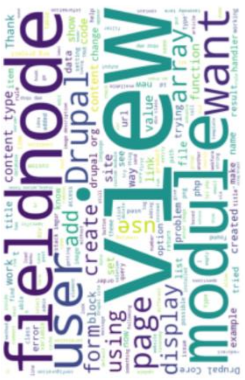
Class 5: Electronics



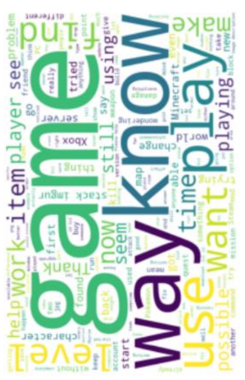
Class 6: Economics



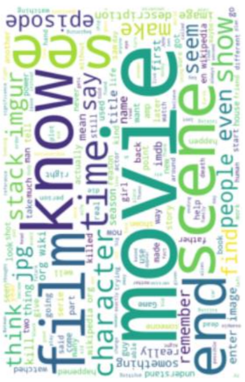
Class 7: Fictions / Books



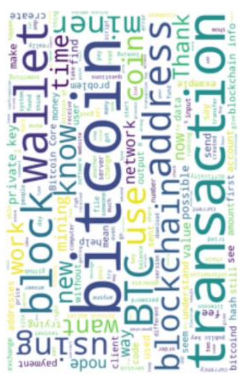
Class 8: Drupal App



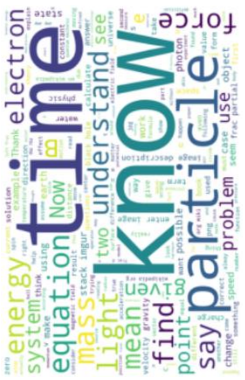
Class 9: Games



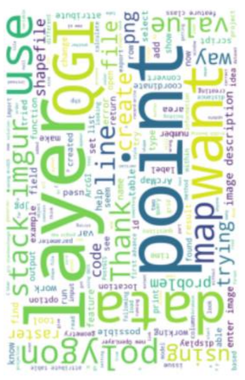
Class 10: Movies



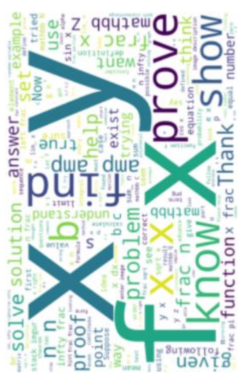
Class 11: Bitcoin



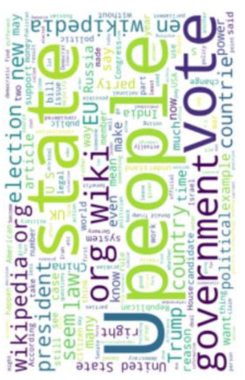
Class 12: Physics



Class 13: Geography



Class 14: Math



Class 15: Politics

Appendix B
Manual Feature Plots and
LDA-based Keyword Lists

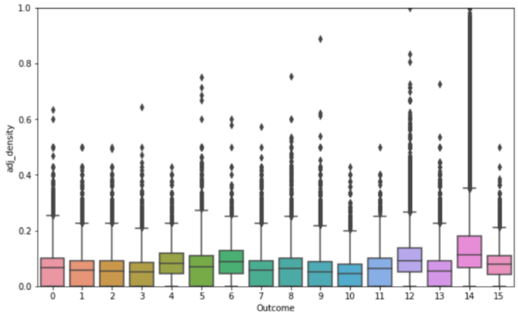


Figure B.1. Density of adjectives
Calculated by dividing the total number of adjustives by the total number of words in a comment.

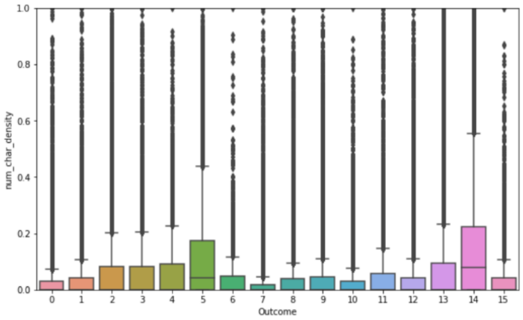


Figure B.2. Density of numerical characters
Calculated by dividing the total number of numerical characters by the total number of words in a comment.

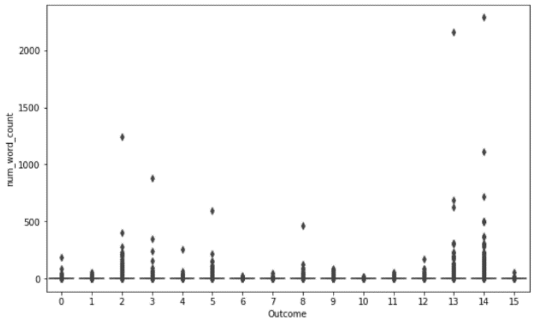


Figure B.3. Count of numbers (as words)
Calculated by counting of the words whose characters are all numerical.

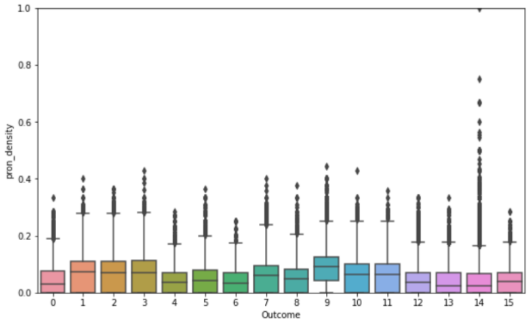


Figure B.4. Density of pronouns
Calculated by dividing the total number of pronouns by the total number of words in a comment.

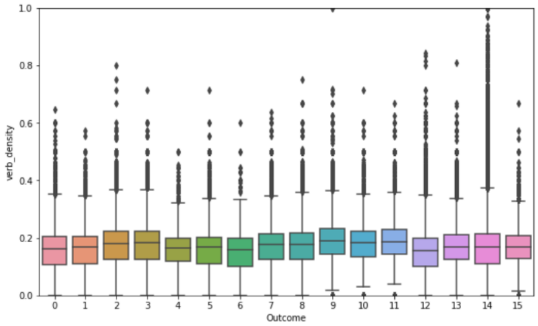


Figure B.5. Density of verbs
Calculated by dividing the total number of verbs by the total number of words in a comment.

Class	Topic	Word List
Class 0	WordPress App	php post page wordpress echo code posts id plugin plugins development function functions
Class 1	Travelling	visa passport air travel flight airport visit uk us schengen customs immigration transit train luggage ticket
Class 2	Apple Related	mac iphone macbook os apple macos ios pro itunes ipad icloud applescript safari imac
Class 3	Android Related	android device google google-play galaxy samsung screen
Class 4	Astronomy	space spacex spacecraft earth moon mars orbit launch rocket rockets spacecraft satellite nasa apollo
Class 5	Electronics	voltage circuit power power-supply current microcontroller transistor battery amplifier resistor
Class 6	Economics	money price rate value demand tax stock income invest credit credit-card mortgage loan bank trade real-estate
Class 7	Fictions / Books	aliens anime book books magic marvel novel read remember story stories
Class 8	Drupal App	block blocks database drupal field module node nodes form forms content node nodes file files user users
Class 9	Games	achievement achievements dota game play pc ps3 ps4 pokemon pokemon-go level xbox
Class 10	Movies	cast casting character cinema dialogue ending movie plot production scene title film films episode
Class 11	Blockchain & Bitcoin	bitcoin wallet transaction address block btc bitcoins transactions mining blockchain
Class 12	Physics	dynamic electro energy fluid force gravity magnetism mass mechnics optics partial quantum light theory wave
Class 13	Geography	arcmap arcpy arcgis geoserver gdal openlayers openstreetmap qgis postgresql postgis pyqgis layer map raster
Class 14	Math	amp calculus derivatives differential frac geometry inequality integration linear-algebra mathbb matrices polynomials probability prove infty function sqrt problem set statistics
Class 15	Politics	republican bill senate republicans states country government president question countries democrats state vote law obama political us house trumps romney court trump gop party