One of the deliverables for the Amazon Multi-Class Classification project is a write-up. The goal of this write-up is for you to a) explain what you learned about the data, b) describe what processing and modeling choices you made, and c) discuss how your model performed, given your choices. To give you a clearer understanding of what we expect in this write-up, we’ve provided this outline. Your project write-up should include the following sections and information.

***Be sure to include your Kaggle username and your Kaggle team name that appears on the leaderboard.***

# What is your Kaggle ID? [5 pts] Xinyue Liang lxyskywalker

# Peijun Qing Kris55

# Team: Kris55

# Data:

* Describe the dataset you worked with, including explanations of interesting feature

variables and the target variable related to this task. Highlight if and how you have changed the feature engineering process in this task from the binary classification task.

As the dataset is the same to the last project, the basic method of us handling the data does not change, ie. Convert train data into a long string separated by comma between each feature and convert the target feature into integers. The feature variables we choose are categories, votes, summaries, and review texts. Among the four features, summary and review texts were from last project, so we won’t put too much emphasis on them here. We chose both new features as we tested adding each original feature one by one, and these two bring the most enhance in F1 score. As for the votes, we converted the Non entries to 0, and converted the integers into strings to better append to the train and test string. Then we applied the category one twice since it’s the only extraction we tried which does not cause overfit on train set. The extraction method is described in the following method part with more detail

# Methods:

* Describe the steps you took to build your model.

1. Feature extraction, we tried to extract keywords such as “stars” and “waste” from text based features such as summary and review text. Although they bring us significant improve during the local split test, they do not perform well on the Kaggle test, which means they could be overfitting the train data. The only feature we found would not cause a overfitting issue is the category, as itself is roughly the same length as keywords, we made a double entry of categories information into the datasets.
2. The second method we tried is trying to use the stemmer and wordlemmatizer to achieve data cleaning. However, this does not make a significant difference on the result. We also tried to remove the stopwords from the data entries, but it only has negative effect.
3. Then we vectorized the string entries like we did in the binary classification project. But what’s the difference is that we altered the max feature limitation for the vectorizer from 5000 to 50000. Also we applied the selectKBest selector to find out the most useful features.
4. The final method the tried is SMOTE, however, we did not use it for the final run as we are not sure if it is allowed. What it does is remove the “skewness” of data by deleting several data points.

* Describe the hyperparameters you used in your best model. Describe how you did your hyperparameter search. What are the things that you did differently than the binary classification task?

The two main hyperparameters we used in our best model (MLP) is the random state and the max iteration. When comparing to the binary classification task, the multi class’s result seems to be more sensitive to the hyper parameters. As for the random state, we tested several values with each increment by 5 from 0 to 40, and 5 gives us the best result. As for the max iteration, it has more influence than the random state. We discovered that the model tends to overfit with more iterations. So we set the suitable range to be 1 to 10 and tried each number in this range, the ideal time of iteration is 4.

# Results:

* Show the confusion matrix, ROC score, macro F1 score, and accuracy for the best set of hyperparameters using 5-fold cross-validation for each of your three models.

Confusion matrices:

Logistic Regression:

[[816 234 55 22 28]

[300 624 185 74 32]

[ 91 237 583 185 58]

[ 38 81 172 679 201]

[ 36 37 45 181 844]]

MultinomialNB

[[752 324 51 14 14]

[231 750 172 48 14]

[ 57 313 619 128 37]

[ 15 143 219 665 129]

[ 19 80 72 230 742]]

MLPClassifier

[[800 258 49 23 25]

[250 682 193 72 18]

[ 71 256 580 203 44]

[ 17 70 151 753 180]

[ 13 33 40 223 834]]

Performance table:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | ROC\_AUC\_Score | Macro F1 | Accuracy |
| Logistic Regression | 0.883782960877786 | 0.6069554305418782 | 0.6073997944501541 |
| MultinomialNB | 0.880689119414332 | 0.610075943245367 | 0.60431654676259 |
| MLPClassifier | 0.8961791015306968 | 0.626194499020307 | 0.6250428228845495 |

* List the score you achieved using Kaggle

0.62258

* What is a suitable metric for this task?

Macro F1 or Accuracy, we also need to compare to train data metric to avoid overfit

* Does your best model provide the best performance across all metrics? If so, why?

Yes. Our guess is that as a Multi layer neural network, the additional layers and the relatively more iterations it requires makes it perform better

* Which class(es) are more challenging to predict? Can you reason why they are harder to predict?

We think that 2 and 4 are more challenging to predict. The data belongs to 2 and 4 are relatively more rare in the train set. The SMOTE we tried but did not use might be a tool to address this issue

# Future Work

* Describe potential future work: what would you do differently next time?

For this assignment, we give up on several methods which might be useful because of the overfit issue due to the time limitation. So we might consider properly handle those overfit problems if we have sufficient time. On the other hand, we might try out other models and other data entry formats which could be interesting.

Note: before we started the project, we used Bert to try out on the datasets, but Kaggle does not allow us to remove those two submissions. The final results we choose are totally under the rules and are above the baseline which has a 0.62258, please refer to those two results instead of the highest two. Thank you! A screenshot of a computer

Description automatically generated