ML1010 – Group Project
Milestone 4

Submitted by: Michael Vasiliou

Milestone Summary

Project recap:

The data being used in these experiments is the Amazon dataset being filtered down to cell phone reviews.   The review text was cleaned, stop words removed, and lowercased. Analysis and comparison was completed on three encodings (Flair, Bert, Textblob) and used in a sentiment comparison using a target of the 1-5 star rating. During this analysis it became apparent that matching text sentiment to something as arbitrary as a 1-5 star user submitted rating was less than ideal. As such comparisons were performed on all three encodings, using XGBoost as the model, and comparing both 5 star sentiment analysis, as well as a normalized positive/negative rating which was mapped from the 5 star sentiment.

The results were that BERT was by far the best of the 3 encodings, and the 2 star rating was a much more reasonable measure to attempt to train the model towards.

In Milestone 3 I added new models, Random Forest, LSTM, and Bi-Directional LSTM to the exploration as well as two new encodings, Glove, and MPNet. All models were run after conducting a feature importance analysis to shrink the feature set of each encoding.

The results showed that there were no noticeable differences in model performance in any of the combination of encodings or models.

Milestone methodology:

This milestone uses two different models:

- Base model: Logistic Regression
- Secondary model: XGBoost

Two data sizes were used:

Small: 21,504
Large: 283,116

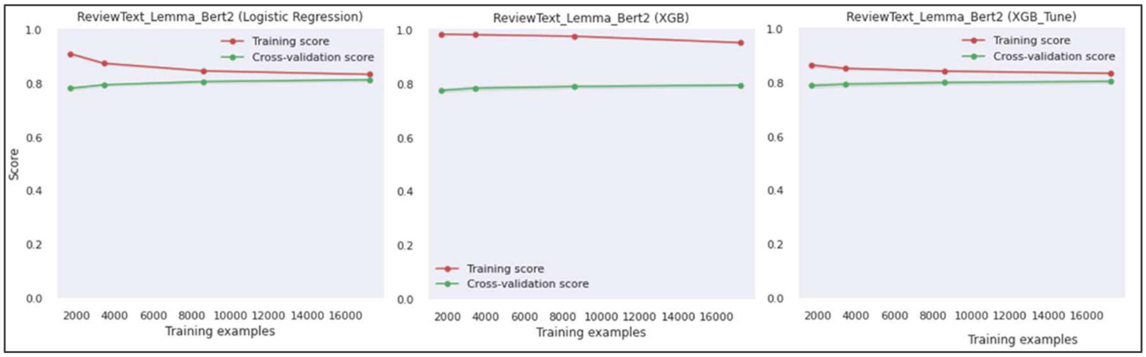
All encodings were done using BERT (multi-qa-distilbert-cos-v1) except where noted in phase three.

Milestone 4 consisted of three phases:

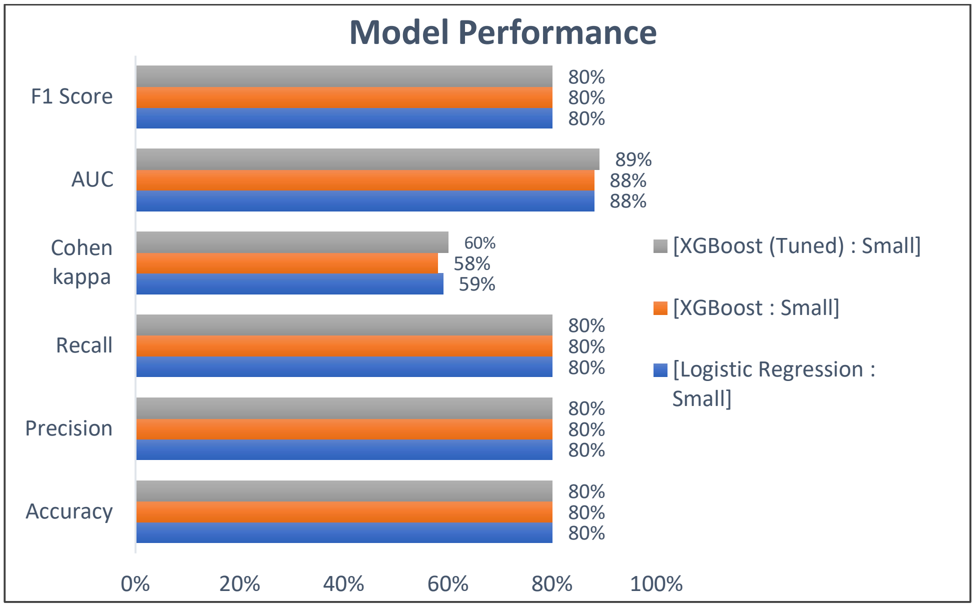
1. Run analysis using base model, secondary model, tuned secondary model
2. Repeat phase 1 experiments but with larger dataset
3. Conduct performance analysis between experiments

Milestone 4: Phase 1

The validation curve results were interesting and showed that the default Logistic Regression and the tuned XGBoost models came close to converging under the current data set size. The default XGBoost model with no tuning applied was far from converging.



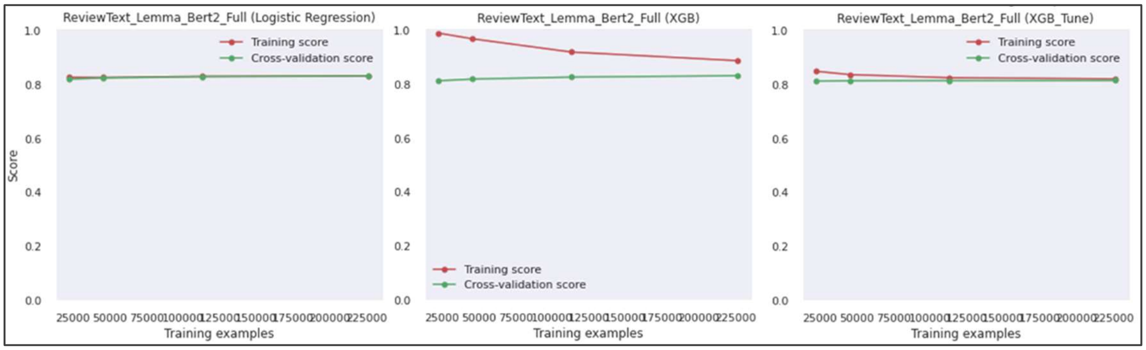
The models themselves performed virtually identically with no clear winner.

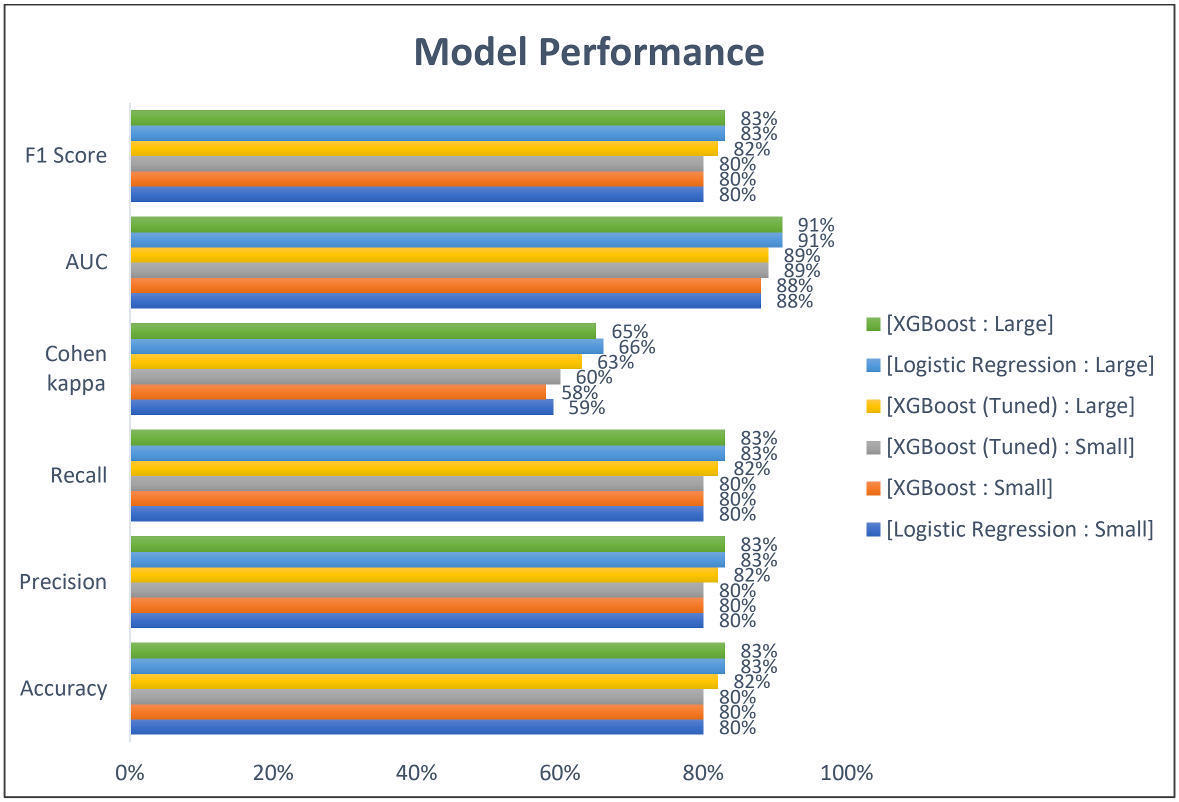


Milestone 4: Phase 2

With none of the models converging and no clear performance winner a second stage was added for Milestone 4. The original dataset was the Amazon small dataset, however for this round the entire Amazon dataset was utilized. Identical encoding, preparation, and analysis of base models were conducted. All model combinations were then re-run using the large dataset. The small dataset had 21,504 records in the training data set, while the large training data set had 283,116.

The validation curves show convergence on both the Logistic Regression model as well as the tuned XGBoost model. The Logistic Regression model converged utilizing a much smaller dataset. As expected with a larger dataset, the new models provided improvement in model prediction power but did not provide any significant differentiation between the performance of the models.





Milestone 4: Phase 3

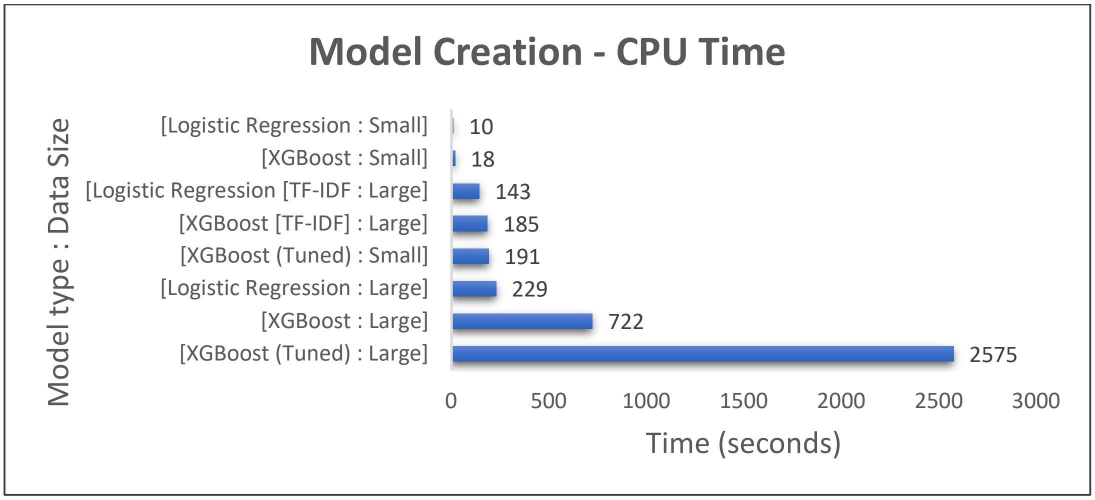
All models at each of the two data sizes performed equally and there was no clear winner. The only noticeable difference between the models was the time/compute power required to generate and evaluate each type of model. This phase was added to address those differences.

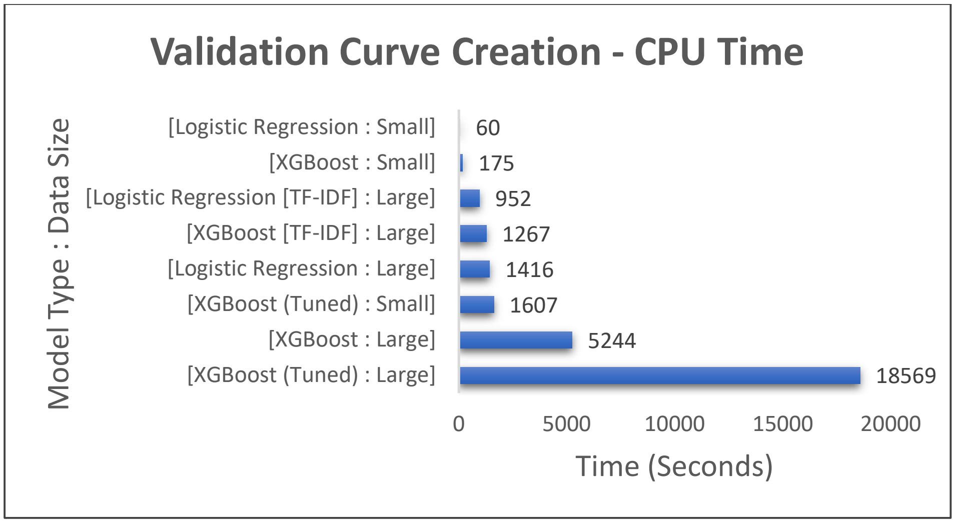
Each of the models was measured on CPU Time for both the validation curve and the model creation. Wall time was not included as multi-core processing was included. All measurements were conducted using the same CPU, no external running processes, and no GPU computations.

As this phase is focused on computation resources two additional model/encoding combinations were included, XGBoost using TF-IDF, and Logistic Regression using TF-IDF both run on the large dataset. TF-IDF was added as encoding with TF-IDF is significantly faster than encoding with BERT.

CPU Time scenarios:

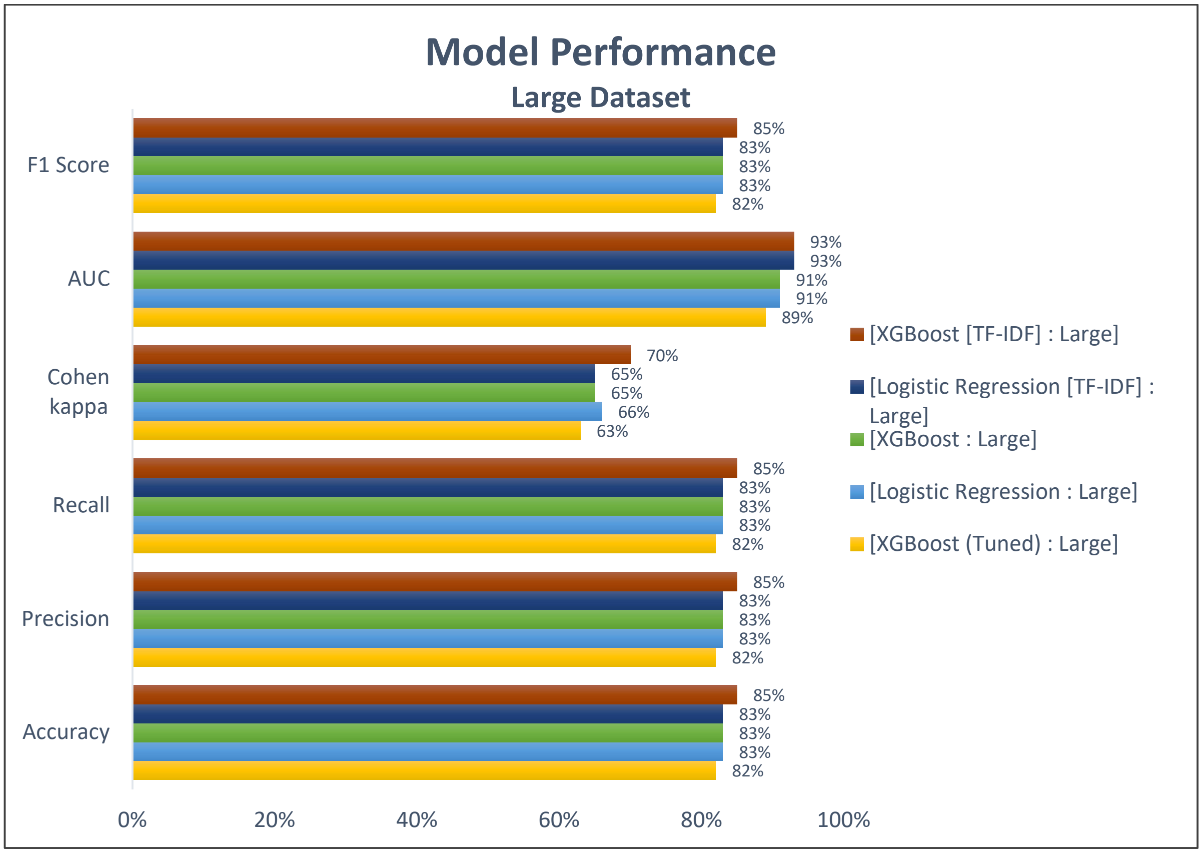
Comparison	Fastest CPU Time	Slowest CPU Time
Model Creation – small dataset	10 seconds	191 seconds (3min 11sec)
Validation Curve – small dataset	60 seconds (1min)	1607 seconds (26min 47sec)
Model Creation – large dataset	143 seconds (2min 23sec)	2575 seconds (42min 55sec)
Validation Curve – large dataset	952 seconds (15min 52sec)	18,569 seconds (5hr 9min 29sec)





Findings:

The large dataset provided better results than the smaller dataset and based on the results of Phase 3 we have seen that there are models with the large dataset that use less CPU Time than some of the models of the small dataset.   The included TF-IDF with XGBoost provided the best results followed closely by the Logistic Regression with TF-IDF.



The final model choice is Logistic Regression on the large dataset with TF-IDF encoding. The XGBoost model with TF-IDF provides slightly better results but with a slight increase in CPU Time. The use case for this scenario is based on sentiment analysis of the Amazon dataset and does not carry the same importance as a medical scenario where a percentage point could mean something significant to the patients wellbeing.

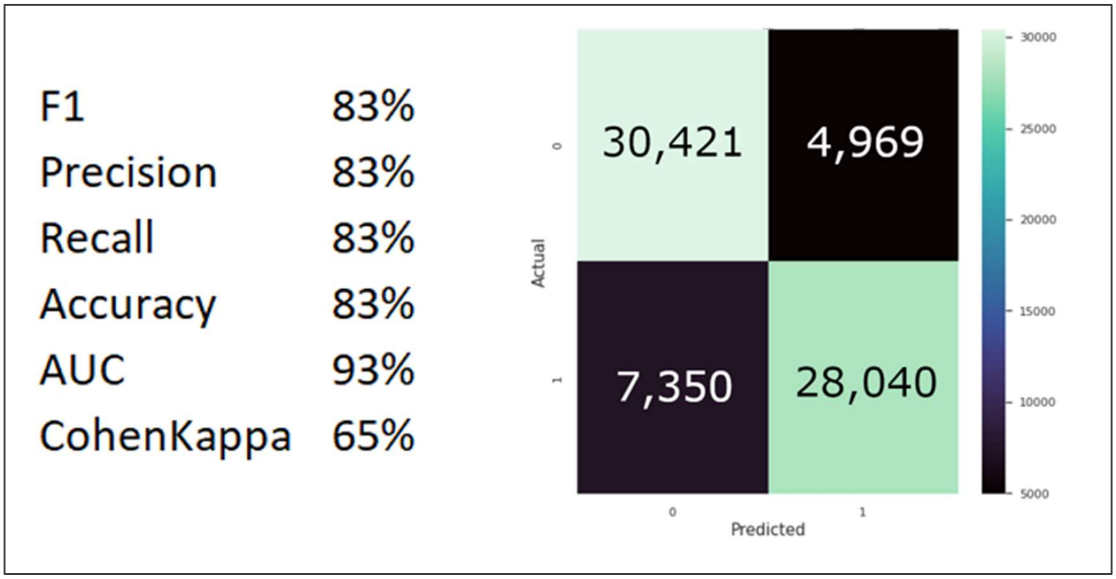
Additionally, the validation curve for Logistic Regression shows that it would be able to converge with a much smaller dataset and provide equivalent results. This would further improve modelling time.

TF-IDF was chosen as the base encoding instead of BERT. The BERT encoding is very resource intensive to implement compared to TF-IDF. Additionally, TF-IDF also provides an understanding of important features based on words instead of the random encoding column. This allows the model to be more transparent as well.

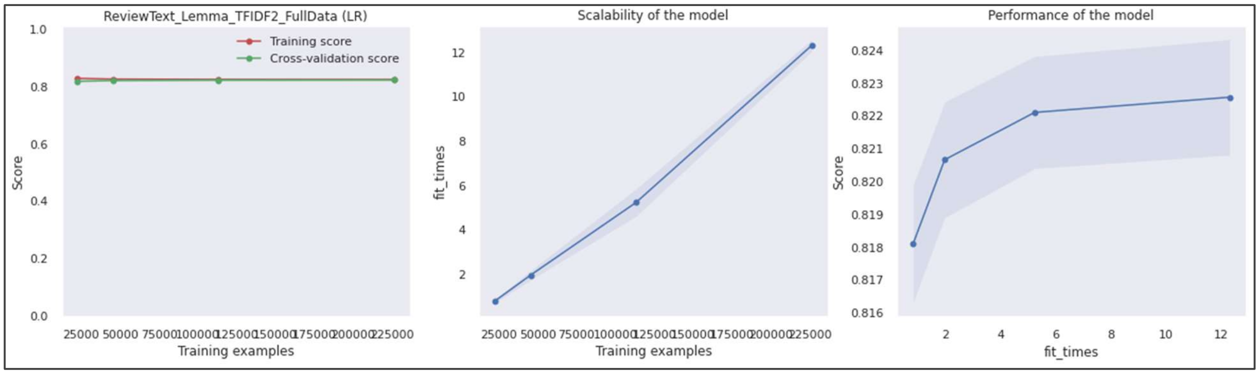
It is understandable that a simpler model and simpler encoding would provide relatively equal performance to the more robust XGBoost and BERT encoding. The dataset and use case is incredibly simplistic with very little nuance in the data. Using XGBoost and BERT encoding which are extremely resource intensive is somewhat of an overkill.

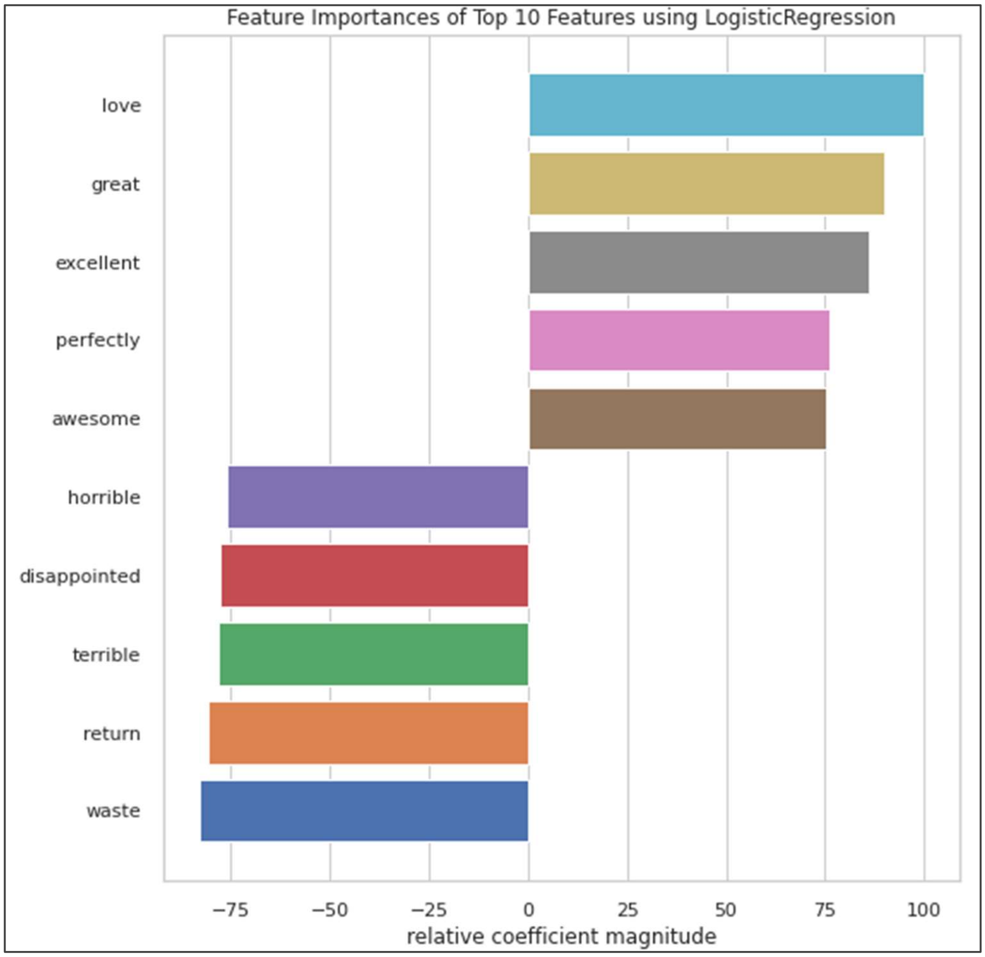
Final Model Summary:

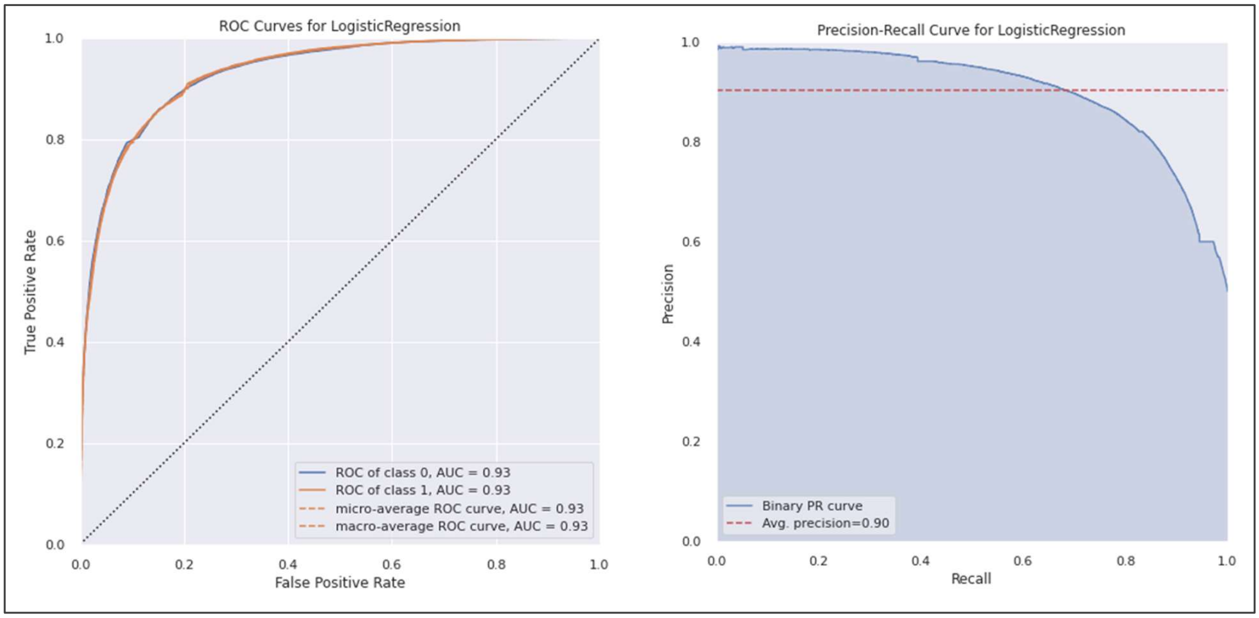
Model: Logistic Regression
Data size: Large dataset (283,116)
Encoding: TF-IDF



F1	83%
Precision	83%
Recall	83%
Accuracy	83%
AUC	93%
CohenKappa	65%




Feature Importances of Top 10 Features using LogisticRegression
love
great
excellent
perfectly
awesome
horrible
disappointed
terrible
return
waste
−75
−50
−25
0
25
50
75
100
relative coefficient magnitude


ROC Curves for LogisticRegression
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC of class 0, AUC = 0.93
ROC of class 1, AUC = 0.93
micro-average ROC curve, AUC = 0.93
macro-average ROC curve, AUC = 0.93
Precision-Recall Curve for LogisticRegression
Precision
Recall
Binary PR curve
Avg. precision=0.90