ML1010 – Group Project
Milestone 3

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.

Model and Encoding for Milestone 3:

For Milestone 3 I am continuing with the sentiment analysis using positive/negative and BERT encodings as my new base, but will be adding new models and new encodings to the exploration. In this iteration I have added two new encodings (Glove, MPNet), and two new models (Random Forest) and NN(LSTM).

Embedding types included in this Milestone:

- Bert – “distilbert-base-nli-mean-tokens”
- Glove – “average_word_embeddings_glove.6B.300d”
- MPNet – “all-mpnet-base-v2”

Model	Encoding
Random Forest	Bert
Random Forest	Glove
Random Forest	MPNet
XGBoost	Bert
XGBoost	Glove
XGBoost	MPNet
LSTM	Bert
LSTM	Glove
LSTM	MPNet

Feature importance analysis

A feature importance analysis was conducted on the each of the models and the top features for each model based on feature importance were retained. In each case the number of features diminished drastically.

Using the top features selected for each experiment, a new model was created with the same parameters. Reduction of features from 768 (for BERT encoding) to 40 or less was achieved with no loss in performance. A significant reduction in model creation and evaluation time was achieved as well.

Additional Reporting:

Additional reporting metrics and charts were added to compare and evaluate models:

- (previously included) Precision, Recall, F1, Accuracy
- (previously included) metrics classification report
- (previously included) confusion matrix
- (new) AUC/ROC curve
- (new) Cohen kappa
- (new) Scalability of model (Training examples x fit times)
- (new) Performance of the model (fit times x score)
- (new) Training and Validation Accuracy
- (new) Training and Validation Loss
- (new) Training and Validation MSE
- (new) Training and Validation AUC
- (new) Training and Validation Precision
- (new) Training and Validation Recall

Tensorflow and NN models:

This is the first time including any neural networks within the analysis. The methodology for implementing and analyzing the models:

1. Initial model creationCreated a simple 1 layer LSTM 100 units
    a. Created a simple 1 layer LSTM network to get it working
    b. Results were below those of Random Forest/XGBoost, but not horrible
    c. Training time was significant
2. Feature importance
    a. In order to address the long training times the previously calculated top features from the XGBoost experiment were used to trim the features
    b. Inputs trimmed from 768 to 40
    c. Model fit times were sped up considerably and accuracy and scoring were improved as well
3. Model Variations
    a. With the significant increase in speed model variations and explorations were included. Added variations of dense, dropout, dropout rates, lstm output units, CONV1d, MaxPooling1D

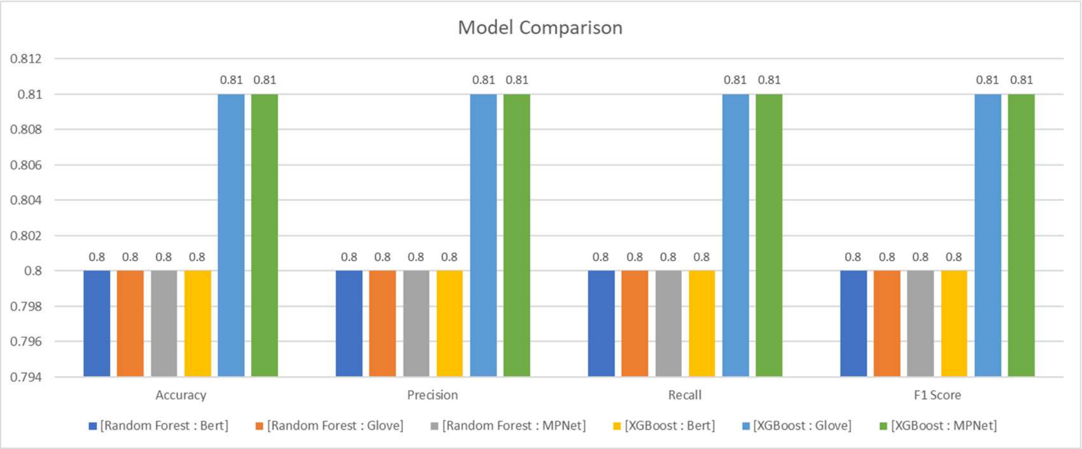
b. Experiment with changing number of nodes per layer

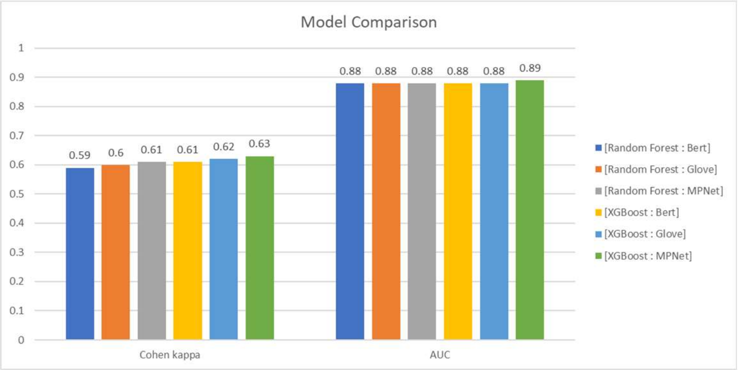
Note: Tensorflow models were added very recently and more work needs to be completed on:

- Keras feature importance
- Layer optimization/inclusion

Model Summary:

All models performed far more similarly than I would have expected. Not included in the below charts are the LSTM models as I have only completed training and validation scores. The scores on test data are currently underway. The LSTM experiments have additional charting (see further on in the experiment section) which includes additional metrics and charting



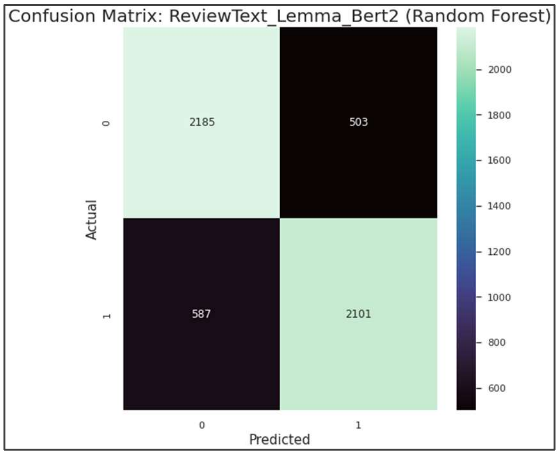


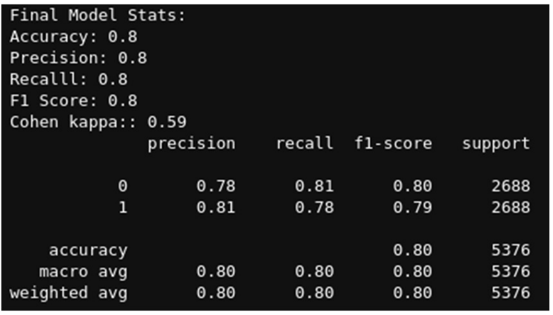
Next Steps:

- Run LSTM models using Glove and MPNet
  - Current analysis of LSTM only included BERT
- Conduct a Keras feature analysis
- Examine variations on the NN models and layers.
  - Explore variations and options on Tensorflow layers to improve
  - Tensorflow model sitting around 76% accuracy, while XGB and Random Forest are around 80%
- Add in addition reporting for the Tensorflow models
- Add in feature importance based on words for each model
- Run some optimizations on other models (XGB, Random Forest)
  - Current models use base model with mostly defaults
  - Opportunities exist for improvement

Data Experiment: Random Forest – Bert Encoding

Confusion Matrix and Summary Grid:



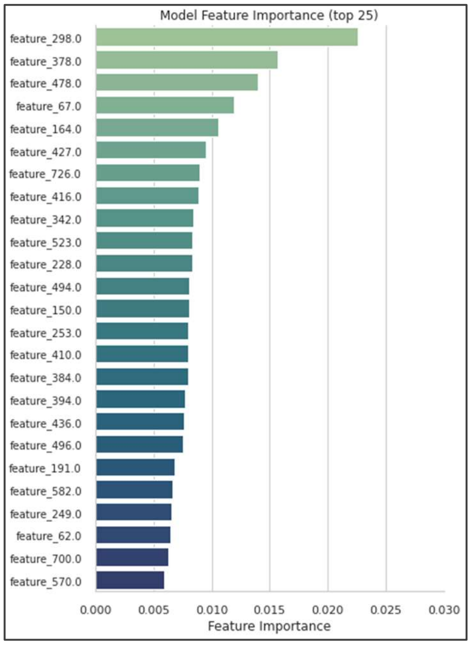


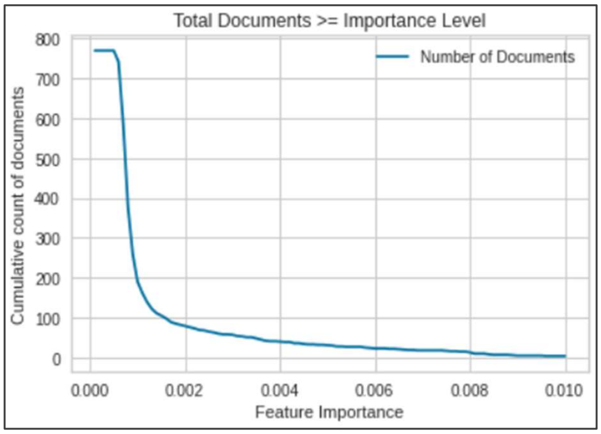
```
Final Model Stats:
Accuracy: 0.8
Precision: 0.8
Recalll: 0.8
F1 Score: 0.8
Cohen kappa:: 0.59
              precision    recall  f1-score   support

           0       0.78      0.81      0.80      2688
           1       0.81      0.78      0.79      2688

    accuracy                           0.80      5376
   macro avg       0.80      0.80      0.80      5376
weighted avg       0.80      0.80      0.80      5376
```

Feature Importance:

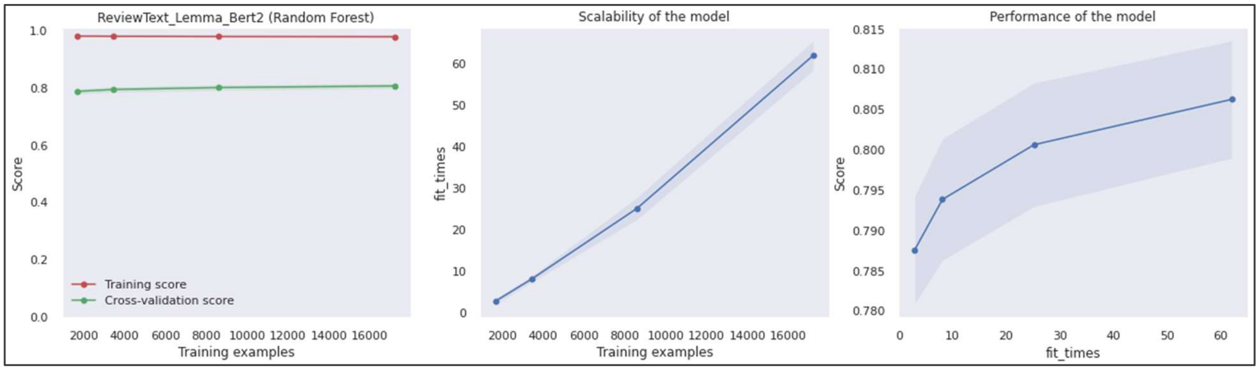
Graph displays number of features that have a feature importance of at least the value on the x-axis. For example there are 768 features in this experiment and all features have importance greater than 0. There are only about 50 features that have an importance of .004 or greater.

The Model features importance (right chart) shows the top 25 features in descending order of importance.

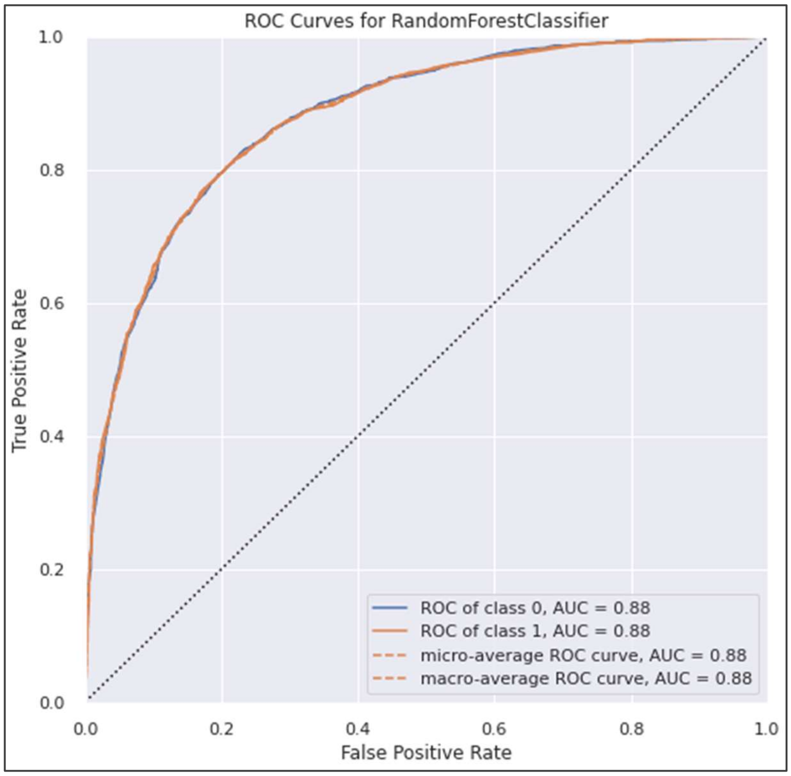




Learning Curves and Scalability

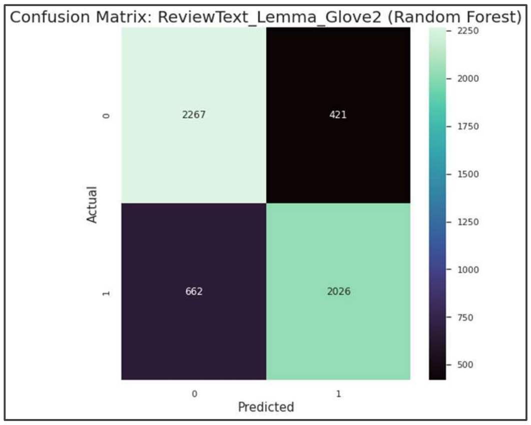


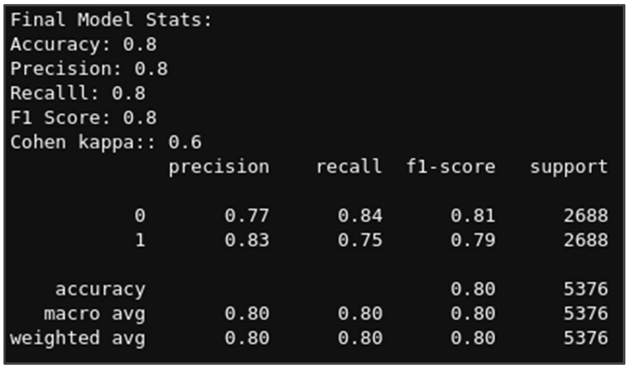
ROC/AUC Curves



Data Experiment: Random Forest – Glove

Confusion Matrix and Summary Grid:



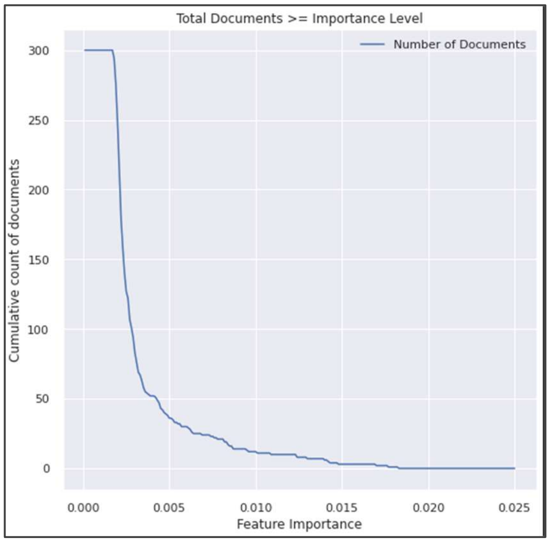


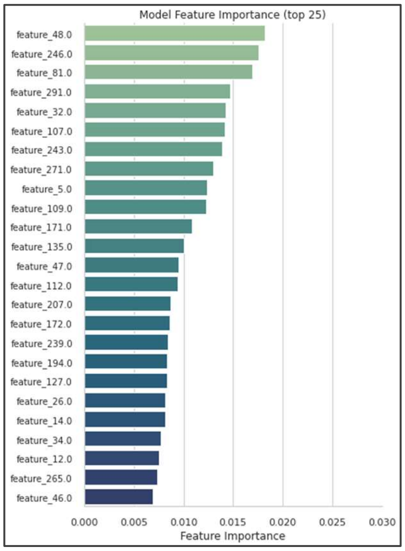
```
Final Model Stats:
Accuracy: 0.8
Precision: 0.8
Recalll: 0.8
F1 Score: 0.8
Cohen kappa:: 0.6
              precision    recall  f1-score   support

           0       0.77      0.84      0.81      2688
           1       0.83      0.75      0.79      2688

    accuracy                           0.80      5376
   macro avg       0.80      0.80      0.80      5376
weighted avg       0.80      0.80      0.80      5376
```

Feature Importance:

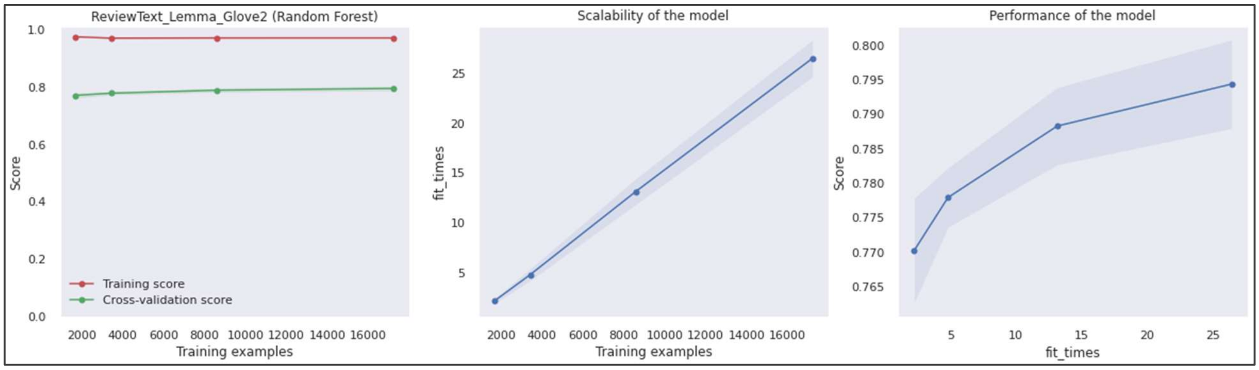
Graph displays number of features that have a feature importance of at least the value on the x-axis. For example there are 300 features in this experiment and all features have importance greater than 0. There are only about 15 features that have an importance of .01 or greater.

The Model features importance (right chart) shows the top 25 features in descending order of importance.

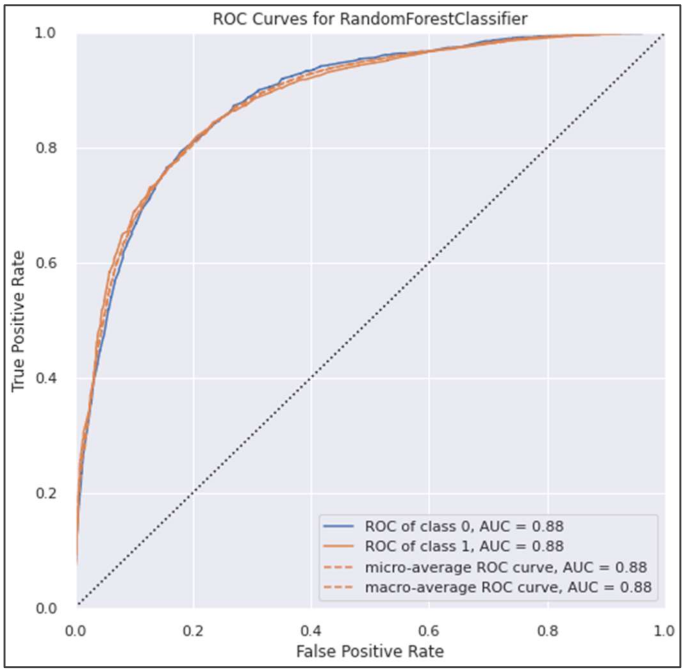




Learning Curves and Scalability

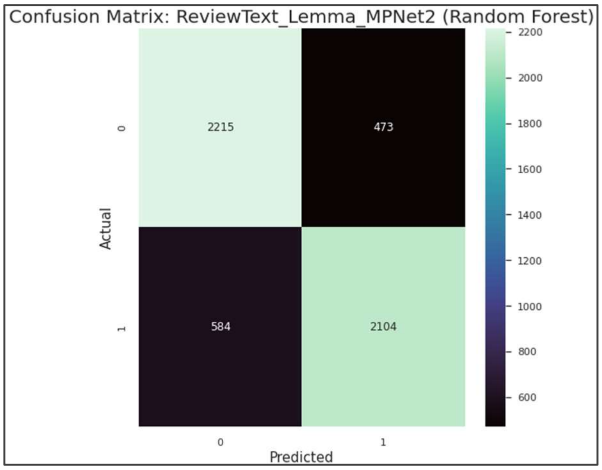


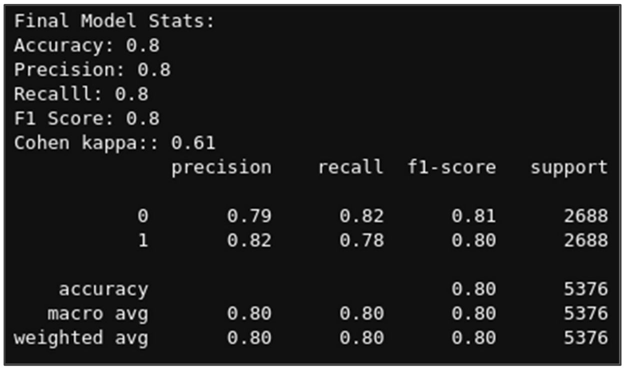
ROC/AUC Curves



Data Experiment: Random Forest – MPNet

Confusion Matrix and Summary Grid:



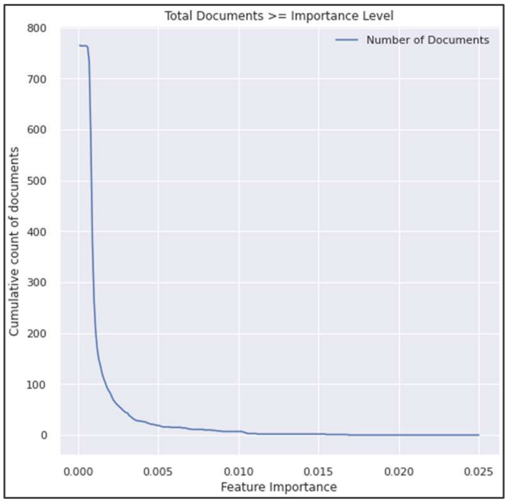


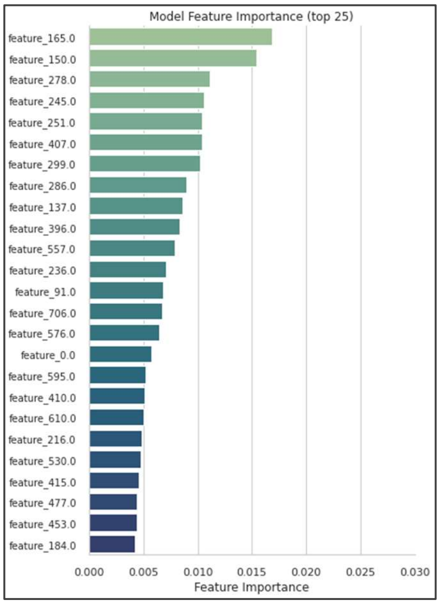
```
Final Model Stats:
Accuracy: 0.8
Precision: 0.8
Recalll: 0.8
F1 Score: 0.8
Cohen kappa:: 0.61
              precision    recall  f1-score   support

           0       0.79      0.82      0.81      2688
           1       0.82      0.78      0.80      2688

    accuracy                           0.80      5376
   macro avg       0.80      0.80      0.80      5376
weighted avg       0.80      0.80      0.80      5376
```

Feature Importance:

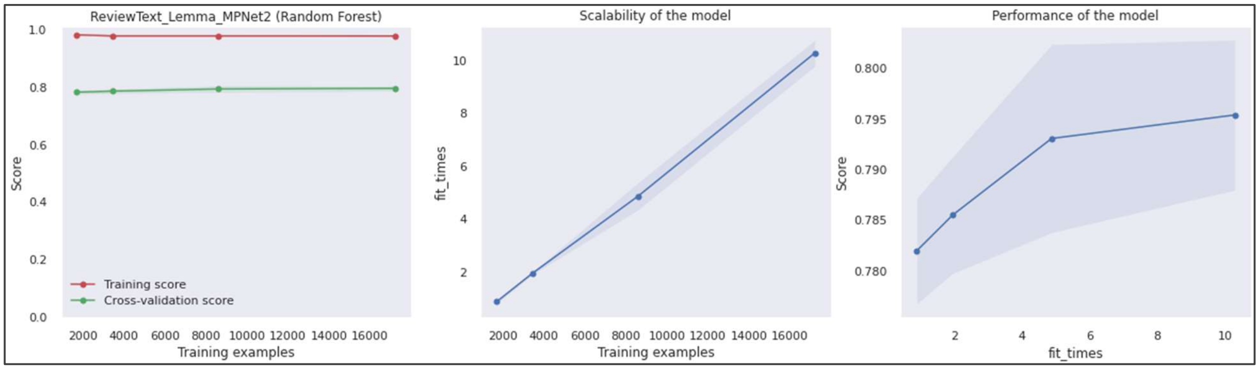
Graph displays number of features that have a feature importance of at least the value on the x-axis. For example there are 768 features in this experiment and all features have importance greater than 0. There are only about 50 features that have an importance of .004 or greater.

The Model features importance (right chart) shows the top 25 features in descending order of importance.

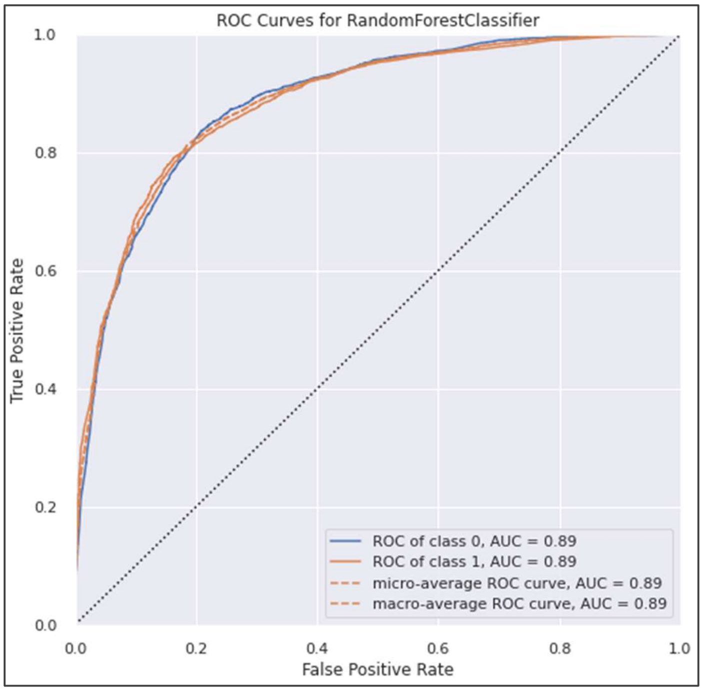




Learning Curves and Scalability

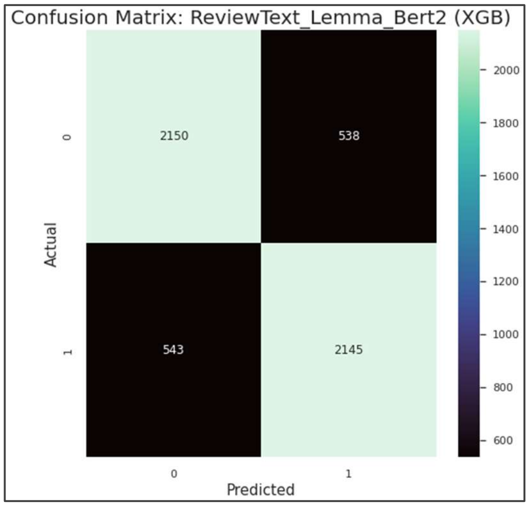


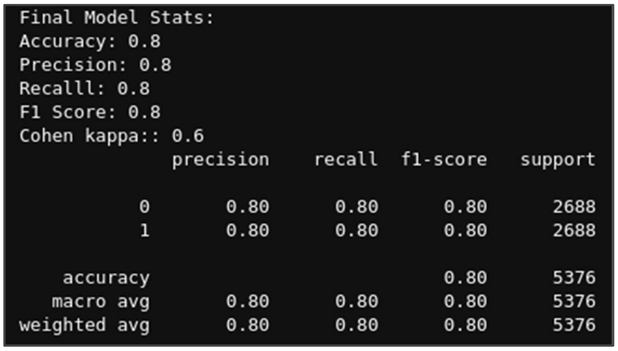
ROC/AUC Curves



Data Experiment: XGBoost – Bert Encoding

Confusion Matrix and Summary Grid:



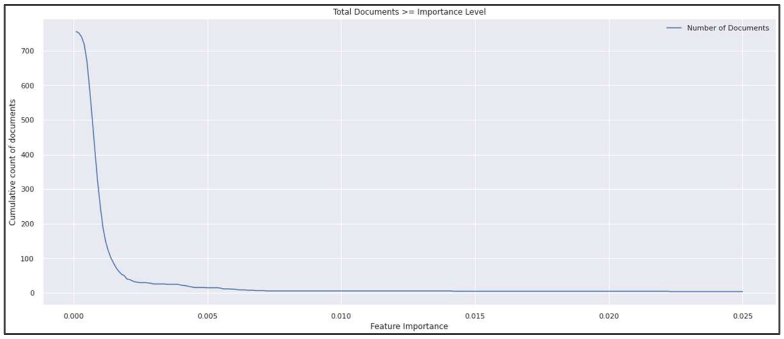


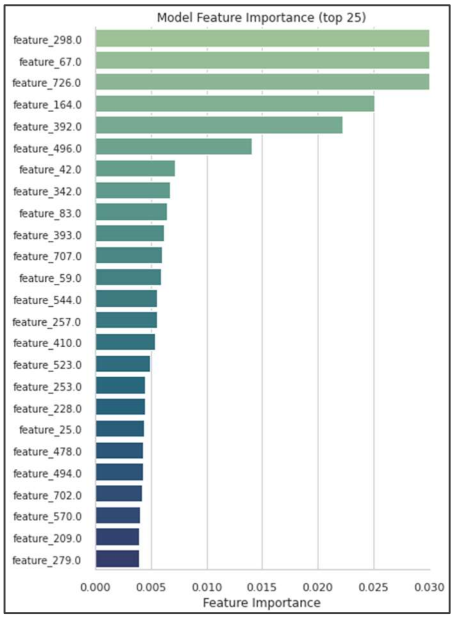
```
Final Model Stats:
Accuracy: 0.8
Precision: 0.8
Recalll: 0.8
F1 Score: 0.8
Cohen kappa:: 0.6
              precision    recall  f1-score   support

           0       0.80      0.80      0.80      2688
           1       0.80      0.80      0.80      2688

    accuracy                           0.80      5376
   macro avg       0.80      0.80      0.80      5376
weighted avg       0.80      0.80      0.80      5376
```

Feature Importance:

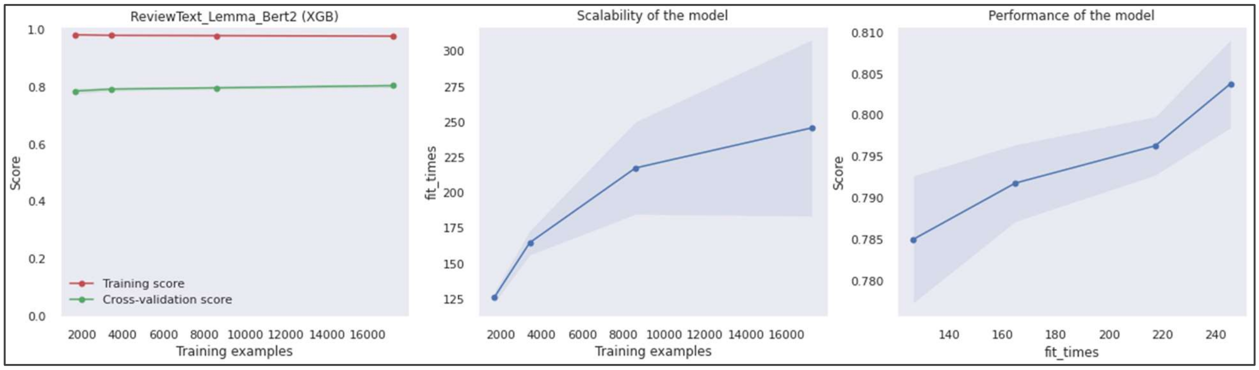
Graph displays number of features that have a feature importance of at least the value on the x-axis. For example there are 768 features in this experiment and all features have importance greater than 0. There are only about 5 features that have an importance of .005 or greater.

The Model features importance (right chart) shows the top 25 features in descending order of importance.

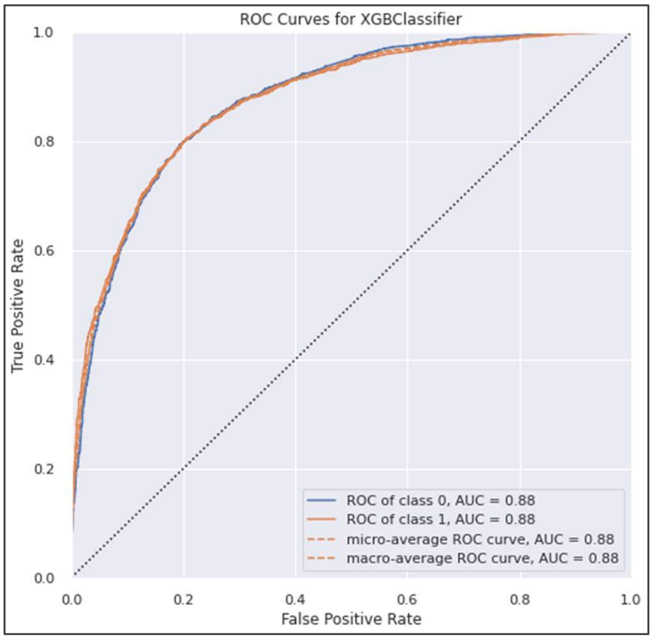




Learning Curves and Scalability

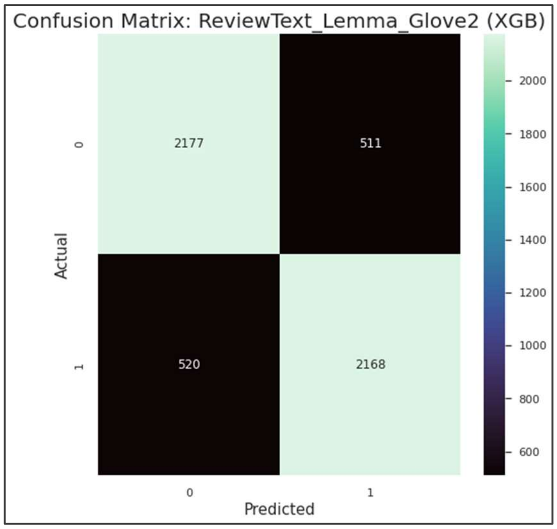


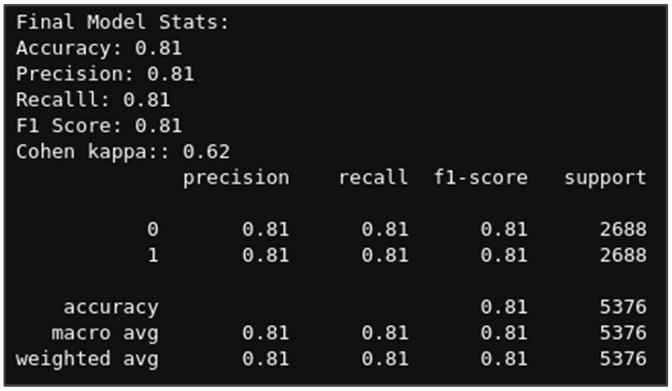
ROC/AUC Curves



Data Experiment: XGBoost – Glove Encoding

Confusion Matrix and Summary Grid:

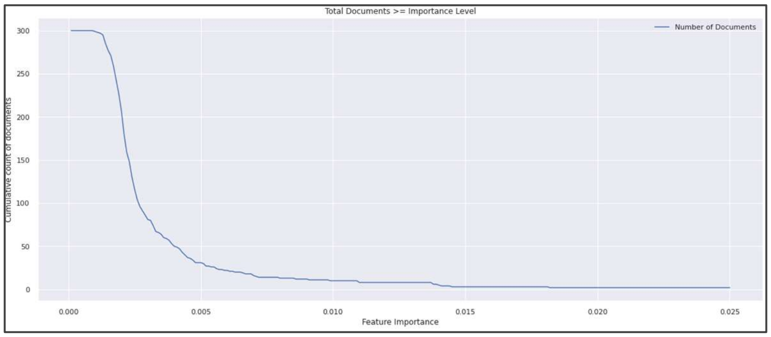


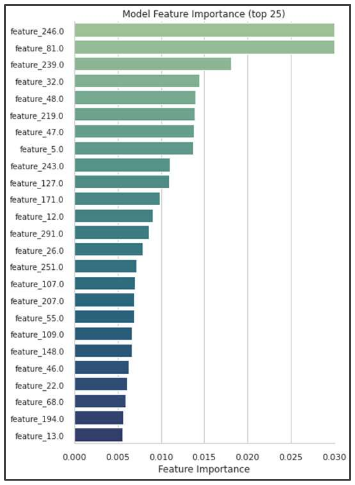


```
Final Model Stats:
Accuracy: 0.81
Precision: 0.81
Recalll: 0.81
F1 Score: 0.81
Cohen kappa:: 0.62
              precision    recall  f1-score   support

           0       0.81      0.81      0.81      2688
           1       0.81      0.81      0.81      2688

    accuracy                           0.81      5376
   macro avg       0.81      0.81      0.81      5376
weighted avg       0.81      0.81      0.81      5376
```

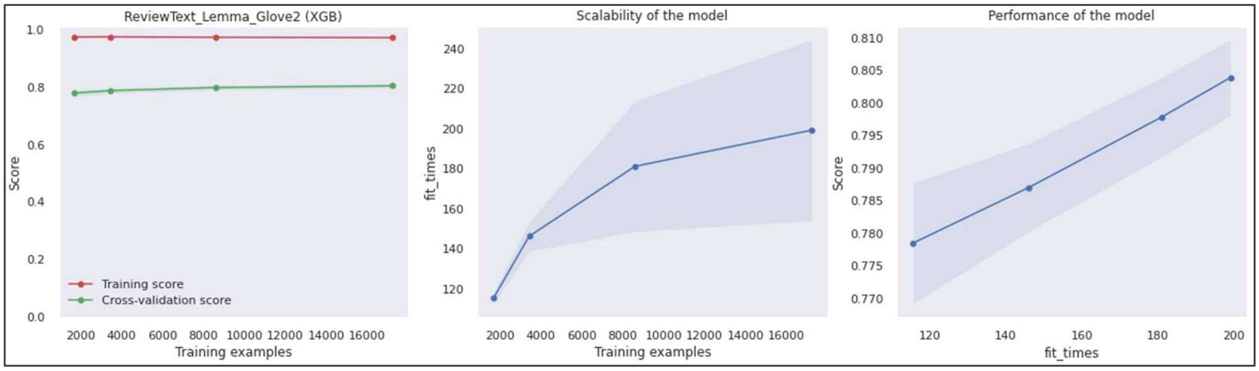
Feature Importance:

Graph displays number of features that have a feature importance of at least the value on the x-axis. For example there are 300 features in this experiment and all features have importance greater than 0.
There are only about 40 features that have an importance of .005 or greater.
The Model features importance (right chart) shows the top 25 features in descending order of importance.

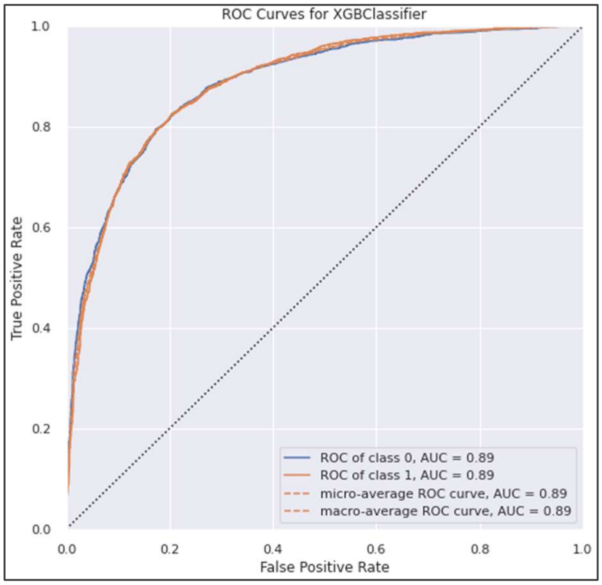




Learning Curves and Scalability

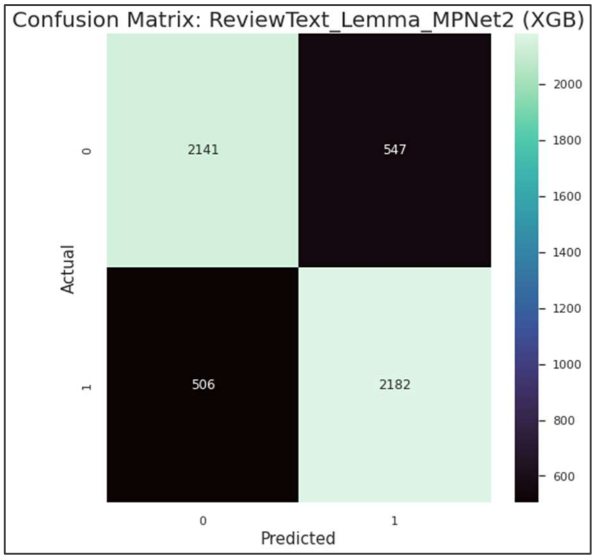


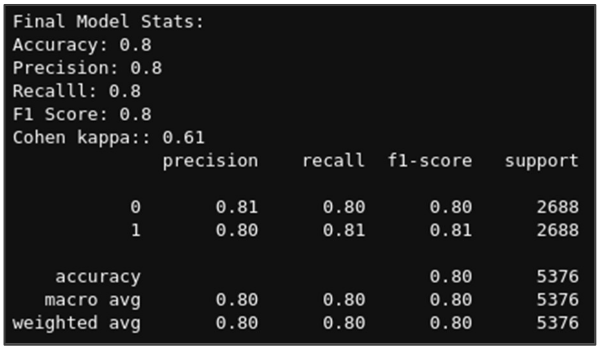
ROC/AUC Curves



Data Experiment: XGBoost – MPNet Encoding

Confusion Matrix and Summary Grid:



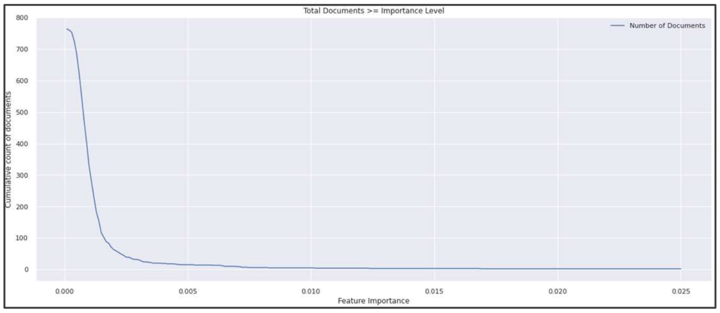


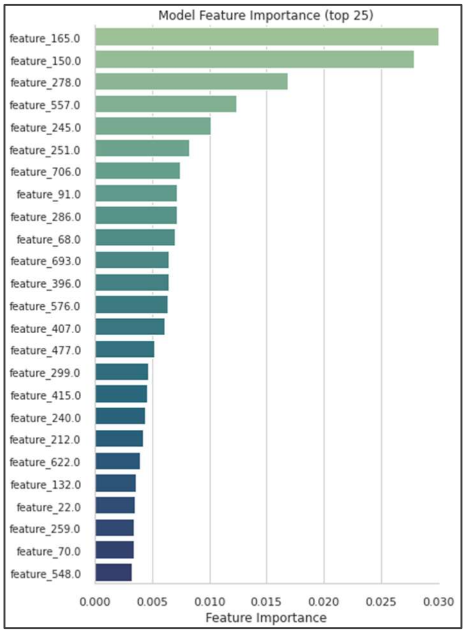
```
Final Model Stats:
Accuracy: 0.8
Precision: 0.8
Recalll: 0.8
F1 Score: 0.8
Cohen kappa:: 0.61
              precision    recall  f1-score   support

           0       0.81      0.80      0.80      2688
           1       0.80      0.81      0.81      2688

    accuracy                           0.80      5376
   macro avg       0.80      0.80      0.80      5376
weighted avg       0.80      0.80      0.80      5376
```

Feature Importance:

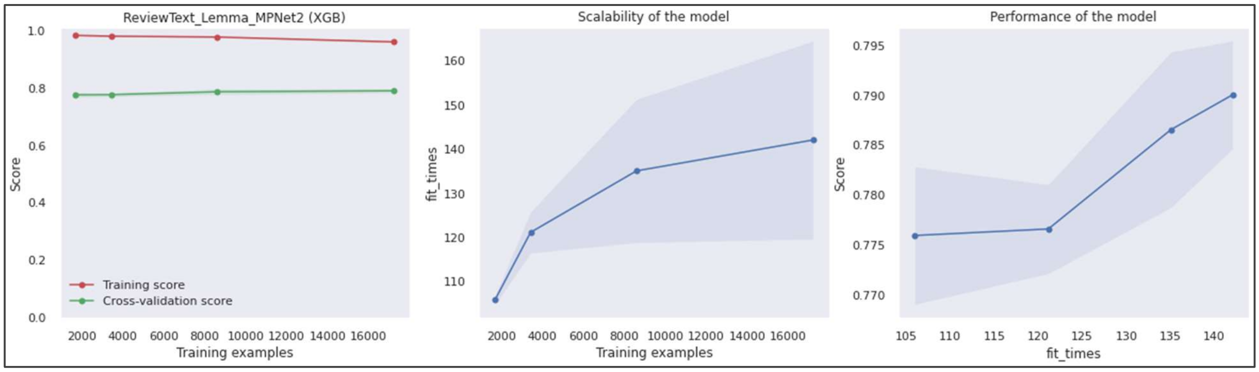
Graph displays number of features that have a feature importance of at least the value on the x-axis. For example there are 768 features in this experiment and all features have importance greater than 0. There are only about 50 features that have an importance of .004 or greater.

The Model features importance (right chart) shows the top 25 features in descending order of importance.

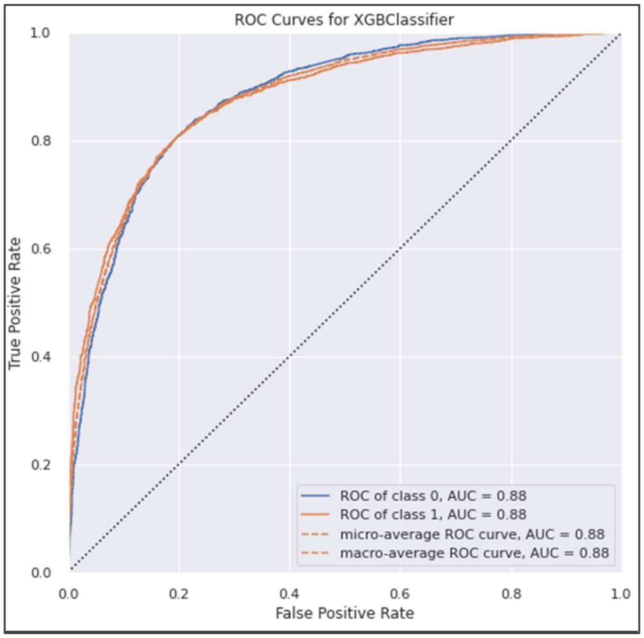




Learning Curves and Scalability

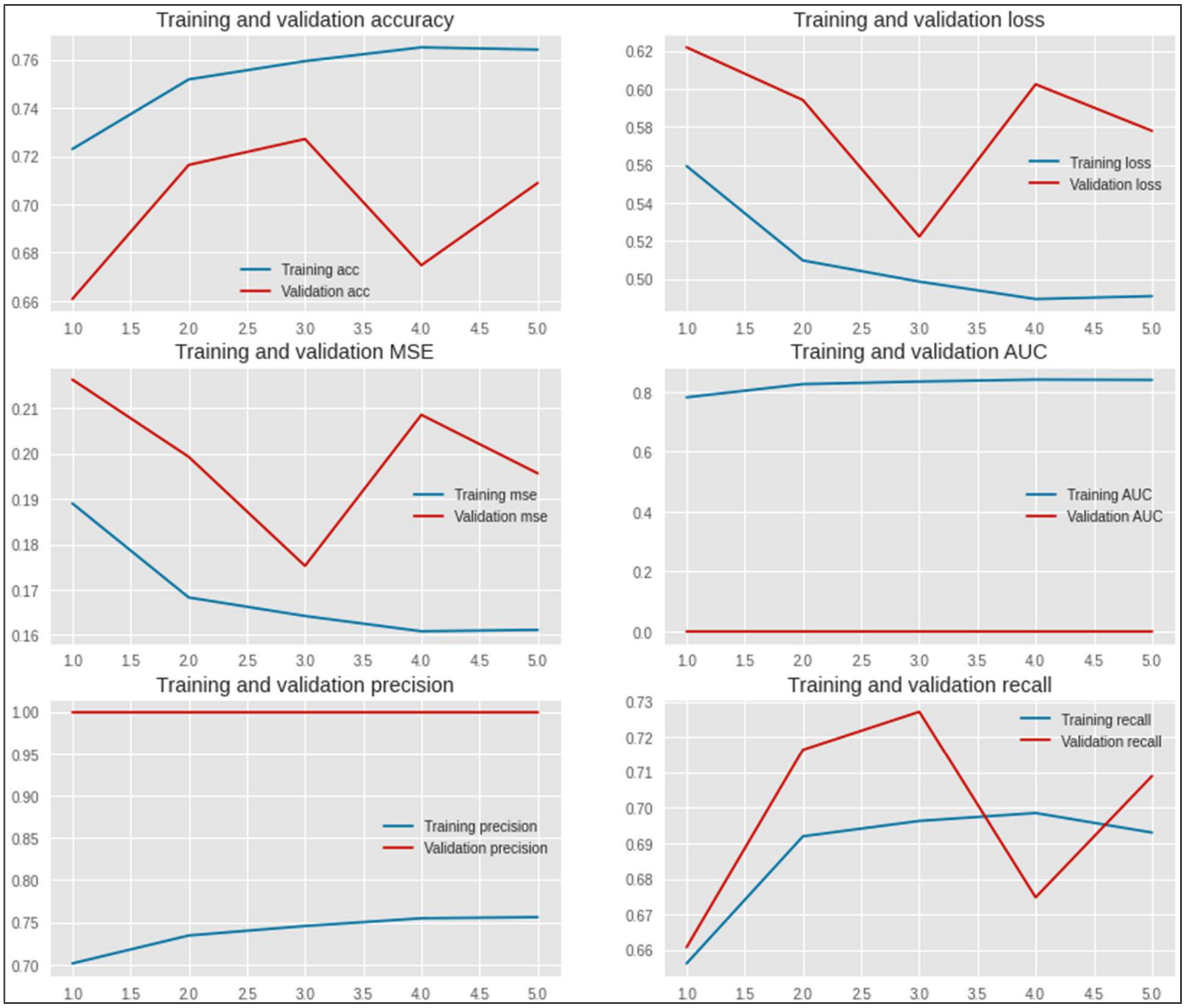


ROC/AUC Curves



Data Experiment: LSTM – Bert Encoding (All Features)

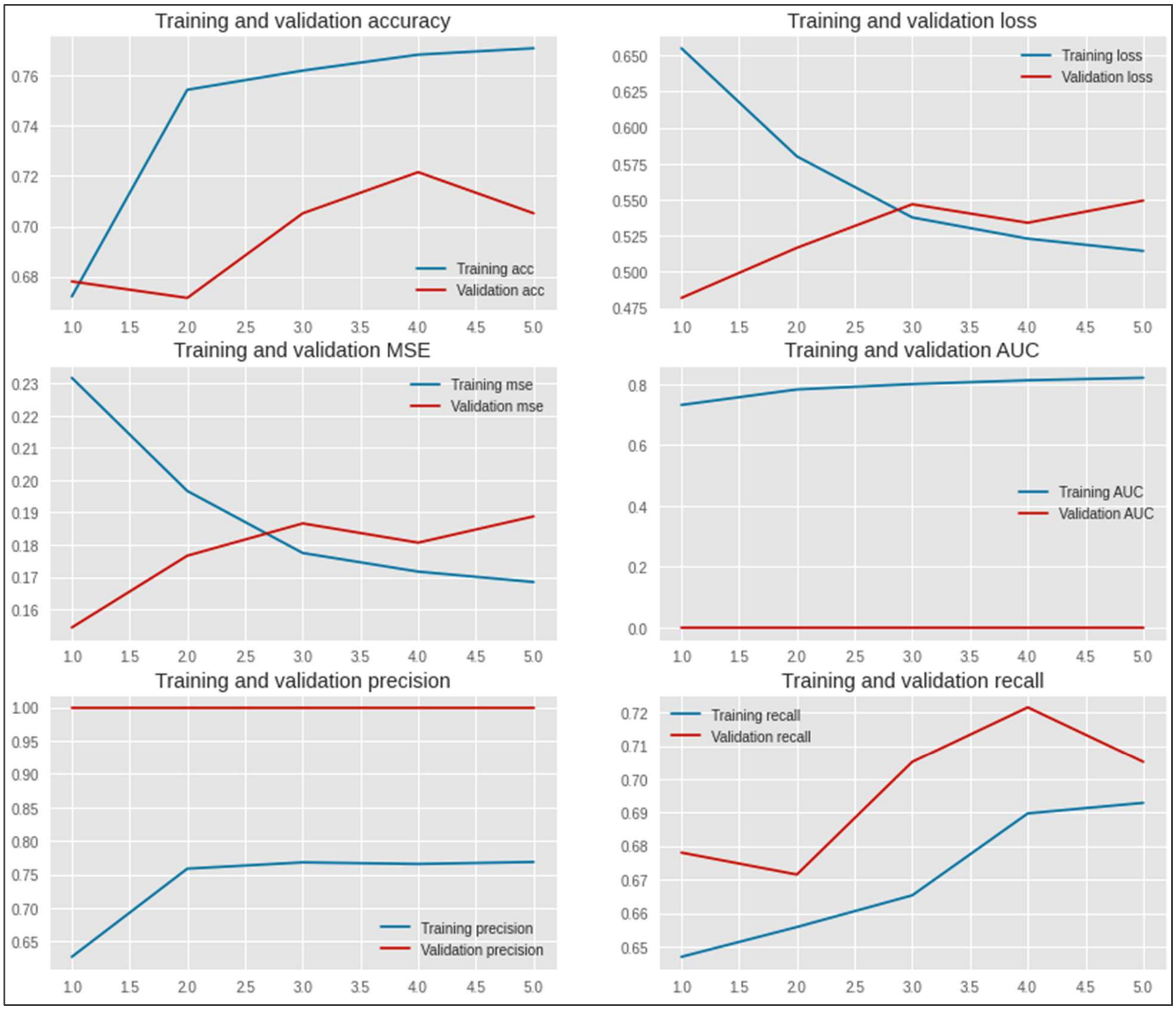
Note: Not all testing metrics for Tensorflow have been included as of yet. Below are the current training charts as a starting point



Data Experiment: LSTM – Bert Encoding (Important Features)

Note: Not all testing metrics for Tensorflow have been included as of yet. Below are the current training charts as a starting point.

The important features component of Tensorflow has not yet been implemented. Current “important features” have been used from the XGB Bert model. Results are better than the full features set.


Training and validation accuracy
Training acc
Validation acc
Training and validation loss
Training loss
Validation loss
Training and validation MSE
Training mse
Validation mse
Training and validation AUC
Training AUC
Validation AUC
Training and validation precision
Training precision
Validation precision
Training and validation recall
Training recall
Validation recall