Final Exam- CCFA-665

Aryan Kargwal, aryan.kargwal@mail.mcgill.ca

The following report has been created to explain the tasks performed under the Final Exam conducted as part of the course. For the exam, we have been presented with a corpus of Amazon reviews, on which we have the task of training different types of models using the techniques we have learned during the course.

The code for the following experiment has been attached as a Python Notebook and can also be found here.

As part of the common steps taken for the 4 models we have trained, here goes:

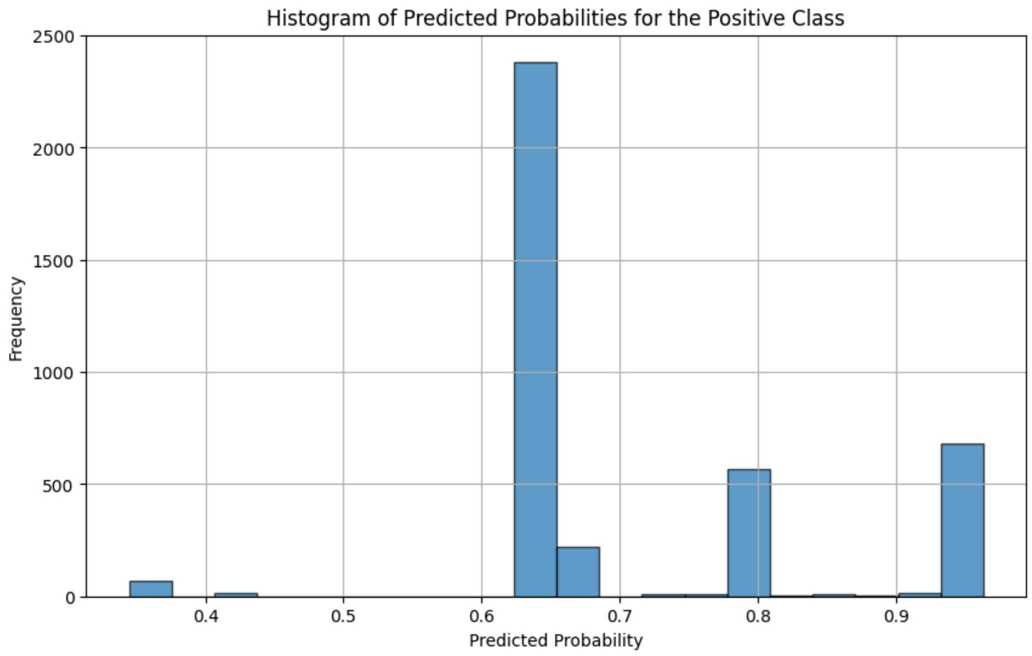
1. All the models have been trained using **T4 12GB VRAM GPU** Available on Google Colab.
2. The models are presented with unclean data cleaned using **NLTK** and other supported libraries.
3. The models, once trained are evaluated with their F1 Score, where **0 is a negative** reaction and **1 is a positive** reaction.
4. The models are then used to plot a histogram where the probability distribution of individual models is plotted.
5. Lastly, the best-optimized hyperparameters achieved using **Random Search** are also presented for each model.

1.1 XGBoost using Bag Of Words

a. F1 Score

Class	Precision	Recall	F1-Score
0	0.88	0.08	0.15
1	0.77	1.00	0.87

b. Predicted Probability Distribution



c. Optimized Hyper-Parameters using Random Search
   subsample: 0.9
   n_estimators: 200
   max_depth: 3
   learning_rate: 0.01
   colsample_bytree: 0.9
d. Accuracy: 0.78

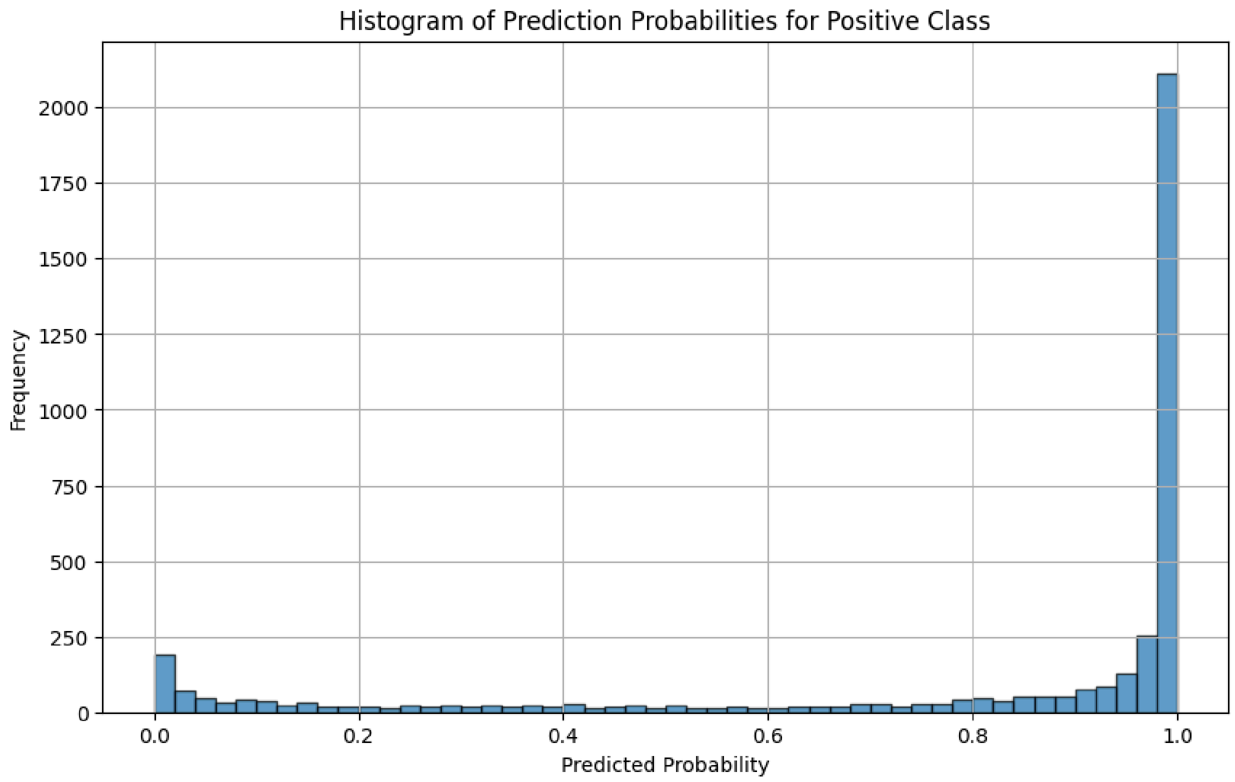
1.2 XGBoost with Word2Vec

Here we noticed a bump in the precision and recall for the false category, and overall improving the positives as well. As plotted in the graph, the overall prediction distribution for the positive class shows a higher frequency in the true range.

a. F1 Score

Class	Precision	Recall	F1-Score
0	0.78	0.65	0.71
1	0.89	0.94	0.92

b. Predicted Probability Distribution



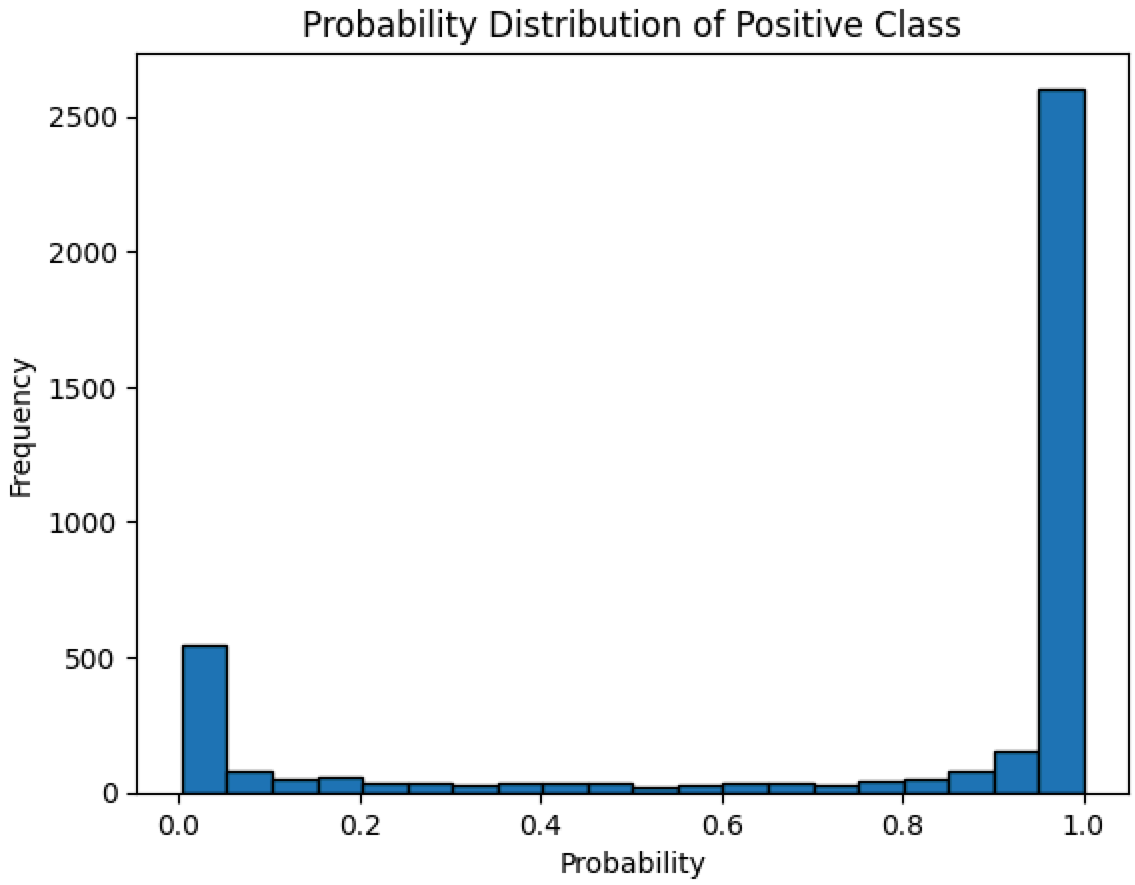
c. Optimized Hyper-Parameters using Random Search
   Optimized Hyperparameters:
   subsample: 0.9
   n_estimators: 300
   max_depth: 5
   learning_rate: 0.2
d. Accuracy: 0.87

1.3 Neural Network with Bag of Words

a. F1 Score

Class	Precision	Recall	F1-Score
0.0	0.77	0.75	0.76
1.0	0.92	0.93	0.93

b. Predicted Probability Distribution



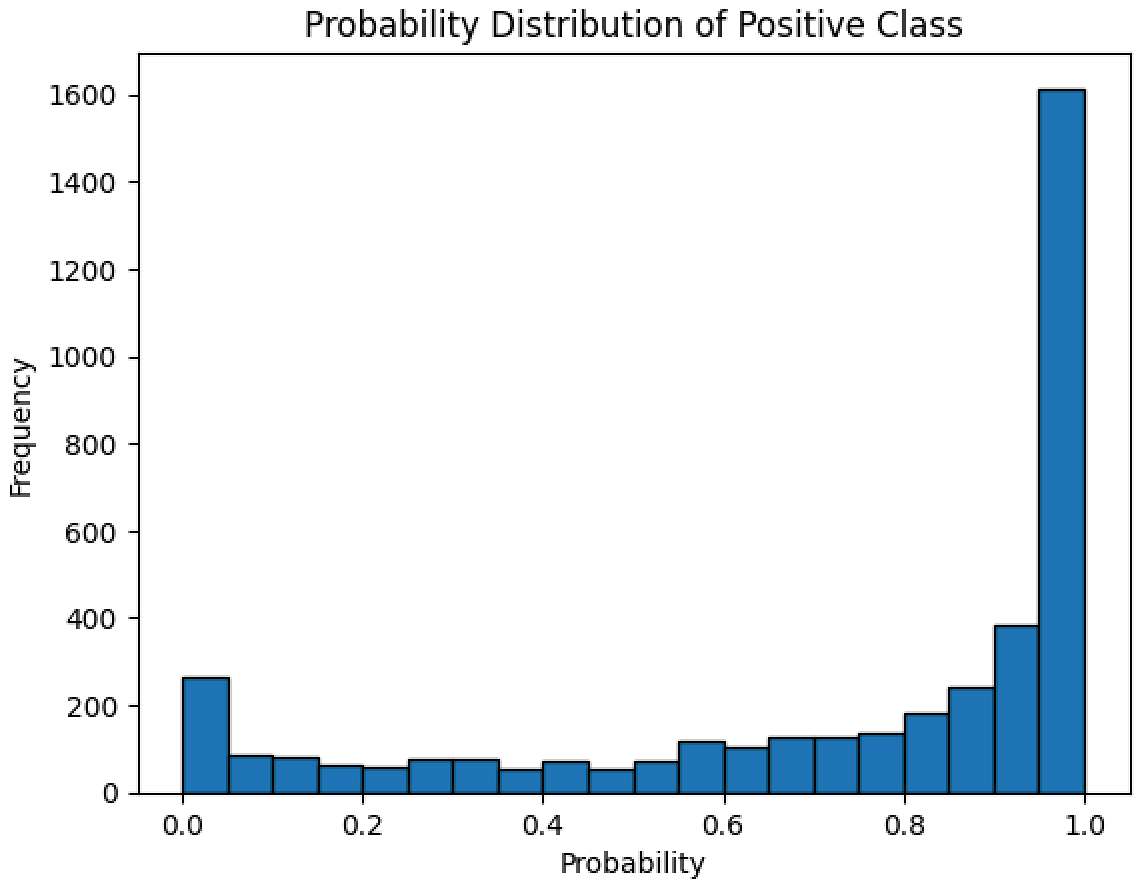
c. Optimized Hyper-Parameters using Random Search
'hidden_size1': 512
hidden_size2': 64
'hidden_size3': 64
'learning_rate': 0.0001
‘batch_size': 128
d. Accuracy: 0.89

1.4 Neural Network with Word2Vec

a. F1 Score

Class	**Precision**	**Recall**	**F1-Score**
0.0	0.76	0.71	0.73
1.0	0.91	0.93	0.92

b. Predicted Probability Distribution



c. Optimized Hyper-Parameters using Random Search
   'hidden_size1': 128
   'hidden_size2': 256
   'hidden_size3': 64
   'learning_rate': 0.01
   'batch_size': 32
d. Accuracy: 0.87

Findings

Going through the results produced by the four models a few observations were made:

1. The Neural network with 3 hidden layers and optimizing dataset using the Bag of Words method gives us the best **Accuracy of 0.89** an **F1 score of 0.93**.
2. The Boosted Tree using Word2Vec from a pretrained model gave us better performance against its counterpart, with an **Accuracy of 0.87** and **F1 score of 0.92**.