CS 182 NLP Project - Yelp Rating Prediction

Yao Liang, Duowei Pan, Tianyi Guan

Introduction:

The objective of our project is to train a text classifier model that can predict the rating of the users given the texts of their reviews. Based on the text reviews and ratings dataset given by Yelp, we are going to train a model that is robust to perturbations and works in production systems. The metric that we will use to evaluate the effectiveness of our model is mean absolute error (MAE), which is the average distance between the predicted ratings and the true values. We will also use accuracy, which is the number of right predictions/ total number of reviews, to evaluate our result. This problem is important and our work will have impacts that both help the Yelp rating system and for later NLP studies. To be more specific, first of all, the model can help filtering out useless reviews for Yelp users. The model may consider reviews to be "not so useful" if it can't help distinguish its rating or has similar scores for every rating. Then those reviews may not give much information to users finding nice restaurants, and Yelp can filter it out for the users. In addition to that, if the robustness of this model is sufficient enough, Yelp can use this model to predict the ratings of restaurants based on more resources other than just the users' reviews like online articles or journal critical essays. Lastly, the Yelp review data set is a very large data set with 6 million text reviews, and studying the set may provide a chance to modify or improve the existing NLP process. During the process of designing and improving the robustness of our model, possible new changes can happen at any steps and may be useful for relevant studies in NLP.

Literature Survey/Related Work:

We find papers "Review Rating Prediction: A Combined Approach", "Character-level Convolutional Networks for Text Classification", "Yelp Review Rating Prediction: Machine Learning and Deep Learning Models", "Identifying Restaurant Features via Sentiment Analysis on Yelp Reviews", "Recurrent Neural Network for Text Classification with Multi-Task Learning" to be relevant when exploring general ideas for our work. The first paper proposed the idea of performing a TF-IDF term weighting and vectorizing the text and will then use KNN model to do the classification. The second paper explored using character-level convolutional networks for text classification. The third paper used four machine learning models, containing Naive Bayes, Logistic Regression, Random Forest, and Linear Support Vector Machine, and four transformer-based models, including BERT, DistilBERT, RoBERTa, and XLNet to do the rating prediction, while the final paper use a support vector machine (SVM) model to do the Sentiment Analysis for each review, and rating can then be predicted accordingly. The fifth paper integrates RNN into the multitask learning framework to jointly learn across multiple related tasks. Inspired by paper five, we will use the basic RNN model as our baseline model. However, we will do it without the multitask learning framework since we only have one task here. Different from paper one and three, four, we will focus on utilizing neural network language models to complete the

work. Inspired by paper two, we will also explore CNN, but we will combine it with BiLSTM, GRU for a possible better result. We will also try using BiLSTM, GRU, CNN to perform a regression instead of classification for this problem.

Background:

Since the Yelp dataset is imbalanced where it has more data with 1,4 and 5 stars than that of 2 and 3 stars, we have added data augmentation where we have trained a LSTM language model to study the reviews of 2 and 3 stars, and make it produce new sentences with certain length. Those new sentences will then be used as training data for our final model so it can better learn the features of reviews with 2 or 3 stars. However, after we tried to do that and add new sentences for training, the result was not improved. Our guess for the reason behind that is that the reviews are unlikely to be 2 or 3 stars, so adding new data may result in providing misleading information. So we eventually delete that part for data preprocess.

Instead for data preprocess, we tackled the problem of the variance of the review sentences length. There are sentences with length smaller than 20, and they can be difficult to learn, so we repeat the content of those sentences to form new reviews with longer length for the model to learn better.

We build our work on CNN and RNN-based models. We start with RNN as our baseline model, then we move to BiLSTM, GRU and CNN. RNN-based models treat text data as sequences of tokens and can capture the dependencies well between those sequences of words. LSTM fixes the short term memory issues of vanilla RNN, and thus works better with the reviews with long length, while GRUs is more simple and can be trained faster compared to LSTM. CNN, on the other hand, is trained to recognize patterns of the text data, thus it can learn key phrases better and help extract information better.

Methods/Approach:

Our baseline model is a simple RNN model with Sparse Categorical Cross Entropy for loss function. We choose to start with RNN because it can capture the local dependencies well for text sequence by storing weights of words based on their location and importance, and thus can do classification with those informations efficiently. Here, we choose Categorical Cross-Entropy loss, which is Softmax activation added to the Cross-Entropy loss, for loss function because it is used for multi-class classification. Other than that, we have tried two classification models and two regression models. The first classification model we have is one CNN layer and one BiLSTM layer. The second classification model we have is a combination of one CNN layer, one BiLSTM layer and one GRU layer. For both of these two classification models, we have one Spatial Dropout before the CNN layer to promote independence between feature maps, dropout layers in between and Sparse Categorical Cross Entropy for loss. We have also tried to make the stars prediction to be a regression problem instead of a classification problem. The first regression model we have is using the BiLSTM model to do linear regression, while the second model we have combined CNN and BiGRU to do linear regression. Both two

models have Mean Absolute Error for loss, and have dropout layers in between. When testing on the test data that we splitted from the training data, the two classification models work better than the baseline model on both the accuracy and the MAE, while the second regression model has similar accuracy as the baseline model, but lower MAE. The first regression model however performs worse than the baseline model on both the accuracy and the MAE on the test data.

For graph for models, see appendix. For better visualization, we also draw a heap map for the confusion matrix of the performance for the 4 models mentioned above on the test set. The CM heap map is in the appendix below. (CM: each column represents a predicted label and each row means the true label, thus, diagonal entries are numbers of True Positives and all other entries are numbers of False Negatives.)

Results:

Our final model will be the ensemble of two regression models and two classification models that we mentioned above with ratio 0.2:0.2:0.3:0.3. Although the performance of both two regression models seems to be worse than the two classification models in the test data as the results shown below, surprisingly, they help the model with challenge data 5 and 8. The details are shown below.

For the test data that we split by ourselves from the given training data, the accuracy for the baseline model is 0.76, and the MAE is 0.34. The accuracy for the first classification model is 0.79 and the MAE is 0.27. The accuracy for the second classification model is 0.78 and the MAE is 0.29. The accuracy for the first regression model is 0.71 and the MAE is 0.43, The accuracy for the second regression model is 0.76 and the MAE is 0.30.

Model:	Baseline:	Class 1	Class 2	Regress 1	Regress 2
Accuracy:	0.758	0.789	0.780	0.712	0.758
MAE:	0.343	0.270	0.292	0.430	0.304

For challenge data 5, the accuracy for the baseline model is 0.25, and the MAE is 0.95. The accuracy for the first classification model is 0.29 and the MAE is 0.91. The accuracy for the second classification model is 0.35 and the MAE is 0.81. Although the two regression models all performed worse on both accuracy and MAE compared to the two classification models individually, the ensemble of the regression models with the two classification models, surprisingly, improves the performance of the model on challenge data 5. Here, the ensemble of the first regression model and the two classification models with ratio 0.4:0.3:0.3 results in accuracy of 0.36 and MAE of 0.84; The ensemble of the second regression model and the two classification models with ratio 0.4:0.3:0.3 results in accuracy of 0.35 and MAE of 0.78; Our final model, the ensemble of two regression models and two classification models together with ratio 0.2:0.2:0.3:0.3 gives us the best performance with accuracy of 0.406 and MAE of 0.788.

Model:	Baseline	Class 1	Class 2	Final Model
Accuracy:	0.25	0.29	0.35	0.41
MAE:	0.95	0.91	0.81	0.79

For challenge data 8, the accuracy for the baseline model is 0.45, and the MAE is 1.23. The accuracy for the first classification model is 0.63 and the MAE is 0.56. The accuracy for the second classification model is 0.63 and the MAE is 0.58. The ensemble of the first regression model and the two classification models with same ratio as above results in accuracy of 0.61 and MAE of 0.56; The ensemble of the second regression model and the two classification models with same ratio as above results in accuracy of 0.63 and MAE of 0.53; Our final model results in accuracy of 0.62 and MAE of 0.54.

Model:	Baseline	Class1	Class 2	Final Model
Accuracy:	0.45	0.63	0.63	0.62
MAE:	1.23	0.56	0.58	0.54

During this process, we found that although the two classification models performed well in the test set that we splitted from the training data, they failed to perform well for challenge data 5. We think it happened because data 5 is rated with all 2 stars, and the 2 stars data was in a small proportion in the training data set compared to other stars, then the network failed to learn its features well.

Conclusion/Lesson Learned:

For this project, we had one RNN model for baseline, and explored four different models with two classification models and two regression models. We combined layers of BiLSTM, CNN and GRU for those models. Our final model is the ensemble of those four models with ratio 0.2:0.2:0.3:0.3. The two classification models work better than the baseline model by improving the accuracy of around 2% and decreasing the MAE by around 0.06 at the test data. The two regression models performed worse than the two classification models on test data. However, the two classification models performed poorly at challenge data 5, while the ensembles of them with the regression models, which is our final model, performed well by helping improve the accuracy of around 5% and decreasing the MAE by around 0.1 compared to the classification models, and improve the accuracy of around 15% and decreasing the MAE by around 0.2 compared to the baseline model. Our final model also improved the accuracy of around 15% and decreased the MAE by around 0.7 compared to the baseline model.

The surprise here is although the two regression models all performed worse on both accuracy and MAE compared to the two classification models individually on the test data,when we combine the regression models with the classification models, the results get better on some

dataset like challenge dataset 5. We think the reason behind it is when consider this problem as a regression problem, the model provide a continuous range for stars prediction instead of give a discontinuous result as the classification model, for example, for a two star review, the classification model will predicted it as 3 or 4, while the regression may predict it as 1.8 or 2.3, and that will help smooth the results to get close to the real one. Here, we learned that solving a question from a different approach, like solving this classification question from a regression approach may yield surprising results, and models that seem to perform worse are not useless. They may take care of other perspectives of the questions, and may yield better performance when combined with other models.

Team Contributions:

Yao Liang (50%): Mainly developed the overall design of the project, in charge of the notebooks and code with data processing, model building and evaluation of various models.

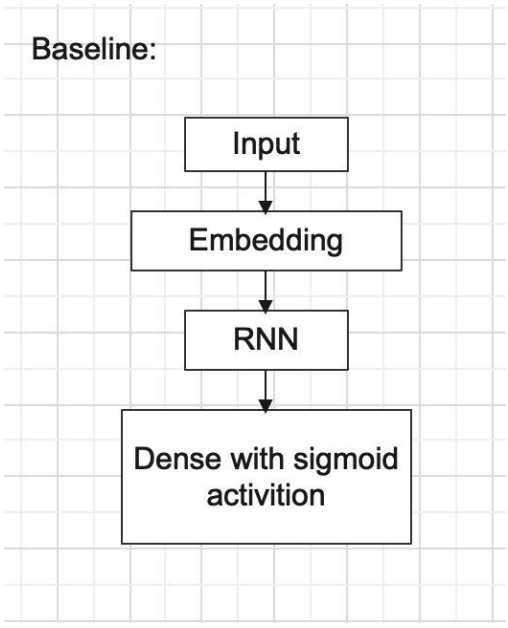
Duowei Pan (35%): Developed and organized the workflow of the project. Worked on the research and the final report.

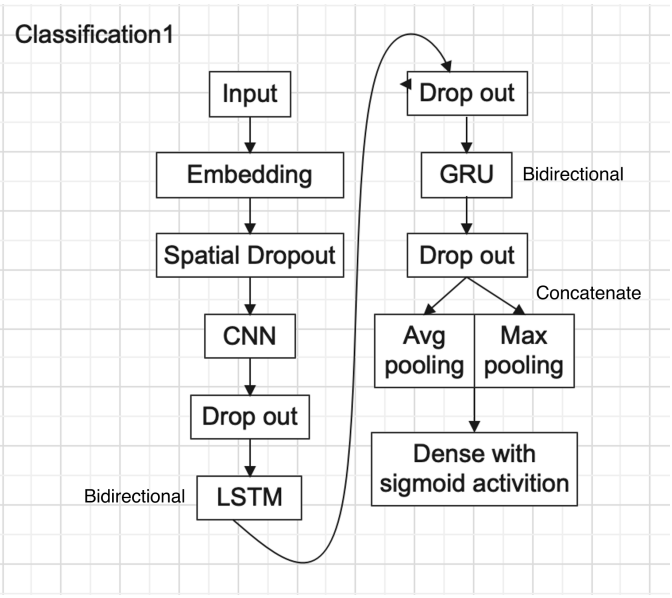
Tianyi Guan (15%): Worked on the visualization for the model and results, and code organization.

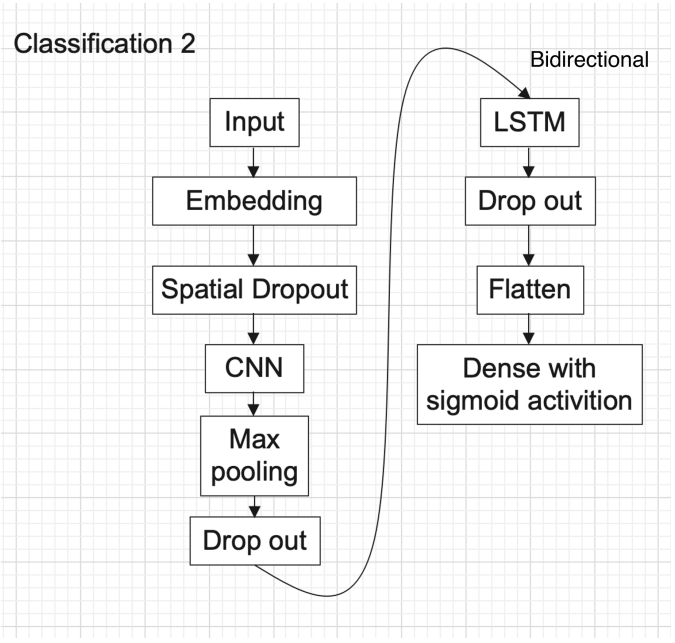
Reference:

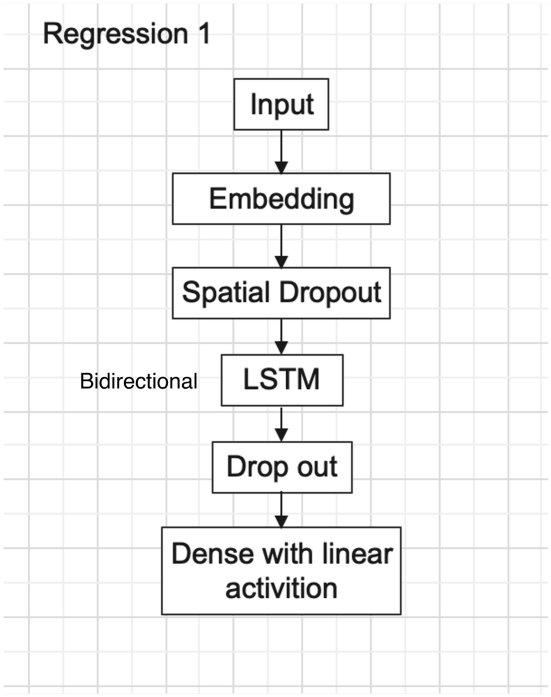
1. "Character-level Convolutional Networks for Text Classification", https://arxiv.org/pdf/1509.01626.pdf
2. Yelp Review Rating Prediction: Machine Learning and Deep Learning Models, https://arxiv.org/abs/2012.06690
3. Identifying Restaurant Features via Sentiment Analysis on Yelp Reviews, https://arxiv.org/pdf/1709.08698.pdf
4. Review Rating Prediction: A Combined Approach, https://towardsdatascience.com/review-rating-prediction-a-combined-approach-538c617c495c
5. Recurrent Neural Network for Text Classification with Multi-Task Learning https://www.ijcai.org/Proceedings/16/Papers/408.pdf

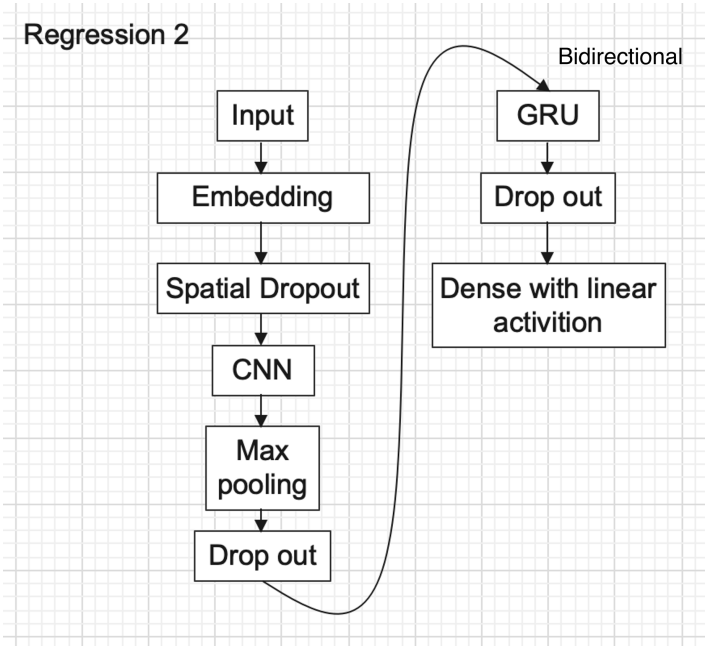
Appendix:
Graph of the baseline model, two classification models, two regression models, and the final ensemble model.

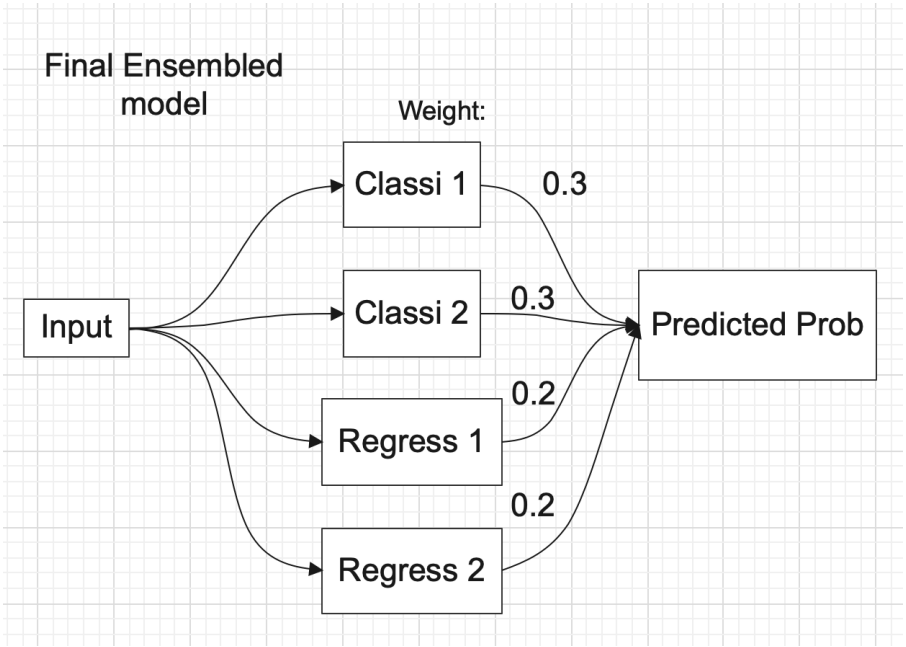




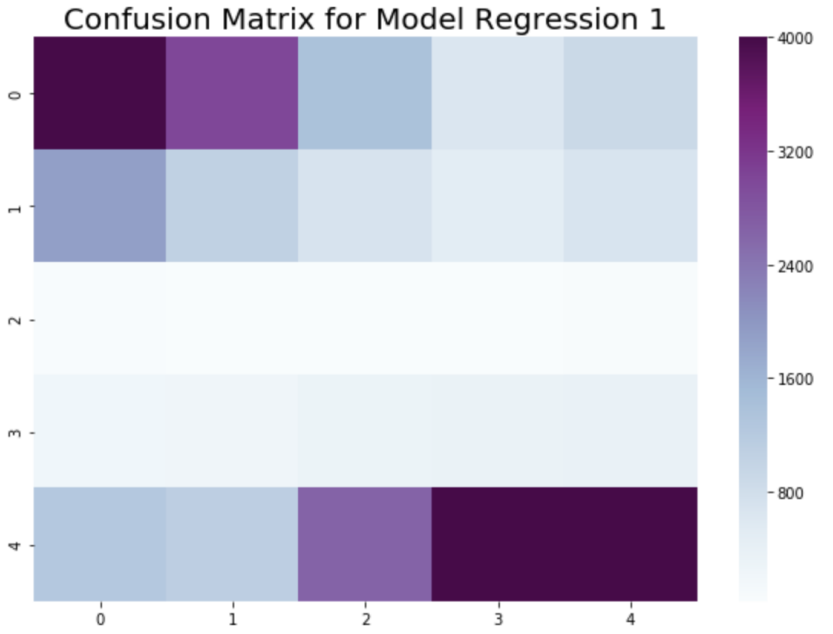


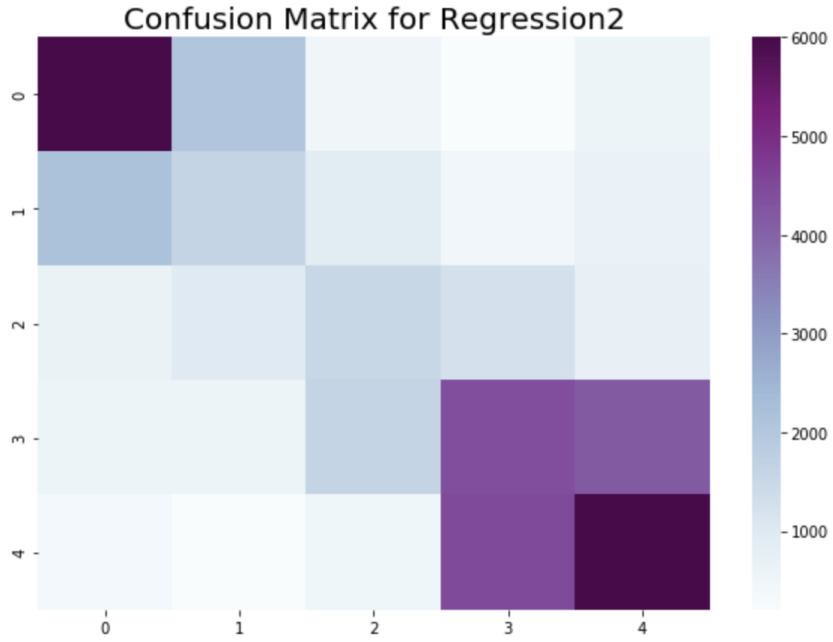

Regression 1
Input
Embedding
Spatial Dropout
Bidirectional
LSTM
Drop out
Dense with linear activition

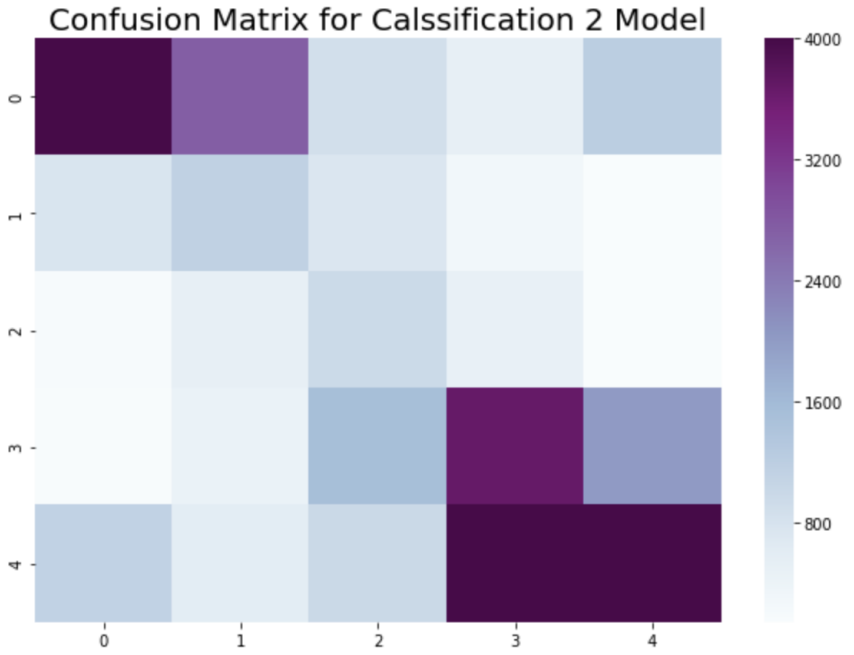

Regression 2
Input
Embedding
Spatial Dropout
CNN
Max pooling
Drop out
Bidirectional
GRU
Drop out
Dense with linear activition

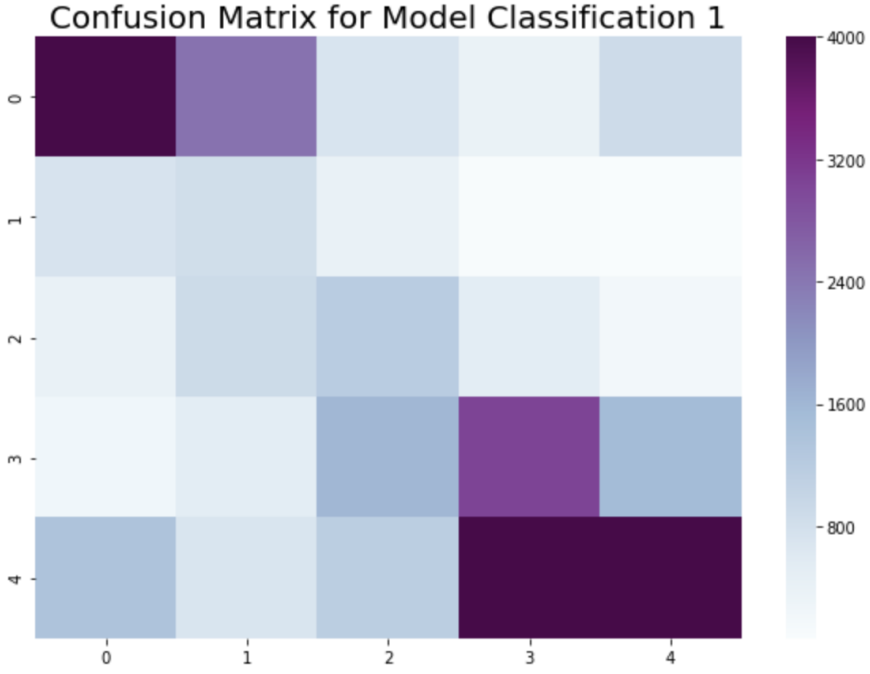

Final Ensembled model
Weight:
Input
Classi 1
0.3
Classi 2
0.3
Regress 1
0.2
Regress 2
0.2
Predicted Prob

Heap map for the two classification models, two regression models on test data
(Index 0-4 responds to stars 1-5)






Confusion Matrix for Calssification 2 Model
0
1
2
3
4
0
1
2
3
4
4000
3200
2400
1600
800


Confusion Matrix for Model Classification 1
0
1
2
3
4
0
1
2
3
4
4000
3200
2400
1600
800