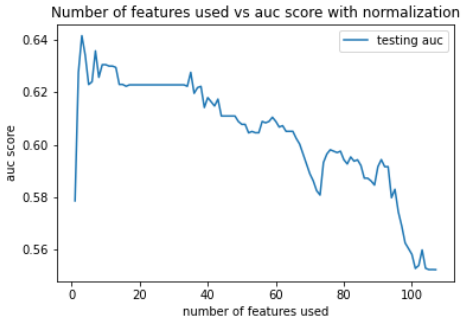
KNN model

Method and Analysis: For the KNN model, the data were separated by a 75:25 ratio, where 75% of the data was used as training data, and 25% of the data was used as testing data. With the raw data and unadjusted learner, the optimal K = 16 was selected. In addition, all features were normalized using the infinity norm.
The unadjusted model with all features selected is inefficient (the prediction of 2000 entries will take 10-15 seconds) but inaccurate (AUC score 0.55520 on Kaggle site). Feature selection was necessary. It is also impractical to select every combination of feature pairs and test the AUC score (around 1.62259*10^32 combination). Therefore, the feature set was selected by the following algorithms.
Step1: The program first starts with an empty set of features. All features were trained individually, and the AUC score was calculated based on testing data. The feature that has the best AUC score will be added to the selected feature set, and the AUC score will be recorded.
Step2: Then, the rest of the feature would be trained along with the selected feature, the best feature set would be selected based on their AUC score, and the selected feature set would be updated accordingly.

After 107 iterations, the number of features used versus the AUC score was plotted. From the plot, we can see that the AUC score first increases as the number of feature increase (model became complex), but it dropped rapidly as the number of features became too much (definitely overfitting here). Since the AUC score is stable when the number of features used is around 30, only 30 selected features were provided to the final learner.



Result and Limitation: The final KNN learner used only 30 features, with 75% of the training data. The final AUC score is 0.66672, roughly a 20% improvement compared to the unadjusted model. However, since there are too many features in the dataset, selecting the optimal feature set is challenging. Furthermore, the prediction cost is high. The average predicting time is around 20 seconds for a dataset with 7000 entries. Therefore, KNN learners might not be the optimal model for this dataset.