6010 AI in Healthcare

Predicting Length of Stay in a Hospital

Dezmon Patten, Rosh Chan

1. Introduction

1.1 Problem Statement

The goal of this project is to develop a predictive model that would be used to predict patients who will likely stay seven or more days in the hospital. Accurate predictive models help healthcare providers in resource allocation and patient management.

1.2 Dataset

Original Data set contained 100,000 patients with 28 features.The dataset used in this analysis contains information about patients, including medical history, number of previous admissions as well as various diagnostic codes and lab results.

2. Data Preprocessing

2.1 Data Cleaning

The dataset was free of missing values.

2.2 Data Transformation

After checking for missing values we then examined our variables to identify variables that would require some sort of transformation for our model. We decided to use an encoder to encode the gender variable along with standardizing our continuous laboratory measurements.

3. Feature Engineering

In this set of our model development we decided to create a new variable called “number_of_issues” that would represent the total number of medical conditions . By doing this we were able to reduce the number of features for our regressor by 11 features. In this step we also removed unnecessary features that were not important when it came to predicting the length of stay. Also in this step we create a binary variable to represent length of stay greater than 7 as this is what we will be using our remaining features to predict.

	Number of Samples	Number of Features	Description
Training Set	80,000	12	This dataset was used for training and model development
Testing Set	20,000	1	This dataset is used to assess the models performance after development.

Table 1

4. Model Building

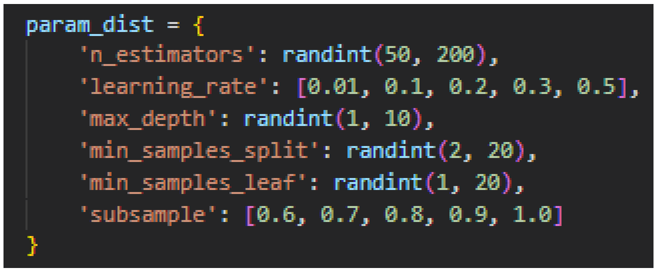
In this section, we run multiple ML models and we split and we use an 80% training and 20% testing split for our dataset. We considered adding a hold Each model is 5-fold cross-validated and a random search is run through each model type to explore and find the best hyperparameters.

4.1 Random Forest

For the random forest, we explored the possible hyperparameters:

4.2 Gradient Boosting Regressor

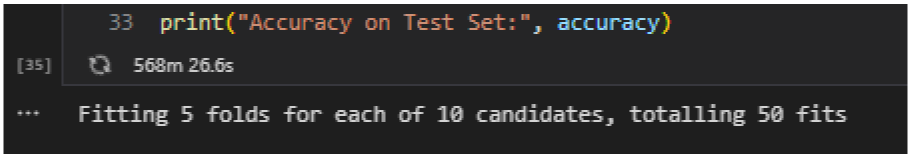
For the Gradient boosting, we explored the possible hyperparameters:



```
param_dist = {
    'n_estimators': randint(50, 200),
    'learning_rate': [0.01, 0.1, 0.2, 0.3, 0.5],
    'max_depth': randint(1, 10),
    'min_samples_split': randint(2, 20),
    'min_samples_leaf': randint(1, 20),
    'subsample': [0.6, 0.7, 0.8, 0.9, 1.0]
}
```

4.2 SVM

We Attempted to train an SVM but it took too long and we stopped the model training at around 9 hours of run time.

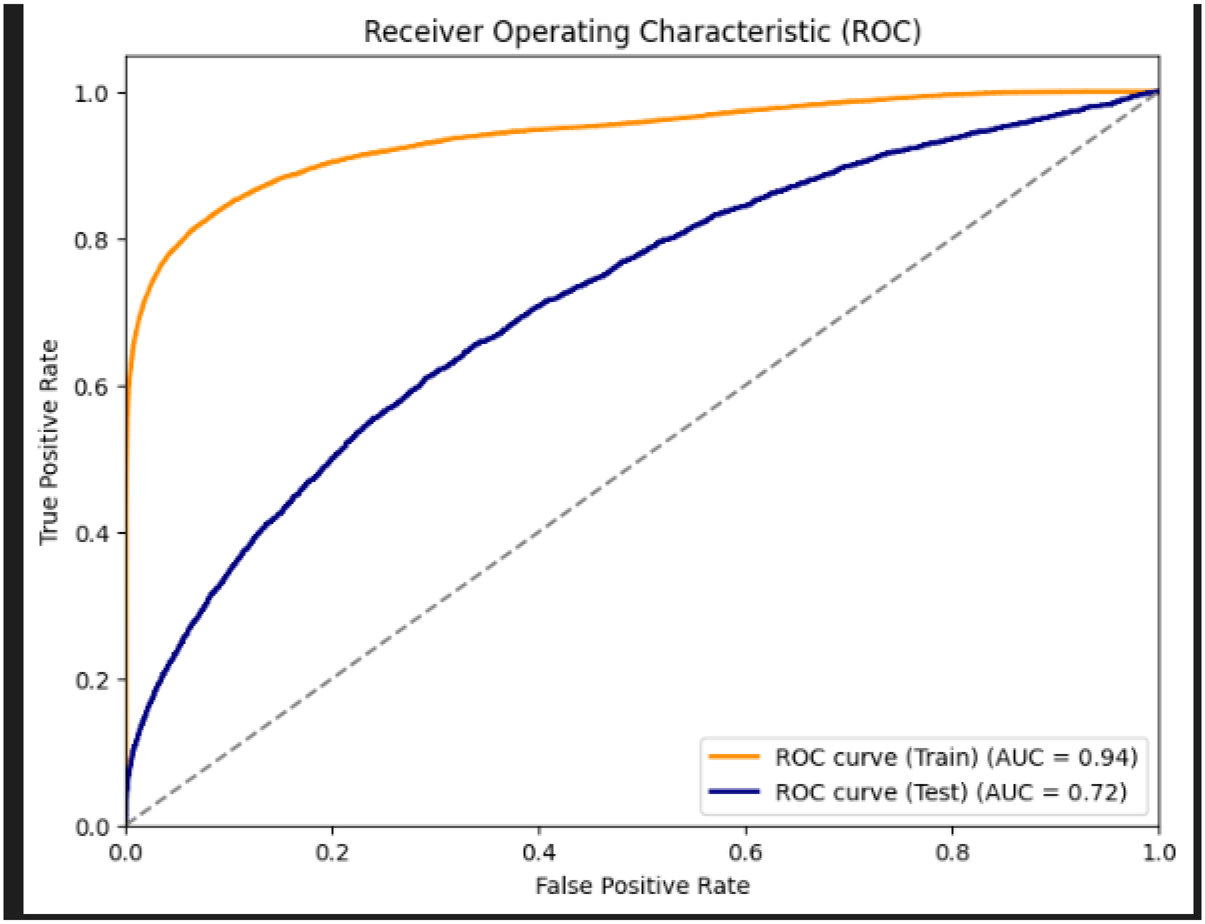


```
print("Accuracy on Test Set:", accuracy)
[35]  568m 26.6s
...  Fitting 5 folds for each of 10 candidates, totalling 50 fits
```

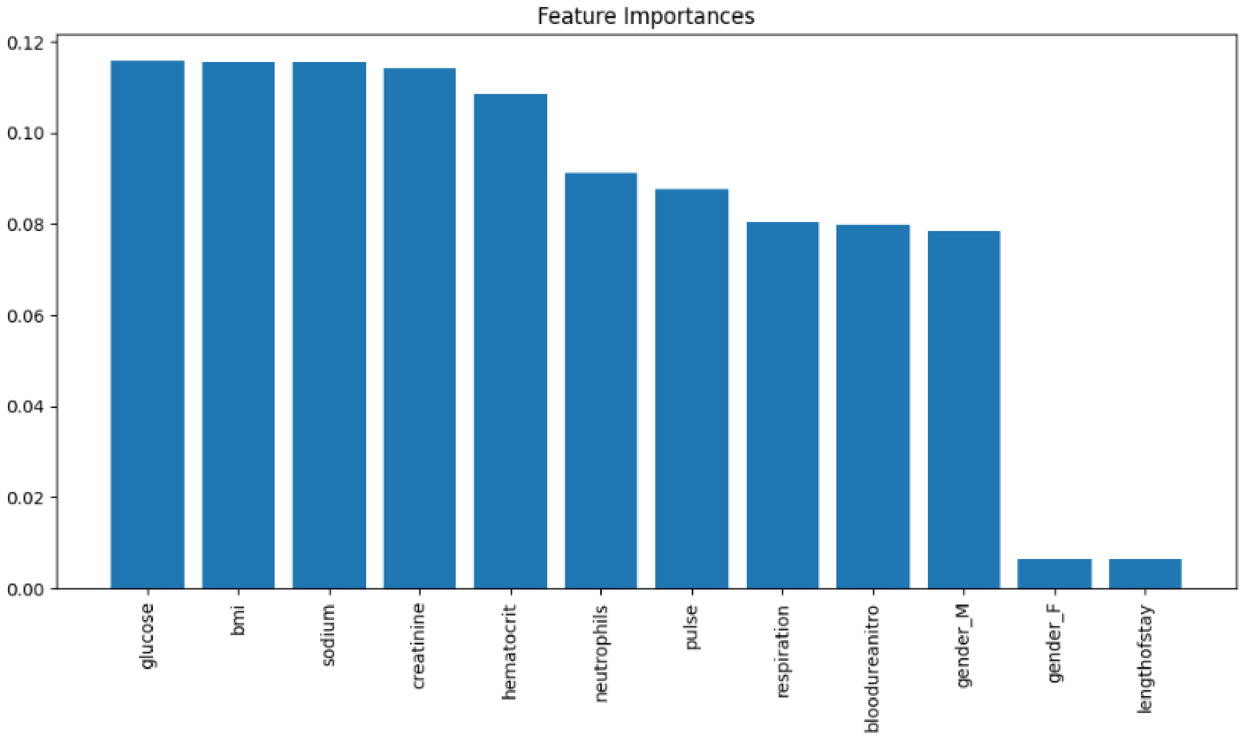
6. Results

Since we converted the y target (LOS) into a binary problem we use classification performance metric ROC. We also wanted to look at feature importance to see if any 1 variable was leading the prediction.

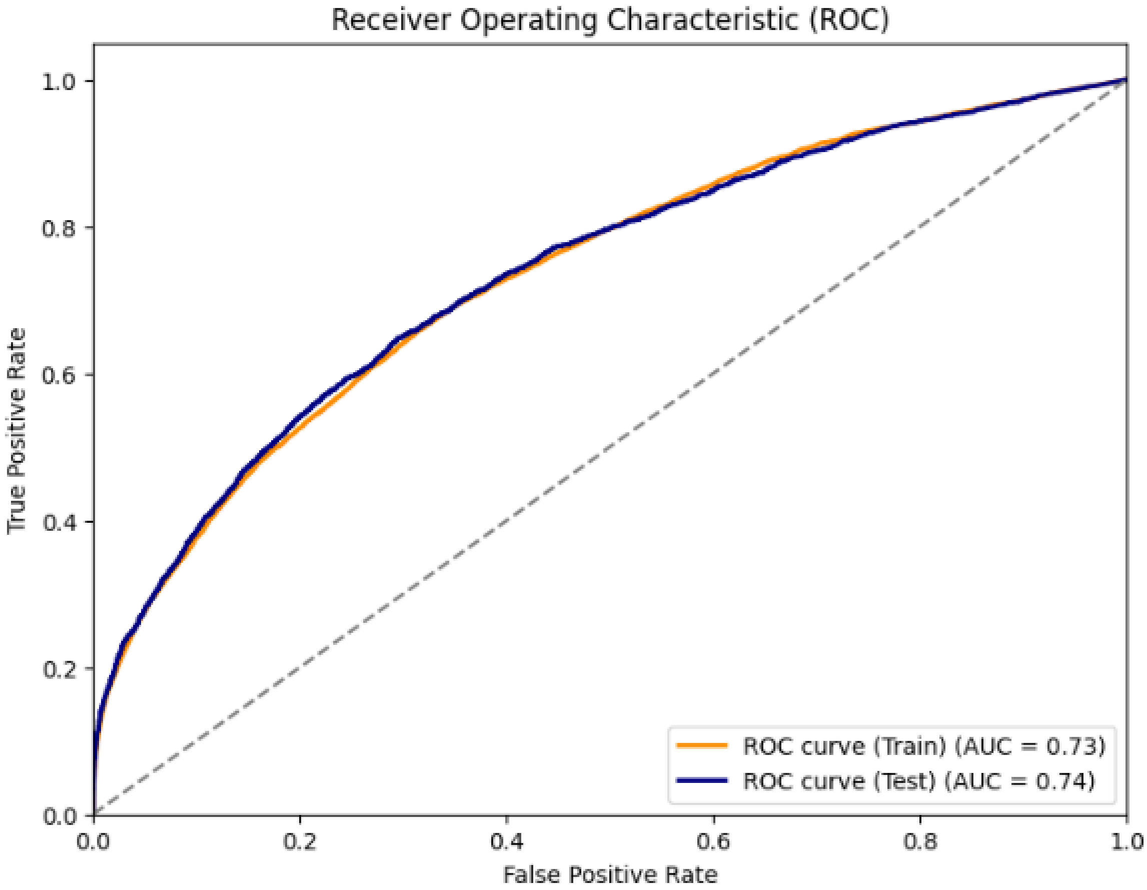
For Random Forest we see that the model is being over trained since there is such a big gap between the test and train ROC:



In terms of Feature importance we see a good distribution of importance features and random forest doesn't rely on one feature too much

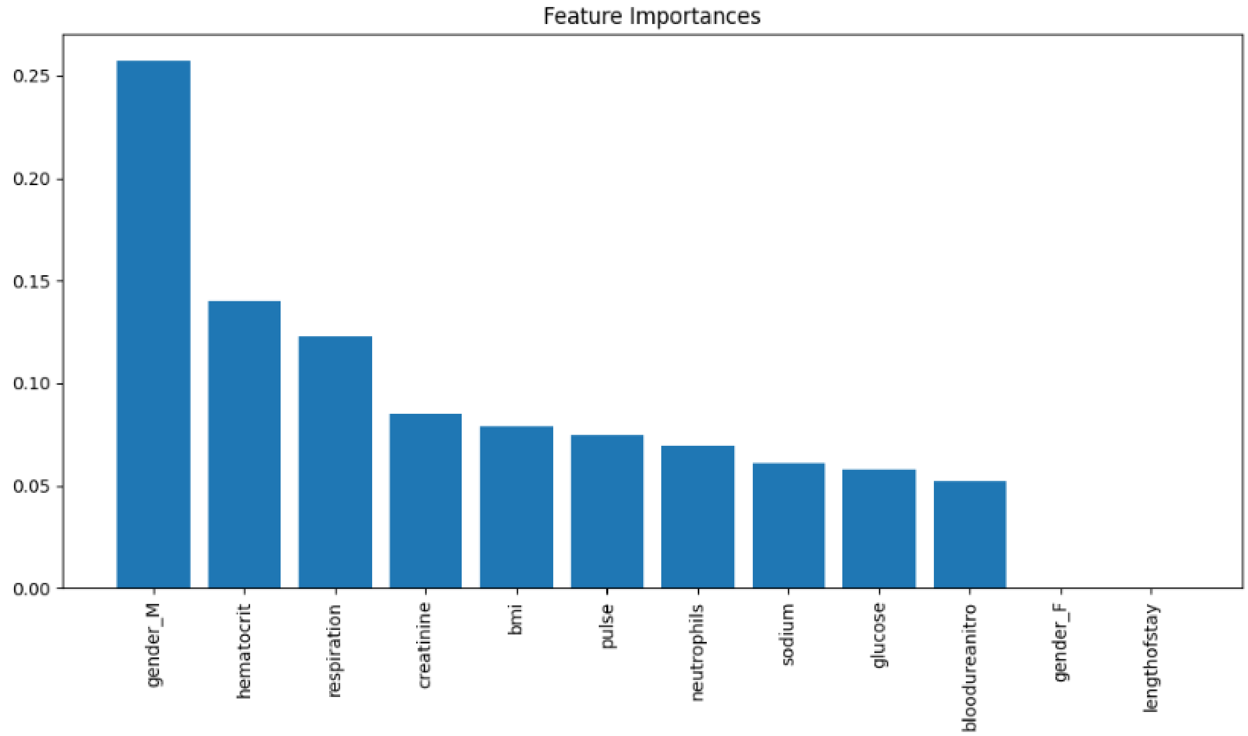


For GBM our roc test vs train is as shown below:



It has a good gap between the test and training but the accuracy of it is relatively low for health care standards. It should be mentioned that since the features included in this dataset are not significant features in determining los based on several papers read in class this model performs decently well.

In terms of feature importance, we see that gender plays a little too much importance in the prediction of the model and it would come down to domain expertise if this model should really be used in the healthcare setting



8. Conclusion

Based on previous research given in class we know that the best features for predictive modeling for LOS are not included in the dataset. So the goal would be to achieve the best model prediction. We found that GBM was the best model in terms of ROC test vs. train performance.

Some future direction we can do is to incorporate more post hoc analysis or even try to include some deep learning to predict loss. Maybe incorporating different types of gradient boosting and allocating more time for the SVM to run would also be another direction that can be taken.