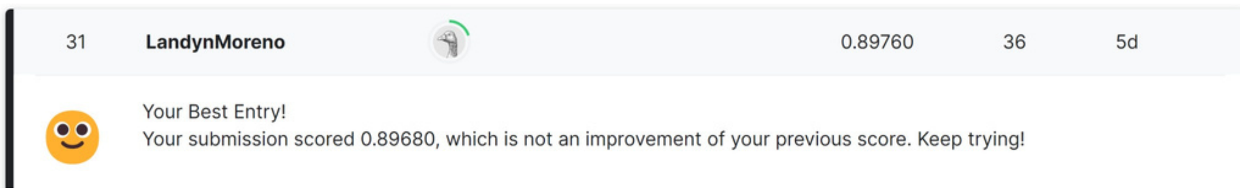
Kaggle Report

Total Entries - 36

Public Leaderboard:

Rank #31

Score - 0.89760



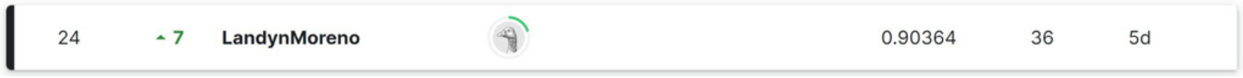
31 LandynMoreno 0.89760 36 5d

Your Best Entry!
Your submission scored 0.89680, which is not an improvement of your previous score. Keep trying!

Private Leaderboard:

Rank #24

Score - 0.90364



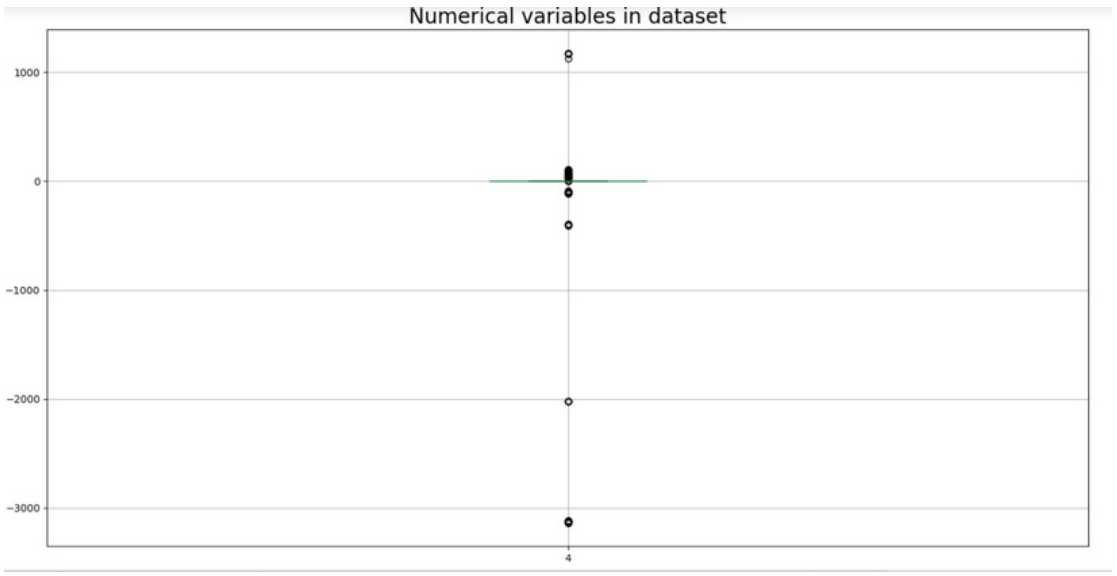
24 ▲ 7 LandynMoreno 0.90364 36 5d

In the private leaderboard, my rank increased by 7 spots.

Start of the lab:
To start this lab, I began by pre-processing the data. This consisted of simply looking through the data, checking for null values, checking for outliers, and making sure that every column is unique in some fashion.

<u>The results of this pre processing were:</u>
I foundthere to be 20 non-unique columns - everyrow has the same value.
I found there to be 0 null values in both the training and test datasets.
I also found there to be columns with outliers, there were about 10 of these.

This is an example of column 4 which I saw potential outliers on through the boxplot I made.



Since I pre-processed the data, I next tried to test how removing some of this data impacted my AUC kaggle score.

<u>Out Of Box models:</u>

For all of these, I used the full training set with no split.

ridge model -0.66812
Random forest - 0.73253
logistic regression model - 0.77487
XGB -
0.87046
Cat - 0.89352

Since CatBoost was the best OOB model, I decided to focus on tuning the hyperparameters for this model.

My first improvement was to a score of 0.89445 with the parameters (iterations=1125, learning_rate=0.015,depth=5, l2_leaf_reg=5, random_strength = 2).

These parameters were my main focus because I saw the most drastic variability with these when I tested the popular and impactful ones.

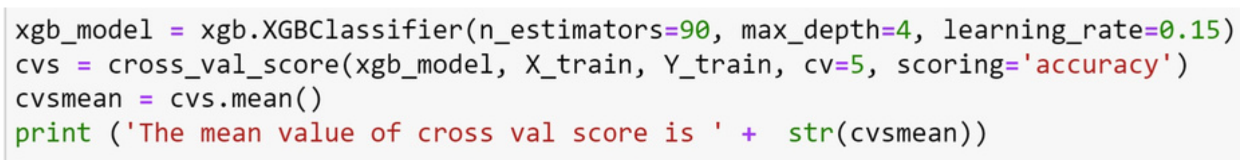
Tuning parameters:

In order to tune parameters, I used a few methods.

Grid search- I tried grid search to find the parametersand used those for xgb and cat models, but it didn't improve them consistently. I believe it only improved XGB once, so I wasn't a fan of it and felt like I wasted a few submissions with this.

Random search -Random search did better than gridsearch for me, and I used the parameters that it provided as a foundation for my manual methods.

Manual Method -Now that I had found a baseline ofparameters and their improved scores from out of box models (thanks to random search), I manually tweaked the parameters to improve my XGB and Cat models. To do so, I tested the accuracy on the training set by using cross_val_score from sklearn. This is an example of code that I ran on manually tuning and tweaking the parameters on the XGB model.



```
xgb_model = xgb.XGBClassifier(n_estimators=90, max_depth=4, learning_rate=0.15)
cvs = cross_val_score(xgb_model, X_train, Y_train, cv=5, scoring='accuracy')
cvsmean = cvs.mean()
print ('The mean value of cross val score is ' +  str(cvsmean))
```

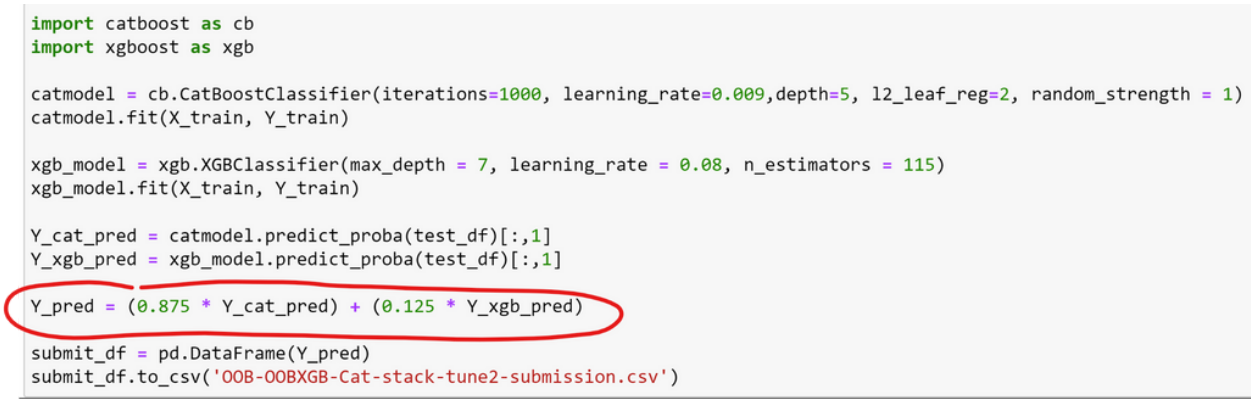
Through this manual method, my score on XGBimprovedfrom 0.87046 to 0.87635. Therefore, I knew it was working. I continued this method with the cat model as well, since XGB and Cat seemed promising by being the best OOB models.

Stacking:

Now that I found improved parameters for my XGB and Cat models. I diverted from the individual improvement of those models and I went into stacking. I used StackingClassifier to build a stacked model, and this improved my score some, but it did not do too much. When I did stacking, I made sure to have cat boost as my final estimator, and this worked the best for me. When other models were the final estimator, it did not do much.

After this attempt, I went into a different approach to gather y_pred values that I remembered from some of the previous labs, but I am not sure of the exact terminology for the method. I think it is called combining?

I went into a combination of XGB and Cat Y_predictions and combined them like so:



```
import catboost as cb
import xgboost as xgb

catmodel = cb.CatBoostClassifier(iterations=1000, learning_rate=0.009,depth=5, l2_leaf_reg=2, random_strength = 1)
catmodel.fit(X_train, Y_train)

xgb_model = xgb.XGBClassifier(max_depth = 7, learning_rate = 0.08, n_estimators = 115)
xgb_model.fit(X_train, Y_train)

Y_cat_pred = catmodel.predict_proba(test_df)[:,1]
Y_xgb_pred = xgb_model.predict_proba(test_df)[:,1]

Y_pred = (0.875 * Y_cat_pred) + (0.125 * Y_xgb_pred)

submit_df = pd.DataFrame(Y_pred)
submit_df.to_csv('OOB-OOBXGB-Cat-stack-tune2-submission.csv')
```

This actually helped my score, so I stuck to a method like this for the ongoing submissions as the deadline approached. Through variations of weighing each model's predictions, I found a good balance and improved my score to the 0.893+ range but it wasn't where I wanted it to be.

LightGBM:

I was recommended at this point in time to try LightGBM, I believe our professor mentioned it in class which led me to trying it. Therefore, I ran OOB LGB and it was okay, I tuned it some with the same methods as above, but it did not do much for me. I kind of wasted half of a day toying with this model and it didn't serve much for me.

Final Attempts/Submissions:

In this whole chronological process above, I never used any data manipulation with the pre processing that I did to start the competition. Therefore, I tried to drop columns that were not unique, or remove outliers, etc. This actually only hurt my score, ESPECIALLY DROPPING COLUMNS. After some attempts with this, I gave up on the aspect of removing data since it practically worsened every model that I trained it on.

<u>Feature Engineering:</u>

Through the recommendation in the canvas announcements, I attempted to manipulate my training set to a combination of columns and append that combination. I did this with lots of columns and found what was best. However, this only improved my score a little. So I went back to my method of combining.

One of my final attempts was a combination of xgb, lgb, and cat with a weight of:

0.86 - cat

0.03 - xgb

0.11 - lgb

This was one of the models that I selected for my final submission. Overall, the three best submissions that I had. Which I will attach in the python code were:

1.Tuned cat boost itself

2.Cat boost + xgb combination

3.Cat boost + xgb + lgb combination with feature engineering above

---

Challenges:

What I found challenging about this competition was some of the information gathering on the internet being complex to follow. For example, when I went into stacking, I struggled to gather and understand all of the information that I needed/wanted. This led to me more so neglecting stacking when I felt like it actually may have improved my score in the long run.

Additionally, I understand that the process of this is almost like guessing and checking, but I felt like there were times where I was almost distracted and did more of “guessing”. Sometimes, I didn’t logically tune a parameter and would rather just guess one and it led to a good amount of wasted time/submissions.

I also had some trouble with judging if a model/method would be a good AUC score on kaggle because sometimes my calculations would say that it was solid, but it was not very good.

Reflection:

Overall this competition was great for me, I actually learned way more than I have through all of the labs combined. I think that it was great because of the time deadline which forced me to work diligently and carefully. This was a cool idea too because it is a direct competition with classmates. Other than the challenges, I did like this assignment a ton and it built my confidence going into the upcoming exam week as well. I think that I picked the 3 best submissions for my private score and I was happy with my private leaderboard score and how I jumped 7 spots.

The most important thing that I learned from this and feel like I grew the most was in parameter tuning. I learned different methods of parameter tuning and found ways to do it in a good manner.