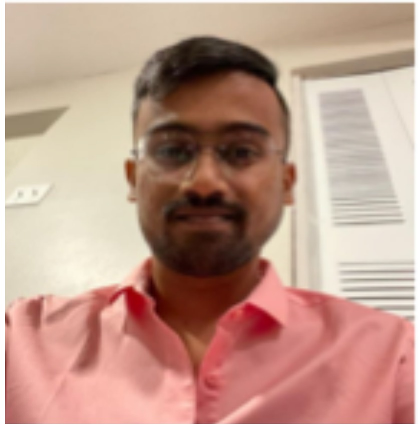
HCDR Team 1

Phase 2

Zack Seliger

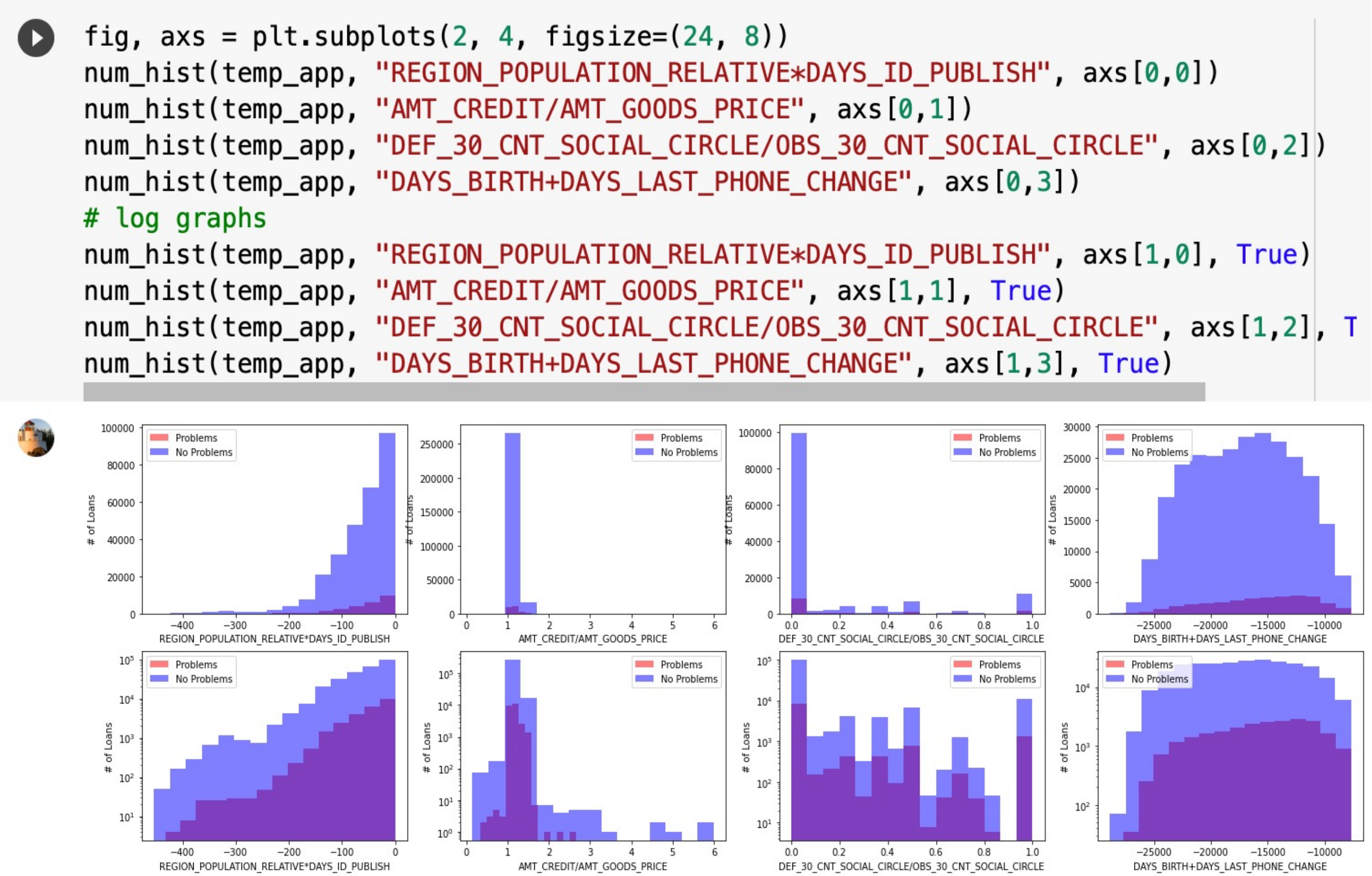
Keegan Moore

Jagan Lakku



Raja Simha Reddy

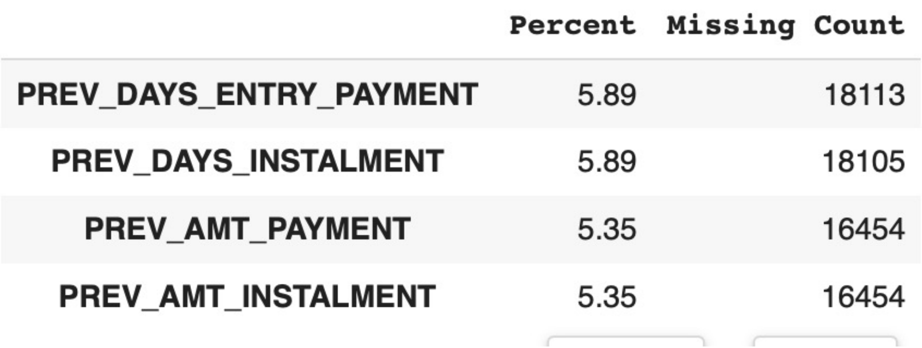
ANALYSIS OF NEW FEATURES



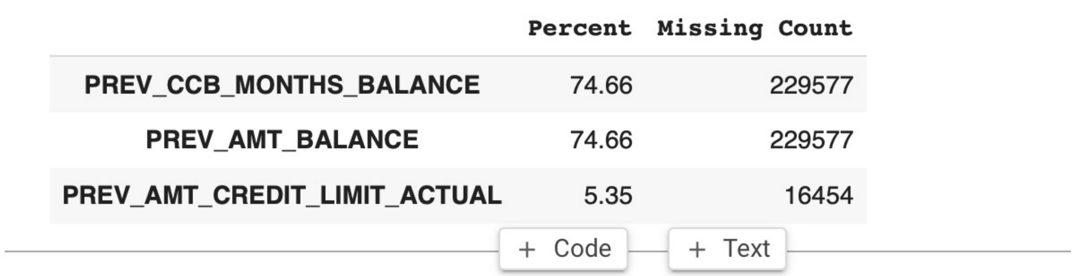
```
fig, axs = plt.subplots(2, 4, figsize=(24, 8))
num_hist(temp_app, "REGION_POPULATION_RELATIVE*DAYS_ID_PUBLISH", axs[0,0])
num_hist(temp_app, "AMT_CREDIT/AMT_GOODS_PRICE", axs[0,1])
num_hist(temp_app, "DEF_30_CNT_SOCIAL_CIRCLE/OBS_30_CNT_SOCIAL_CIRCLE", axs[0,2])
num_hist(temp_app, "DAYS_BIRTH+DAYS_LAST_PHONE_CHANGE", axs[0,3])
# log graphs
num_hist(temp_app, "REGION_POPULATION_RELATIVE*DAYS_ID_PUBLISH", axs[1,0], True)
num_hist(temp_app, "AMT_CREDIT/AMT_GOODS_PRICE", axs[1,1], True)
num_hist(temp_app, "DEF_30_CNT_SOCIAL_CIRCLE/OBS_30_CNT_SOCIAL_CIRCLE", axs[1,2], T
num_hist(temp_app, "DAYS_BIRTH+DAYS_LAST_PHONE_CHANGE", axs[1,3], True)
```

- ➢ Extending the work done in phase 1 , we explored more deep into the data sets considering all the secondary data and These relationships are quite interesting
- ➢ By referencing the graphs above this set, REGION_POPULATION_RELATIVE and DAYS_ID_PUBLISH have graphs with one high point around the middle.
  However, REGION_POPULATION_RELATION*DAYS_ID_PUBLISH has a clear trend that, the further to the right, the more problems the client has with repayment

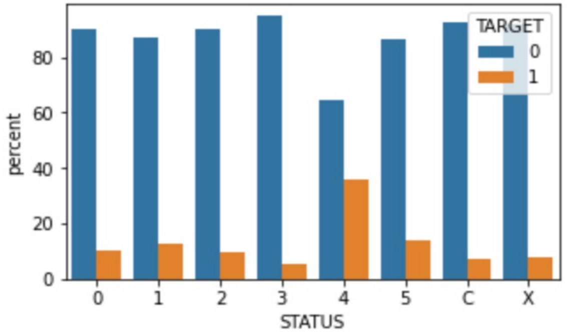
More New Features



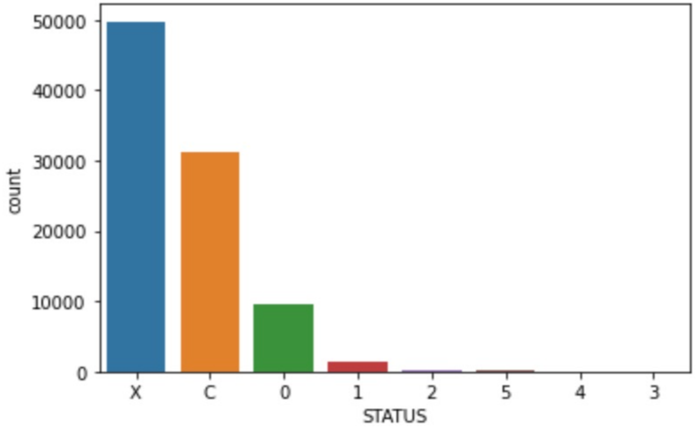
	Percent	Missing Count
PREV_DAYS_ENTRY_PAYMENT	5.89	18113
PREV_DAYS_INSTALMENT	5.89	18105
PREV_AMT_PAYMENT	5.35	16454
PREV_AMT_INSTALMENT	5.35	16454



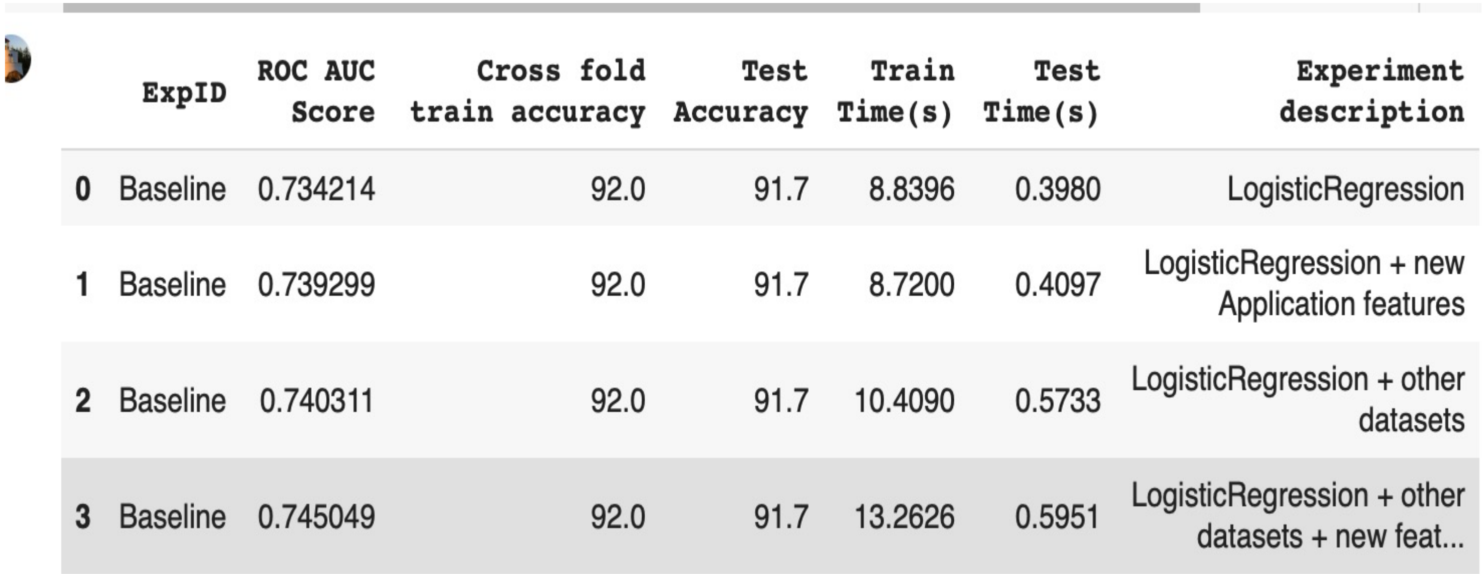
	Percent	Missing Count
PREV_CCB_MONTHS_BALANCE	74.66	229577
PREV_AMT_BALANCE	74.66	229577
PREV_AMT_CREDIT_LIMIT_ACTUAL	5.35	16454



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa501a8ee50>
```



Exploration Of Datasets on Baselinemodel

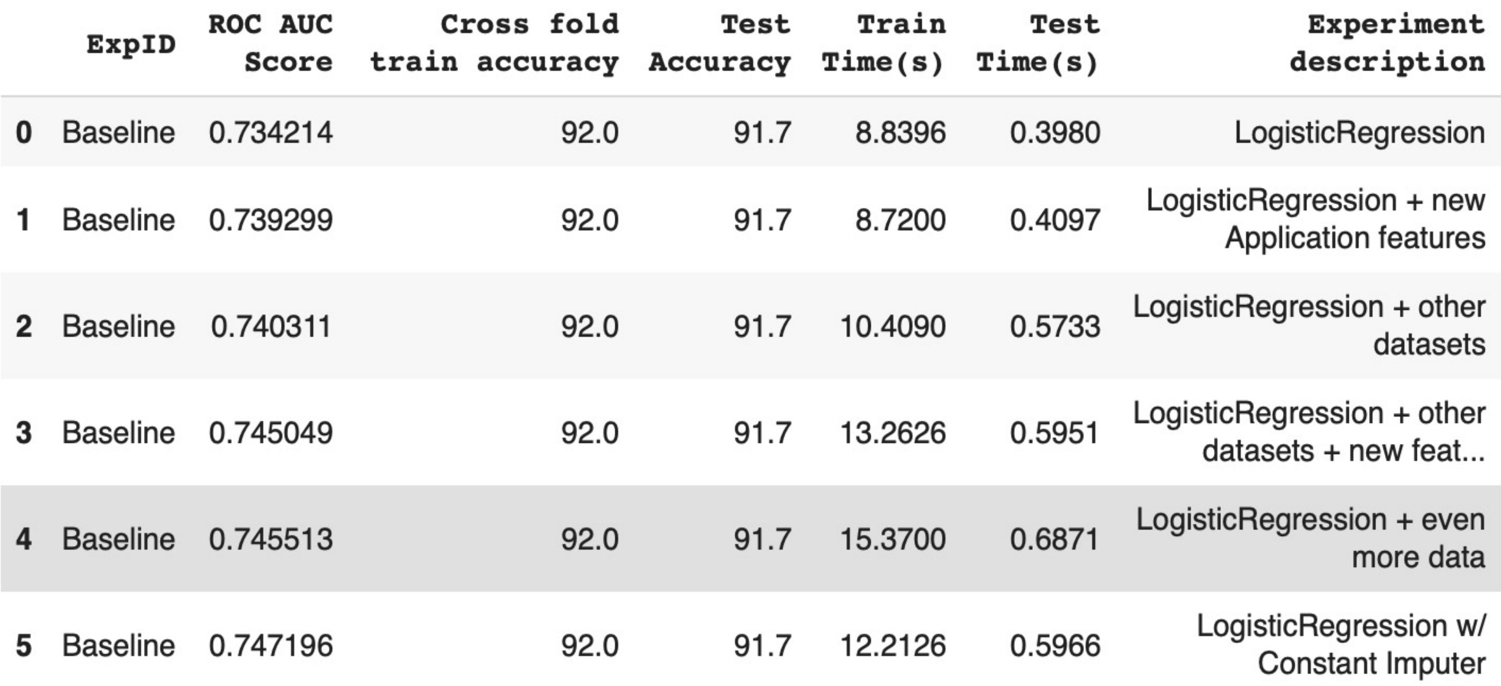


	ExpID	ROC AUC Score	Cross fold train accuracy	Test Accuracy	Train Time(s)	Test Time(s)	Experiment description
0	Baseline	0.734214	92.0	91.7	8.8396	0.3980	LogisticRegression
1	Baseline	0.739299	92.0	91.7	8.7200	0.4097	LogisticRegression + new Application features
2	Baseline	0.740311	92.0	91.7	10.4090	0.5733	LogisticRegression + other datasets
3	Baseline	0.745049	92.0	91.7	13.2626	0.5951	LogisticRegression + other datasets + new feat...

- ➢ Conducted EDA on all the Secondary Data
- ➢ Did Baseline Model for Application data and all Secondary Data
- ➢ Explored LogisticRegression on New Application Features and other Datasets and found the ROC AUC score which doesn’t really improve
- ➢ LogisticRegression seemed to improve with the new feature on all the Datasets

TWEAKING IMPUTERS

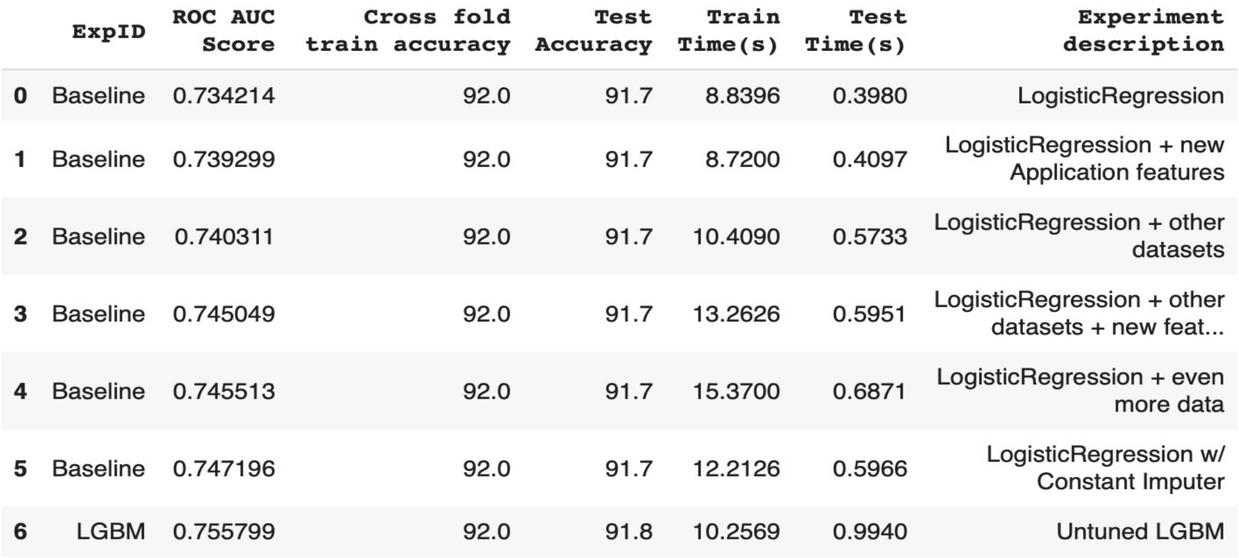
- We should be using the categorical imputer with a constant strategy. Instead of assigning NaN data with the most frequent category, maybe we should instead create a new category for all of this data.
- This would deal with certain categories, like employment data types, where it seemed that unemployed clients were labelled as NaN and shouldn't be grouped in with other categories.
- When compared it with experiment 3, it performs better. We should continue using this change to the imputer in the future



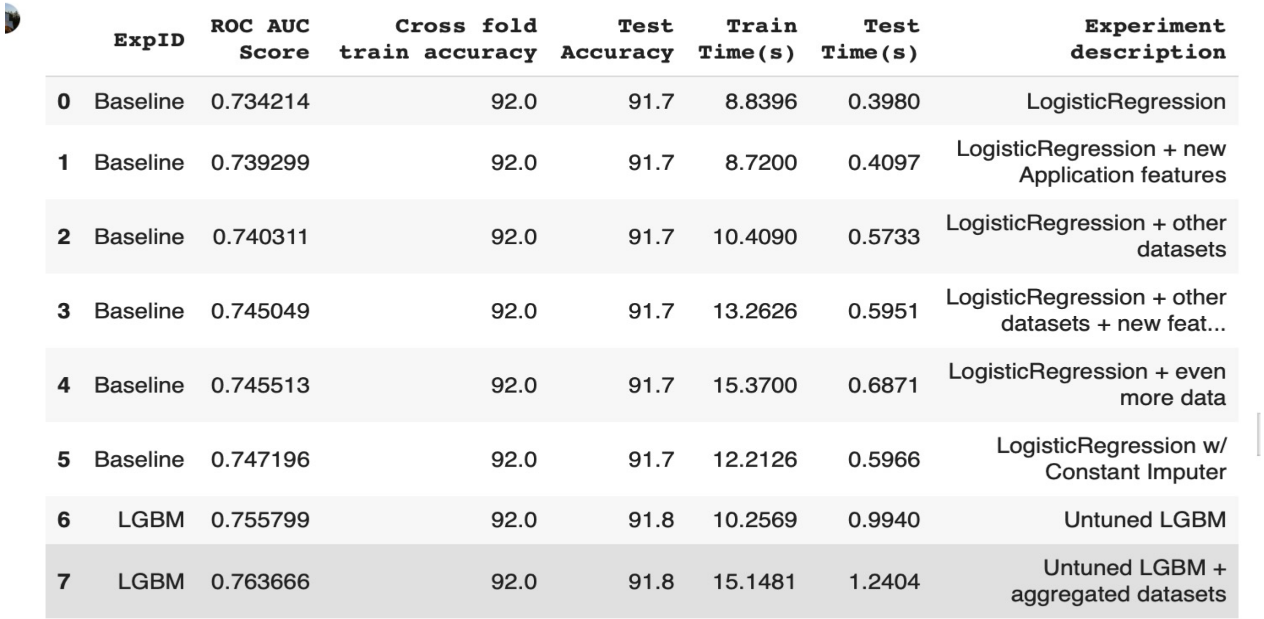
	ExpID	ROC AUC Score	Cross fold train accuracy	Test Accuracy	Train Time(s)	Test Time(s)	Experiment description
0	Baseline	0.734214	92.0	91.7	8.8396	0.3980	LogisticRegression
1	Baseline	0.739299	92.0	91.7	8.7200	0.4097	LogisticRegression + new Application features
2	Baseline	0.740311	92.0	91.7	10.4090	0.5733	LogisticRegression + other datasets
3	Baseline	0.745049	92.0	91.7	13.2626	0.5951	LogisticRegression + other datasets + new feat...
4	Baseline	0.745513	92.0	91.7	15.3700	0.6871	LogisticRegression + even more data
5	Baseline	0.747196	92.0	91.7	12.2126	0.5966	LogisticRegression w/ Constant Imputer

UNTUNED LGBM

- Trained an LGBM Classifier on the data from application plus the data from our datasets and new engineered features
- It performs the best of any models we have yet. We still need to tune it, and can add more data
- These new features clearly make our model better. Now, we need to do optimize it using Grid Search

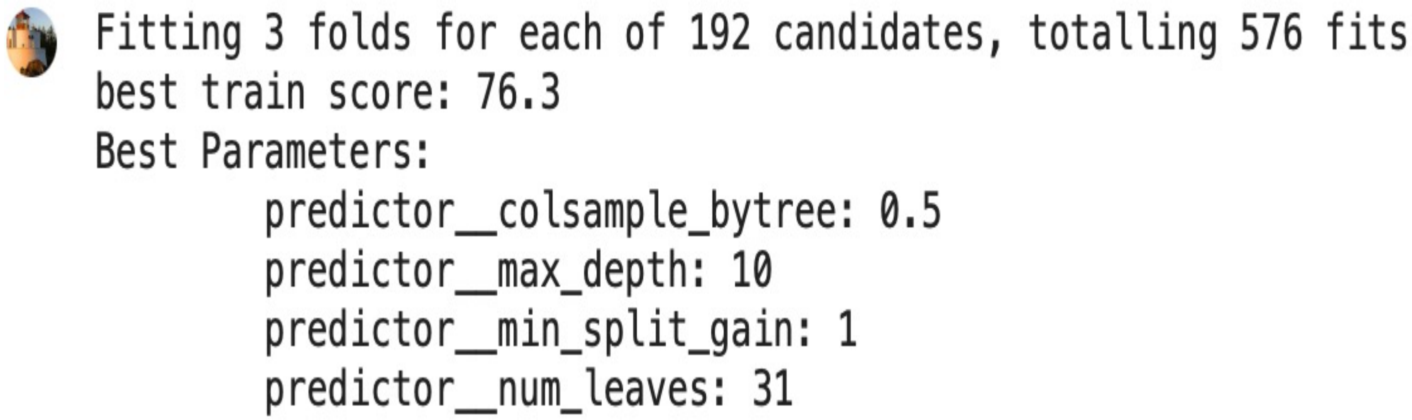


	ExpID	ROC AUC Score	Cross fold train accuracy	Test Accuracy	Train Time(s)	Test Time(s)	Experiment description
0	Baseline	0.734214	92.0	91.7	8.8396	0.3980	LogisticRegression
1	Baseline	0.739299	92.0	91.7	8.7200	0.4097	LogisticRegression + new Application features
2	Baseline	0.740311	92.0	91.7	10.4090	0.5733	LogisticRegression + other datasets
3	Baseline	0.745049	92.0	91.7	13.2626	0.5951	LogisticRegression + other datasets + new feat...
4	Baseline	0.745513	92.0	91.7	15.3700	0.6871	LogisticRegression + even more data
5	Baseline	0.747196	92.0	91.7	12.2126	0.5966	LogisticRegression w/ Constant Imputer
6	LGBM	0.755799	92.0	91.8	10.2569	0.9940	Untuned LGBM



	ExpID	ROC AUC Score	Cross fold train accuracy	Test Accuracy	Train Time(s)	Test Time(s)	Experiment description
0	Baseline	0.734214	92.0	91.7	8.8396	0.3980	LogisticRegression
1	Baseline	0.739299	92.0	91.7	8.7200	0.4097	LogisticRegression + new Application features
2	Baseline	0.740311	92.0	91.7	10.4090	0.5733	LogisticRegression + other datasets
3	Baseline	0.745049	92.0	91.7	13.2626	0.5951	LogisticRegression + other datasets + new feat...
4	Baseline	0.745513	92.0	91.7	15.3700	0.6871	LogisticRegression + even more data
5	Baseline	0.747196	92.0	91.7	12.2126	0.5966	LogisticRegression w/ Constant Imputer
6	LGBM	0.755799	92.0	91.8	10.2569	0.9940	Untuned LGBM
7	LGBM	0.763666	92.0	91.8	15.1481	1.2404	Untuned LGBM + aggregated datasets

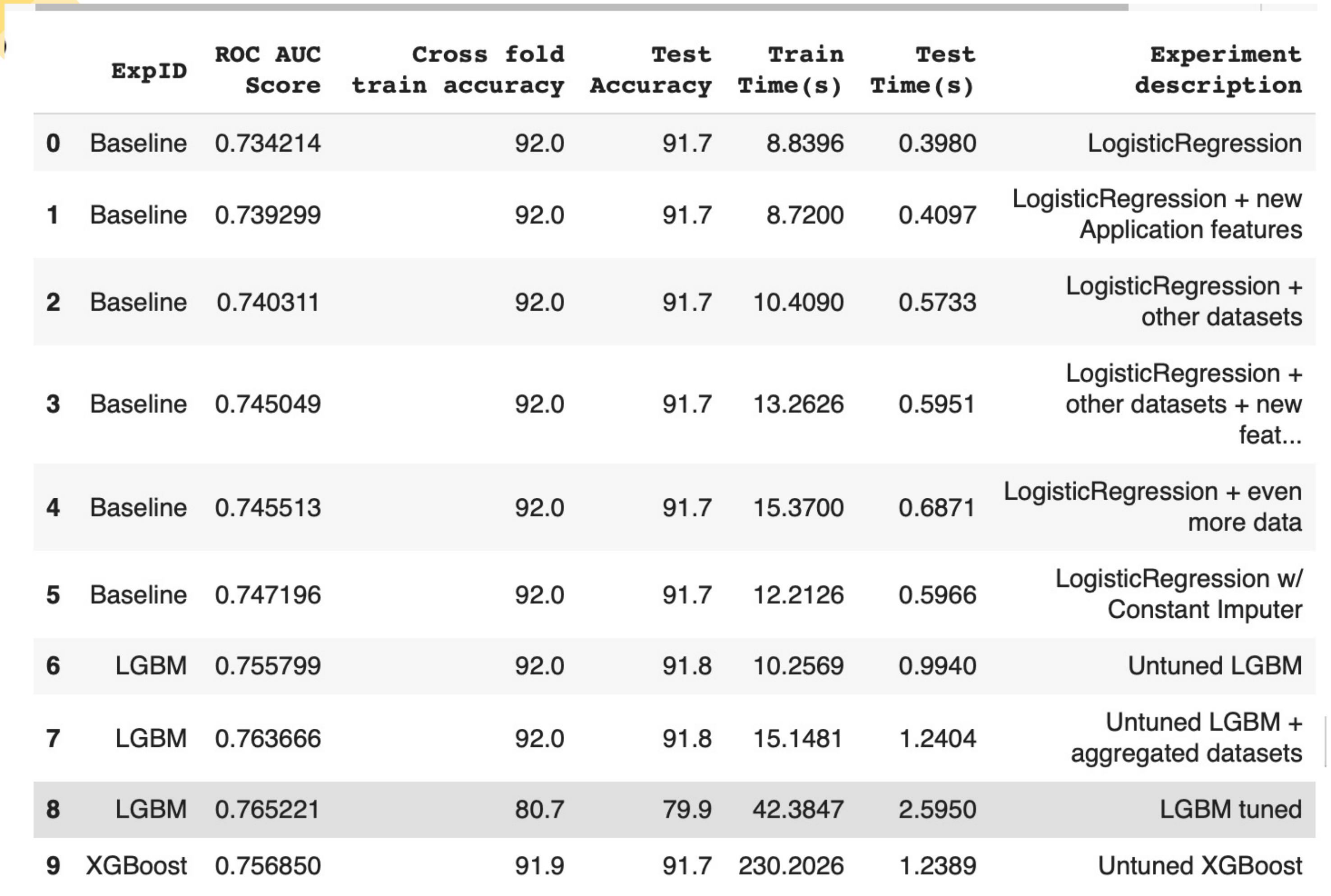
GRID SEARCH FOR LGBM



```
Fitting 3 folds for each of 192 candidates, totalling 576 fits
best train score: 76.3
Best Parameters:
        predictor__colsample_bytree: 0.5
        predictor__max_depth: 10
        predictor__min_split_gain: 1
        predictor__num_leaves: 31
```

- Since Grid Search takes a while when testing for larger values of n_estimators, we tested all other hyperparameters before testing n_estimators
- And the best parameters which fit the LGBM model for given data is as below

TUNED LGBM

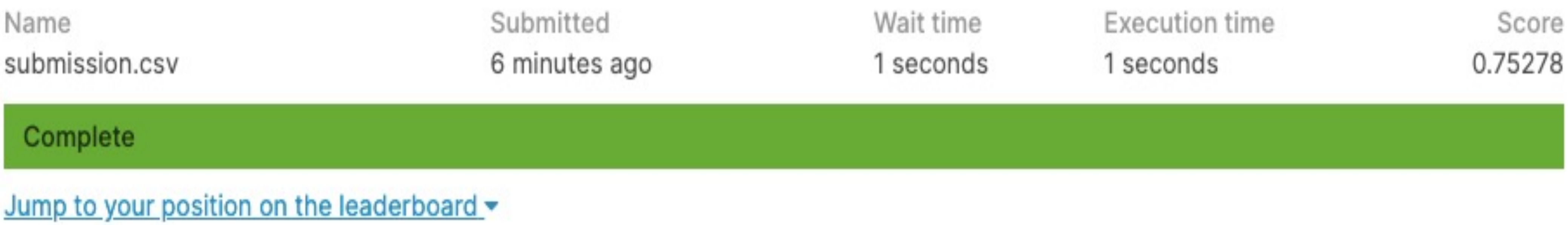


	ExpID	ROC AUC Score	Cross fold train accuracy	Test Accuracy	Train Time(s)	Test Time(s)	Experiment description
0	Baseline	0.734214	92.0	91.7	8.8396	0.3980	LogisticRegression
1	Baseline	0.739299	92.0	91.7	8.7200	0.4097	LogisticRegression + new Application features
2	Baseline	0.740311	92.0	91.7	10.4090	0.5733	LogisticRegression + other datasets
3	Baseline	0.745049	92.0	91.7	13.2626	0.5951	LogisticRegression + other datasets + new feat...
4	Baseline	0.745513	92.0	91.7	15.3700	0.6871	LogisticRegression + even more data
5	Baseline	0.747196	92.0	91.7	12.2126	0.5966	LogisticRegression w/ Constant Imputer
6	LGBM	0.755799	92.0	91.8	10.2569	0.9940	Untuned LGBM
7	LGBM	0.763666	92.0	91.8	15.1481	1.2404	Untuned LGBM + aggregated datasets
8	LGBM	0.765221	80.7	79.9	42.3847	2.5950	LGBM tuned
9	XGBoost	0.756850	91.9	91.7	230.2026	1.2389	Untuned XGBoost

- We have added the extra data that we got from the "More Data" experiment to an LGBM model with tuned hyperparameters to see if it can improve the
- Untuned XGBoost Model:

We have tested if XGBoost can do any better than Tuned LGBM , but LGBM does it more better .

KAGGLE SUBMISSION



Name	Submitted	Wait time	Execution time	Score
submission.csv	6 minutes ago	1 seconds	1 seconds	0.75278

Complete

Jump to your position on the leaderboard ▾