# FP Phase 2 - Final Project

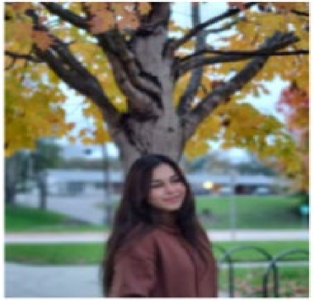
## Final Project HCDR - feature engineering + hyperparameter tuning

## Group: 11

Aditi Mulye
adimulye@iu.edu

Keshav Likhar
klikhar@iu.edu

Nikunj Malpani
nmalpani@iu.edu



Prashasti Karlekar
prkarl@iu.edu

# Slides Outline:

- **Project Description:** Here, we’ve summarized what the project is all about and the goals we are expecting to achieve from this project!
- **Summary EDA:** In this section, we have described our dataset visually and tried to understand the data and gather insights from it.
- **Feature Engineering:** Here, we merged the additional files of the dataset and tried using them in our model prediction. We also created some additional features which were correlated with the target variable.
- **Modelling Pipeline/ Hyperparameter Tuning:** Here, we’ve given an overview of the modelling techniques that we used in our baseline model creation and then selected important features, built new columns and tried to improve the model score and also focused on  performing hyperparameter tuning.
- **Result & Result Discussion:** In this section, we have attached our Kaggle Submission Screenshot, and shown the accuracy score of model performance of different models.
- **Conclusion and Next Steps:** For this section, we have briefly discussed the steps to be taken further and improvements that can be done in the model!

Thus, we have divided the entire Phase II Slide into these sections and tried to work on the data.

# Project Description

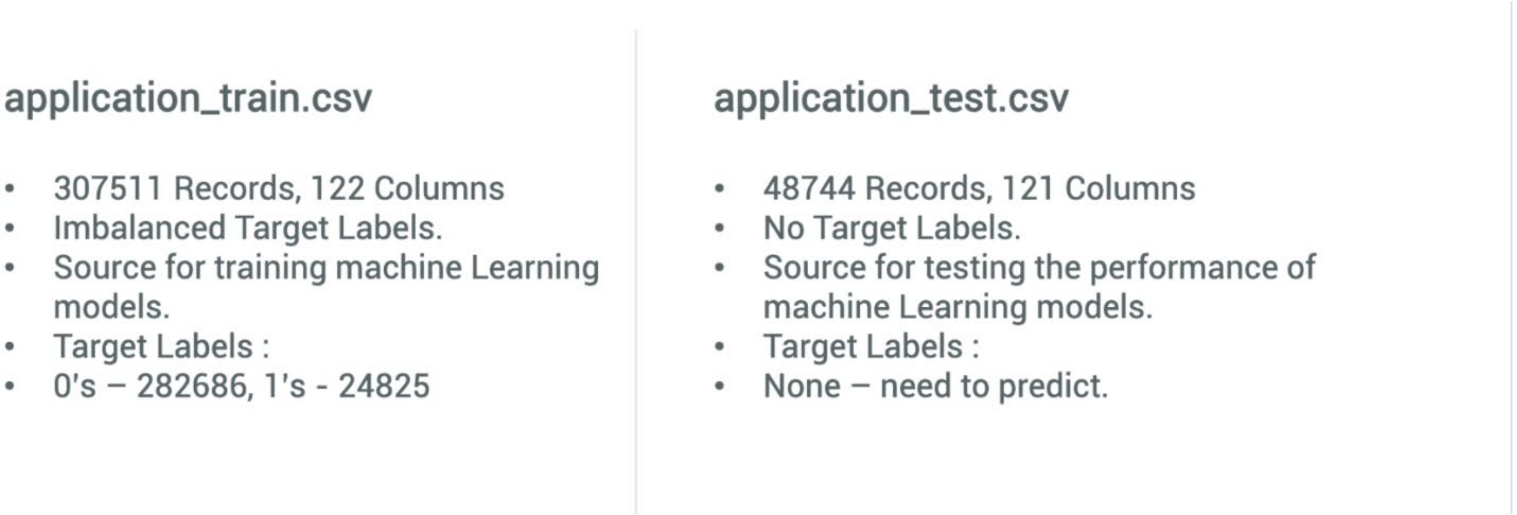
The objective of Home Credit is to correctly offer loans to individuals who can pay back and turn away those who cannot. The challenge lies in the great diversity of backgrounds of individuals who come to Home Credit to procure the loan. In this analysis, we aim to implement a variety of techniques to assess the idiosyncrasies of each customer and determine whether a customer will default.

In this phase of the project, we performed feature engineering by building new columns with the additional datasets provided within the data and used them to make further predictions. We performed Hyperparameter Tuning by using GridSearchCV and documented our results. We used ROC Curve, Confusion Metrics and Log Loss to predict our model performance.

## Steps :-

- During the Phase I of the project, we achieved an accuracy of 73% on Kaggle Submission.
- For this Phase of the project, we combined all the 7 datasets for our predictions, also building new columns according to their correlation with the target variable.
- We tried considering resampling techniques like SMOTE and ADASYN, but, to our utter surprise, the model performance decreased in this case.
- Finally, we performed Hyperparameter Tuning using GridSearchCV and enhanced the model score on Kaggle by 1.5%

# Data Description



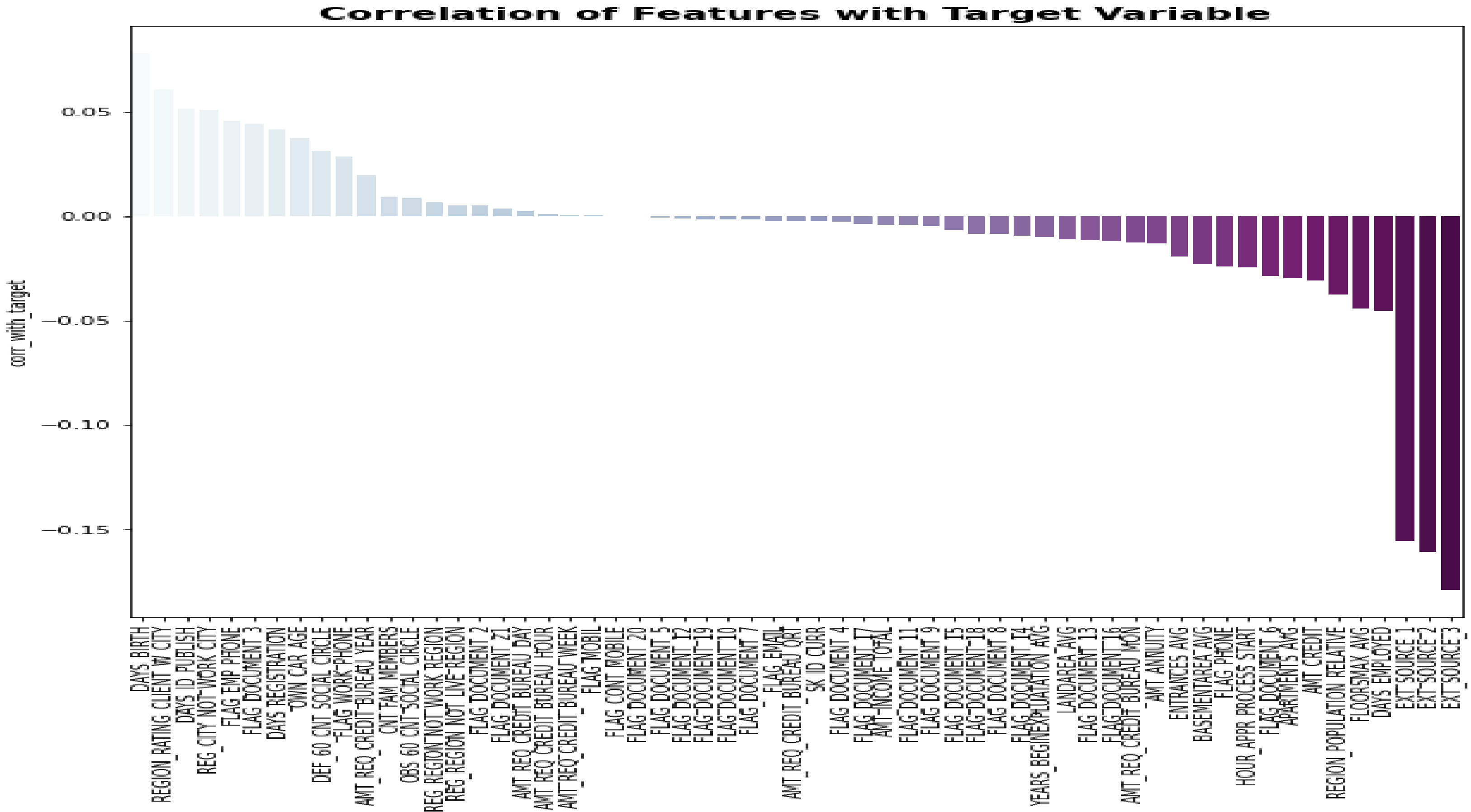
## application_train.csv

- 307511 Records, 122 Columns
- Imbalanced Target Labels.
- Source for training machine Learning models.
- Target Labels :
- 0's – 282686, 1's - 24825

## application_test.csv

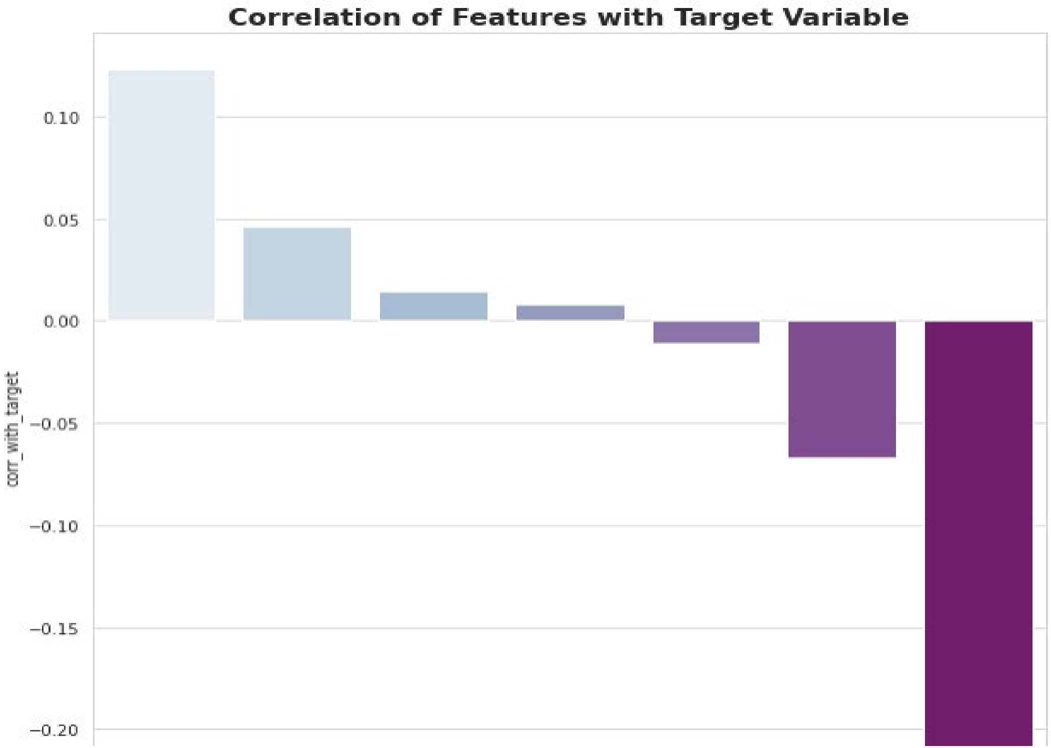
- 48744 Records, 121 Columns
- No Target Labels.
- Source for testing the performance of machine Learning models.
- Target Labels :
- None – need to predict.

# A Few Visual Exploratory Data Analysis



# Exploratory Data Analysis on New Engineered Features

**Here is the plot of correlation of the new variables with the target variable. We can see that the correlation has improved on performing Feature Engineering.**



# Modelling Pipeline

Since the data has both numerical and categorical features, it is required to create two pipelines (one for each category of data) because they require different transformations. After finishing that, the two pipelines should be unified to produce one full pipeline that performs transformation on all the dataset.

**Logistic Regression:** Logistic regression is a process of modeling the probability of a discrete outcome given an input variable.
The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.
Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes. Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful analytic technique.

**Random Forest Classifier:** A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.
A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

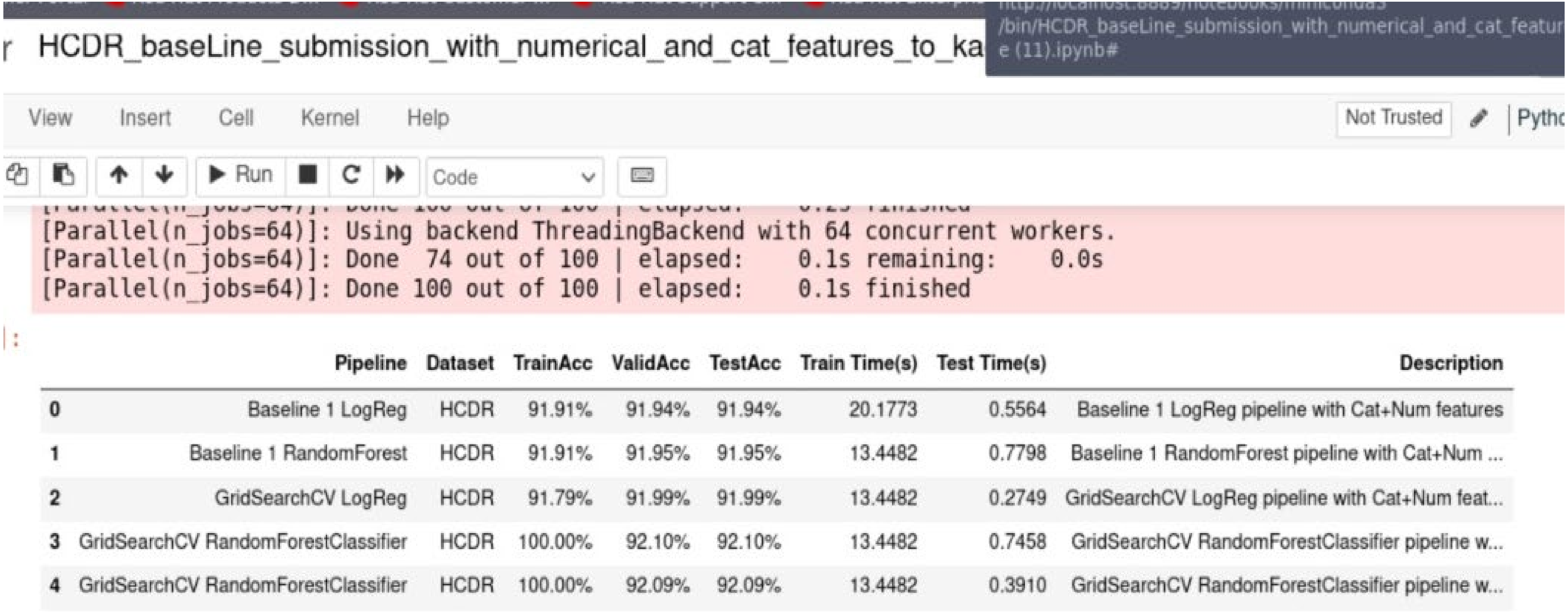
# Hyperparameter Tuning & GridSearchCV

**Hyperparameter tuning** is choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a model argument whose value is set before the learning process begins. The key to machine learning algorithms is hyperparameter tuning.

- We used **Hyperparameter Tuning** for both of our models, Random Forest and Logistic Regression.
- Our model was performing better in the case of **Random Forest** and we achieved an accuracy of about 92.10% in case of RF, whereas 91.9% for **Logistic Regression**.

In **GridSearchCV,** along with Grid Search, cross-validation is also performed. Cross-Validation is used while training the model. As we know that before training the model with data, we divide the data into two parts – train data and test data. In cross-validation, the process divides the train data further into two parts – the train data and the validation data.

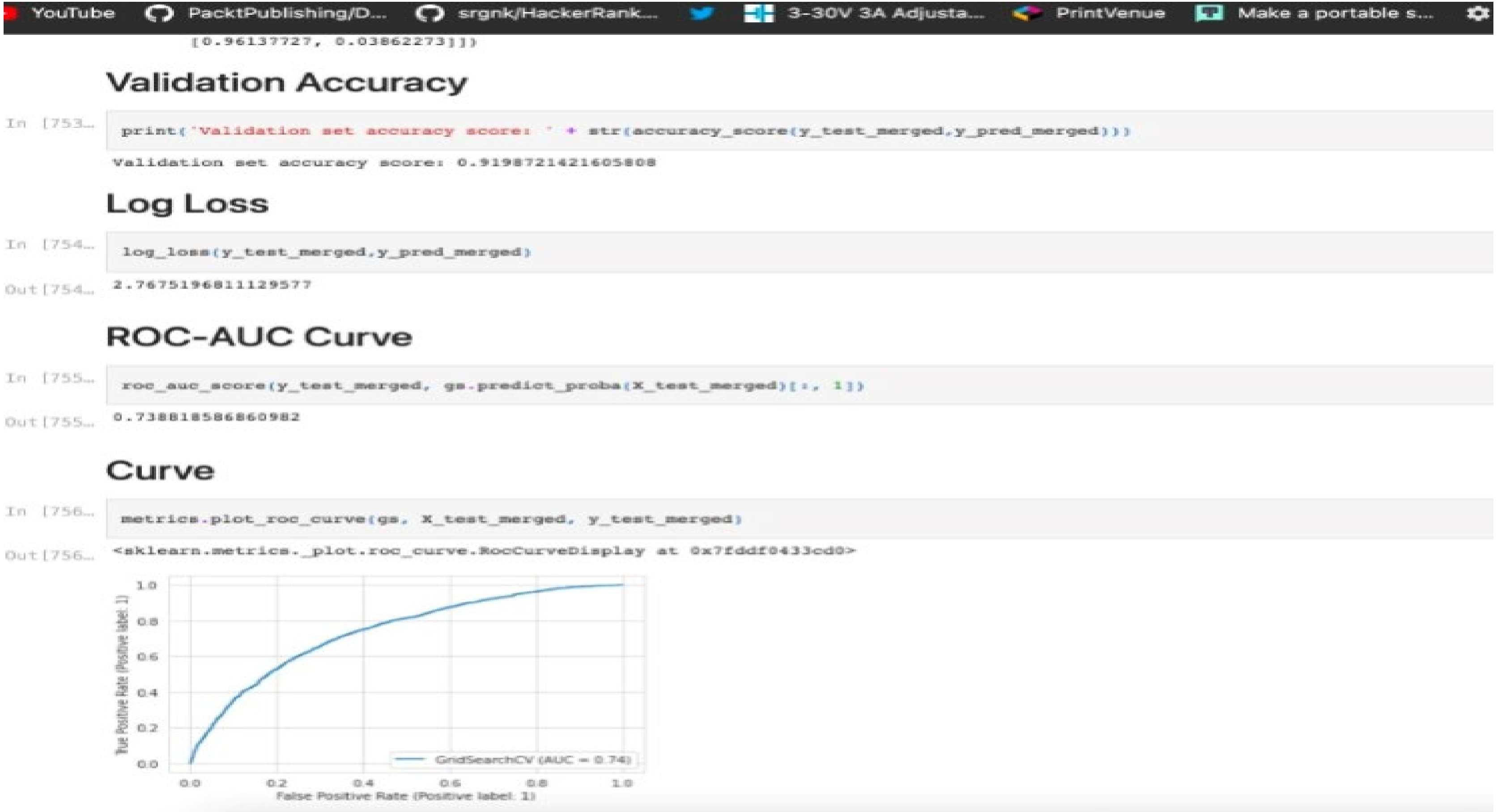
# Results & Discussion of Results:


r HCDR_baseLine_submission_with_numerical_and_cat_features_to_ka
http://localhost:8889/notebooks/miniconda3
/bin/HCDR_baseLine_submission_with_numerical_and_cat_featur
e (11).ipynb#
View
Insert
Cell
Kernel
Help
Not Trusted
Pytho
Run
Code

```
[Parallel(n_jobs=64)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=64)]: Using backend ThreadingBackend with 64 concurrent workers.
[Parallel(n_jobs=64)]: Done  74 out of 100 | elapsed:    0.1s remaining:    0.0s
[Parallel(n_jobs=64)]: Done 100 out of 100 | elapsed:    0.1s finished
```

|  | Pipeline | Dataset | TrainAcc | ValidAcc | TestAcc | Train Time(s) | Test Time(s) | Description |
|---|---|---|---|---|---|---|---|---|
| 0 | Baseline 1 LogReg | HCDR | 91.91% | 91.94% | 91.94% | 20.1773 | 0.5564 | Baseline 1 LogReg pipeline with Cat+Num features |
| 1 | Baseline 1 RandomForest | HCDR | 91.91% | 91.95% | 91.95% | 13.4482 | 0.7798 | Baseline 1 RandomForest pipeline with Cat+Num ... |
| 2 | GridSearchCV LogReg | HCDR | 91.79% | 91.99% | 91.99% | 13.4482 | 0.2749 | GridSearchCV LogReg pipeline with Cat+Num feat... |
| 3 | GridSearchCV RandomForestClassifier | HCDR | 100.00% | 92.10% | 92.10% | 13.4482 | 0.7458 | GridSearchCV RandomForestClassifier pipeline w... |
| 4 | GridSearchCV RandomForestClassifier | HCDR | 100.00% | 92.09% | 92.09% | 13.4482 | 0.3910 | GridSearchCV RandomForestClassifier pipeline w... |

# Results & Discussion of Results:



```
[0.96137727, 0.03862273]])
```

## Validation Accuracy

```
In [753… print('Validation set accuracy score: ' + str(accuracy_score(y_test_merged,y_pred_merged)))
```

```
Validation set accuracy score: 0.9198721421605808
```

## Log Loss

```
In [754… log_loss(y_test_merged,y_pred_merged)
```

```
Out[754… 2.7675196811129577
```

## ROC-AUC Curve

```
In [755… roc_auc_score(y_test_merged, gs.predict_proba(X_test_merged)[:, 1])
```

```
Out[755… 0.7388185868609 82
```

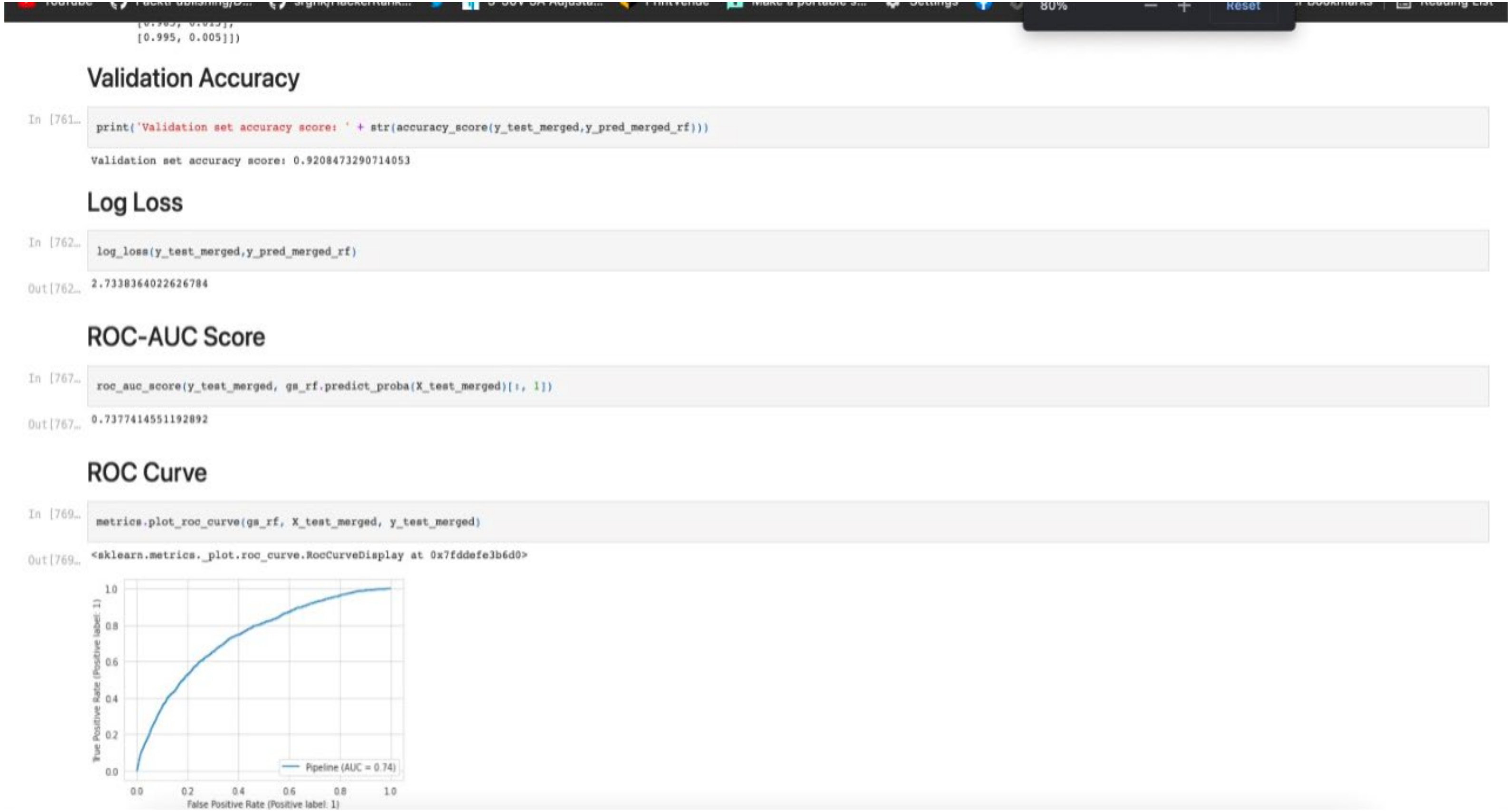
## Curve

```
In [756… metrics.plot_roc_curve(gs, X_test_merged, y_test_merged)
```

```
Out[756… <sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x7fddf0433cd0>
```

1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (Positive label: 1)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1)
GridSearchCV (AUC = 0.74)

# Results & Discussion of Results:


80%
Reset
Reading List
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (Positive label: 1)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1)
Pipeline (AUC = 0.74)

```
[0.995, 0.005]])
```

## Validation Accuracy

```
In [761…  print('Validation set accuracy score: ' + str(accuracy_score(y_test_merged,y_pred_merged_rf)))

Validation set accuracy score: 0.9208473290714053
```

## Log Loss

```
In [762…  log_loss(y_test_merged,y_pred_merged_rf)

Out[762…  2.7338364022626784
```

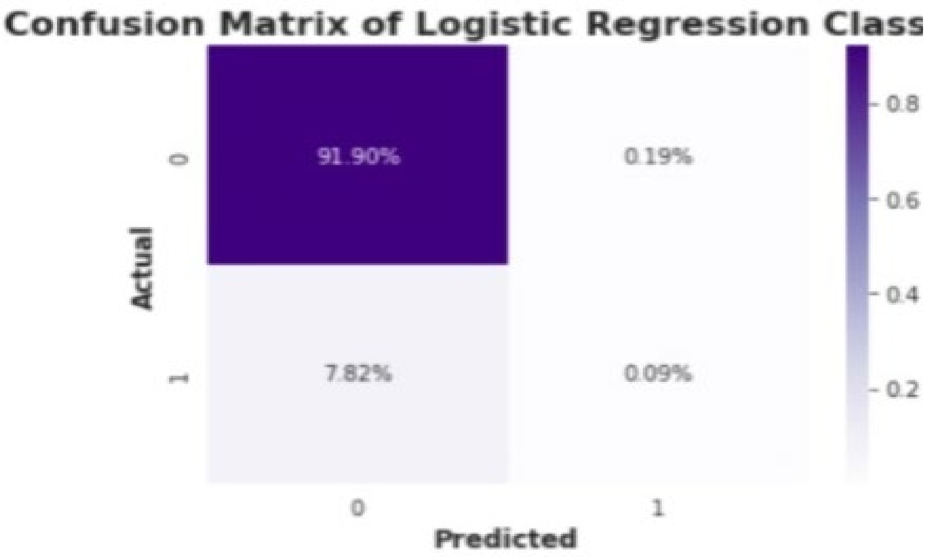
## ROC-AUC Score

```
In [767…  roc_auc_score(y_test_merged, gs_rf.predict_proba(X_test_merged)[:, 1])

Out[767…  0.7377414551192892
```

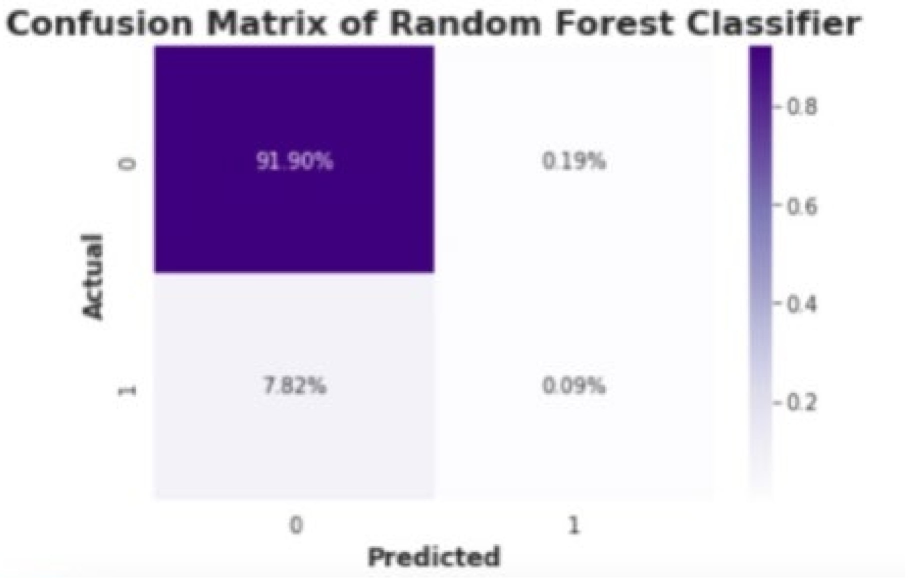
## ROC Curve

```
In [769…  metrics.plot_roc_curve(gs_rf, X_test_merged, y_test_merged)

Out[769…  <sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x7fddefe3b6d0>
```

# Results & Discussion of Results:

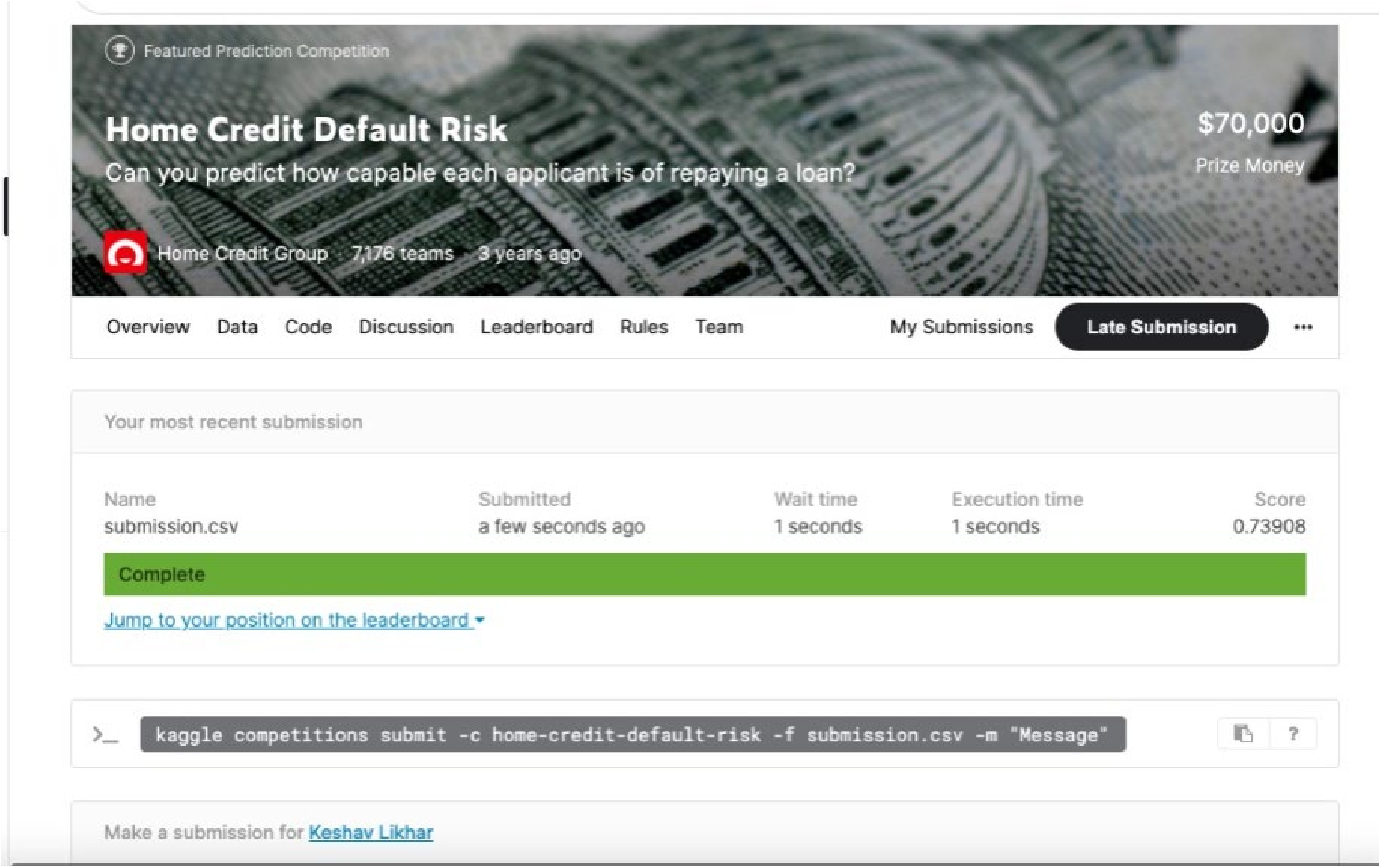


Confusion Matrix of Logistic Regression Class



Confusion Matrix of Random Forest Classifier

# Screenshot of Kaggle Submission:



Home Credit Default Risk

Can you predict how capable each applicant is of repaying a loan?

$70,000
Prize Money

Home Credit Group · 7,176 teams · 3 years ago

Your most recent submission

| Name | Submitted | Wait time | Execution time | Score |
|---|---|---|---|---|
| submission.csv | a few seconds ago | 1 seconds | 1 seconds | 0.73908 |

Complete

Jump to your position on the leaderboard ▾

```
kaggle competitions submit -c home-credit-default-risk -f submission.csv -m "Message"
```

Make a submission for Keshav Likhar

# Conclusion and Next Steps:

- For the Phase II of this Project, we built Logistic Regression and Random Forest Model, performed Hyperparameter Tuning using GridSearchCV  and achieved an accuracy of over 92.1% with Random Forest and 92% with Logistic Regression.
- Initially, we tried merging all the different datasets and created several new columns to understand the relation with target column and gain fruitful insights from the dataset.
- On analysis, we found out that out of the new columns, a few of them were highly correlated with the target variable, thus helping us improve the model predictions.
- We also tried using synthetic sampling techniques like SMOTE and ADASYN to check if there was any change in the model performance.
- Finally, we created a pipeline to perform Logistic Regression and Random Forest Classifier, using Hyperparameter Tuning to gather the best features
- Successively, we ran our model on the best parameters and found out an increase in the model accuracy and submitted the file on Kaggle.

The next steps in our project would be to improve our model performance through implementation of Neural Networks.