Predicting Home Credit Loan Default

Group 3 - Navy Seals

Pritam Vanmore pvanmore@iu.edu
Gautam Ashok gaashok@iu.edu
Sudheer Alluri naalluri@iu.edu
Raman Kahlon rakahlon@iu.edu

Outline

1. Background
2. Data Description & EDA
3. Modeling Pipeline
4. Results
5. Conclusion & Next Steps

Background

Goal: Build a machine learning model which can accurately predict whether the customer defaults on a loan or not.

- This is a supervised classification problem because the target variable takes on one of two values (Default = 1, No default = 0)
- Baseline Model: Logistic Regression (basic and fine-tuned parameters)

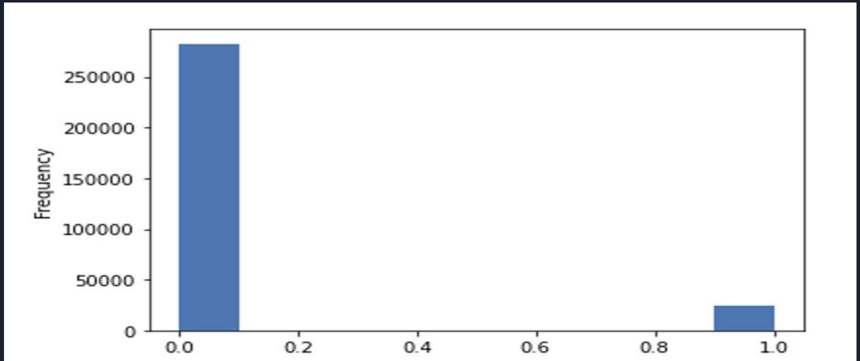
Completed Activities:

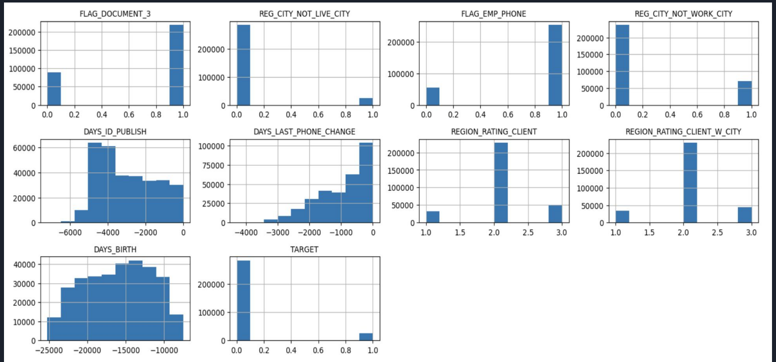
- Examined relationships between main and secondary datasets and data types
- Identified algorithms, pipeline process, and evaluation metrics: accuracy scores, AUC, p-value, RMSE, and MAE.

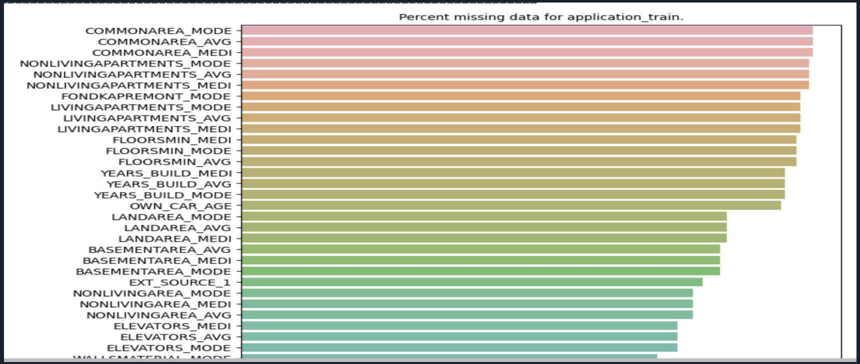
Data Description & EDA

```
Size of each dataset :

dataset application_train        : [    307,511,   124]
dataset application_test         : [     48,744,   123]
dataset bureau                   : [  1,716,428,    17]
dataset bureau_balance           : [ 27,299,925,     3]
dataset credit_card_balance      : [  3,840,312,    23]
dataset installments_payments    : [ 13,605,401,     8]
dataset previous_application     : [  1,670,214,    37]
dataset POS_CASH_balance         : [ 10,001,358,     8]
```







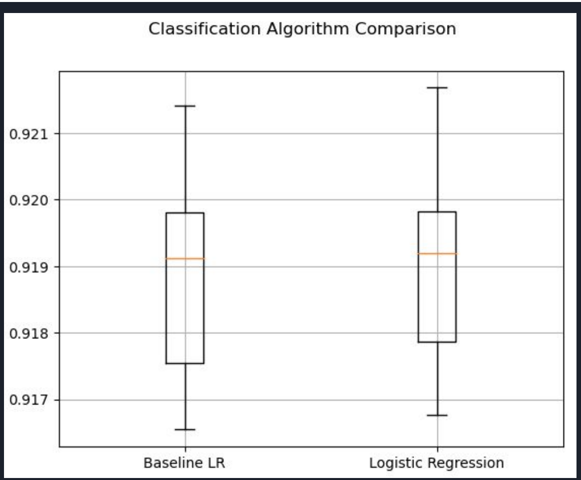
Modeling Pipeline

1. Download data, perform data pre-processing tasks (joining primary and secondary datasets, transformation), and EDA
2. Perform feature engineering activities: imputing missing values, creating new features, and set-up data pipeline with highly correlated numerical and categorical features
3. Create model with data pipeline and baseline model to fit training dataset
4. Evaluate the model using accuracy score, AUC score, p-value, RMSE and MAE for train, validation and test datasets.
5. Perform Grid Search to tune the Logistic Regression model with regularization ('l1', 'l2'), tolerance (0.0001, 0.00001, 0.0000001), and C (10, 1, 0.1, 0.01) hyper parameters and 5-fold cross-validation.
6. Record parameters of best estimator and perform predictions on test data for Kaggle submission.

Results & Discussion

	exp_name	Train Acc	Valid Acc	Test Acc	Train AUC	Valid AUC	Test AUC	P Score	Train RMSE	Valid RMSE	Test RMSE	Train MAE	Valid MAE	Test MAE	Train Time	Test Time	Description
0	Baseline_57_features	0.9189	0.9194	0.9184	0.7538	0.7440	0.7478	0.000	0.262	0.262	0.263	0.137	0.138	0.137	21.7724	0.6446	Baseline LR 57
1	Logistic Regression	0.9188	0.9195	0.9186	0.7509	0.7433	0.7464	0.051	0.262	0.262	0.263	0.138	0.138	0.138	1.6323	0.5853	[["predictor__C", 0.01], ["predictor__penalty"...

- Both baseline and best hyperparameter model performed similarly across all evaluation metrics
- No statistical significance achieved between two experiment (p = 0.051)
- Possibly due to imbalanced distribution of target variables
- Avoided using MAPE due to undefined values caused by division by zero
- Kaggle score (0.74306) and submission:



Name	Submitted	Wait time	Execution time	Score
submission.csv	just now	1 seconds	1 seconds	0.74306

Complete

Conclusions & Next Steps

- Implemented baseline model with Linear Regression and fine tuned parameters

Project Challenges:

- **Memory issues -** needed to increase to 7 CPU's and 11GB memory in Docker to run the model.
- **Outlier Data -** needed to identify highly correlated features due to large amount of missing data and outliers.
- Technical issues on Git versioning
- Faced challenges in implementing pipeline as the results were not showing much difference.

Next Steps:

- Incorporate Log Loss (CXE) calculation into model evaluation
- Design and build additional features from bureau dataset
- Analyze other classification algorithms outlined in our Proposal (NB, SVM, Decision Tree, Random Forest)