Applied Machine Learning
Final Project:
Home Credit Default Risk

Group 16

Aravind Reddy Sheru, Sai Charan Chintala, Seongbo Sim and Yun Joo An

Indiana University

December 2021

Contents

1. Four P's
2. Final Project: Home Credit Default Risk
    1. Feature Selection
    2. Hyperparameter Tuning
    3. Results and Discussion
3. Conclusion and next steps

Four P's

- Past
  - We started the HCDR Project, which predict whether borrowers are defaulters or not based on various financial and nonfinancial data.
  - In the first phase, we collected the data, did EDA, and established the baseline model using logistic regression.
  - The baseline model performed well with high accuracy on the test data.
  - To improve the result, we tried to balance the data making the number of defaulters and non-defaulters even, but it was not successful.
- Present
  - First, we selected more relevant features, and filled missing values of them by imputation.
  - To find the best model, we implemented new models of decision tree, lasso regression, ridge regression and XGBoost model.
  - We tuned the hyperparameters of the models, and found the best parameters using Grid Search.
  - In the steps above, we used the appropriate metrics for the each model to evaluate the models.

Cont'd

- Planned
  - In Phase 3, we will implement a deep learning model.
  - We will build a multi-layer perception model in PyTorch for loan default classification.
  - As a stretch goal, we will develop and implement and new multitask loss function in PyTorch.
  - Finally, we will submit our results on Kaggle, and report our scores and rank.
- Problems
  - With the feature selection and imputation step, we could not improve the test accuracy or AUC of baseline model.
  - Unlike regression models, we could not develop a classification model better than the baseline model.

Feature Selection

- Feature selection and imputation
  - We discarded features with missing values more than 30%, and dropped irrelevant features.
  - We filled the missing values of selected features.
  - e.g. CNT_SOCIAL_CIRCLE with 0 and CNT_FAM_MEMBERS with median.
- Adding more relevant data
  - We decided to add relevant features based on our prior knowledge.
  - e.g. AMT_CREDIT_TO_ANNUITY RATIO, Salary_to_credit.

Hyperparameter Tuning

- To find the best model, we train and evaluate several models.
  - Our models are "Logistic Regression", "Decision Tree Model", "Lasso Regression", "Ridge Regression", and "XGBoost".
  - We use different evaluation metrics to have a concrete evaluation of candidate models including "Accuracy", "AUC", and "negative MSE".
- Also, we tuned the hyperparameters with the help of "GridSearchCV".
  - For the "Logistic Regression" model, we used different C parameters to control the penalty strength.
  - For the "Decision Tree Model", we used different maximum depth and number of samples split.
  - For the "Lasso Regression", and "Ridge Regression", we used different α parameters to control the weighting of the penalty to the loss function.
  - For the "XGBoost", we used different maximum depth and the number of trees.

Results and Discussion

- Results for baseline model

Model	Cross fold train accuracy	Test accuracy	Test MSE	AUC
Baseline (Logistic Regression)	91.9	92.0	-	0.504
Baseline with 79 inputs	91.9	91.9	-	0.506
Gridsearch Baseline with 79 inputs	91.9	92.0	-	0.500
Decision Tree Model	-	-	7.56	0.739
Lasso Regression	-	-	6.87	0.756
Ridge Regression	-	-	6.87	0.757
XGBoost	-	-	6.87	0.757

  - Classification models built in Phase 2 could not win the baseline model.
  - Among regression models, "Ridge Regression" model shows the best performance.

Conclusion and Next Steps

- In Phase 2,
  - We estimated several models including both classification and regression models.
  - Also, we conducted feature selection, data imputation, and hyperparameter tuning.
- In Phase 3,
  - We will implement a deep learning model.
  - We will build additional models in PyTorch.
  - Finally, we will submit our results on Kaggle, and report our scores and rank.