Prediction on Lending Default

GROUP 22   Jinchen Gong, Zhuoran Yu, Chixu Ni, Jian Zhou, Zeyi Huang

Menu

- Introduction
- Data preprocessing
- Model building
- Comparison with ROS
- Conclusion

Introduction

In recent years, financial institutions have been required to provide more diverse credit services to new customer segments. As one of the main providers of credit, banks have high risk control requirements. However, it is sometimes difficult for banks to manage risk control for new customer segments. This hinders the implementation of financial inclusion. How to use the bank's existing data to serve new scenes and segment new customer groups becomes a valuable research direction.

Data Preprocessing

Dataset Basics:

- 10000 records, 38 raw features (csv columns)
- Imbalanced Labels: negative:positive=4:1 (positive isDefault=1)
- Split into train, validation, and test dataset

Data Preprocessing

Remove unnecessary features:

- Features without meaningful information: loan_id, user_id, …
- Low-variance features: policy_code, app_code, …

Merging and Split Features:

- Merge similar features by summing up: f = f_0 + f_1 + … + f_5
- Split Features: loan_date into loan_year and loan_month
  - Otherwise, so many unique dates
  - We also drop other features with too many unique values

Data Preprocessing

Finalizing Feature Engineering:

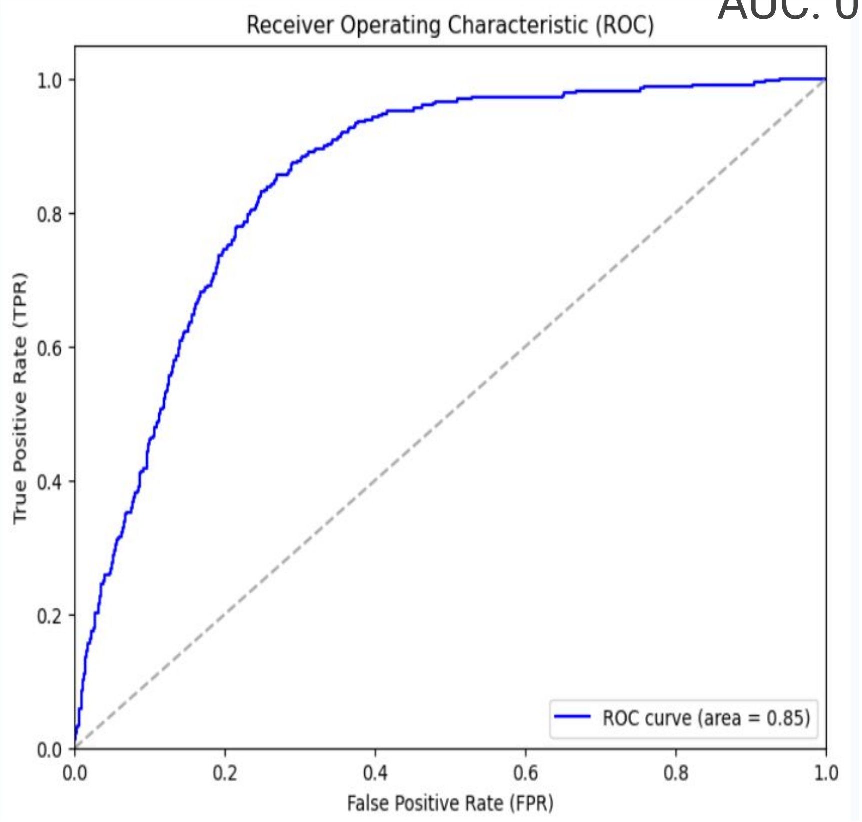
- Remove redundant features with similar semantics
  - E.g, keep postal_code and remove region_code
- Convert numeric features into categorical (if necessary)
  - E.g, postal_code is better to be used as categories other than large numbers
  - Also for boolean features (0 and 1), loaded as integers by default
- End up with 29 features
- Finalize by converting categories with one-hot encoding

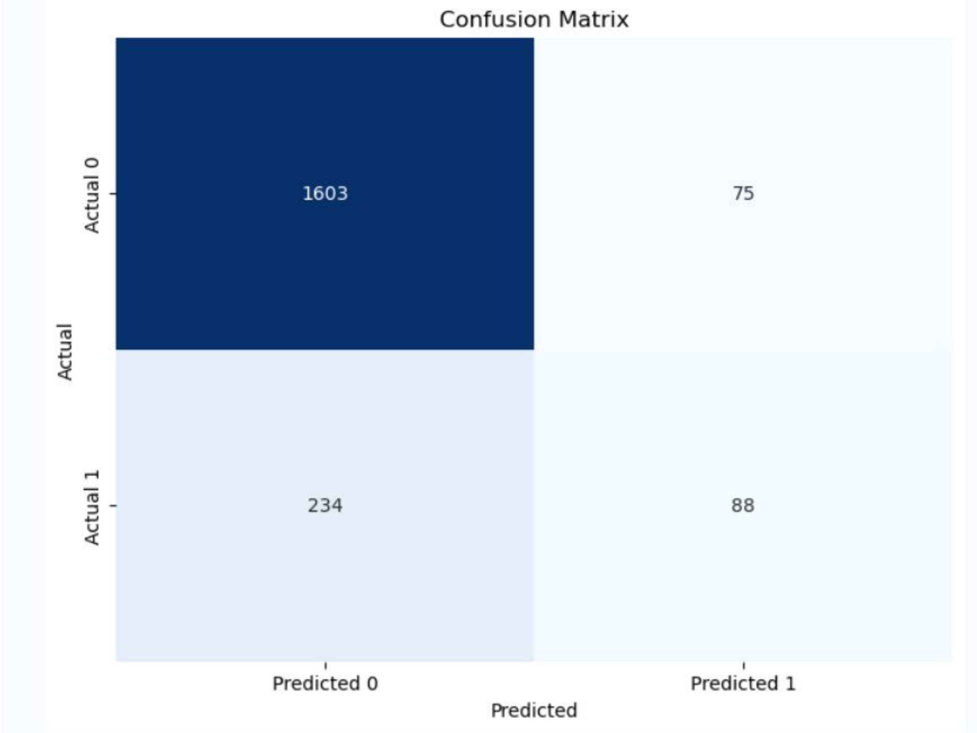
Model building

- Logistic regression
- KNN
- XGboost

Logistic regression

AUC: 0.846



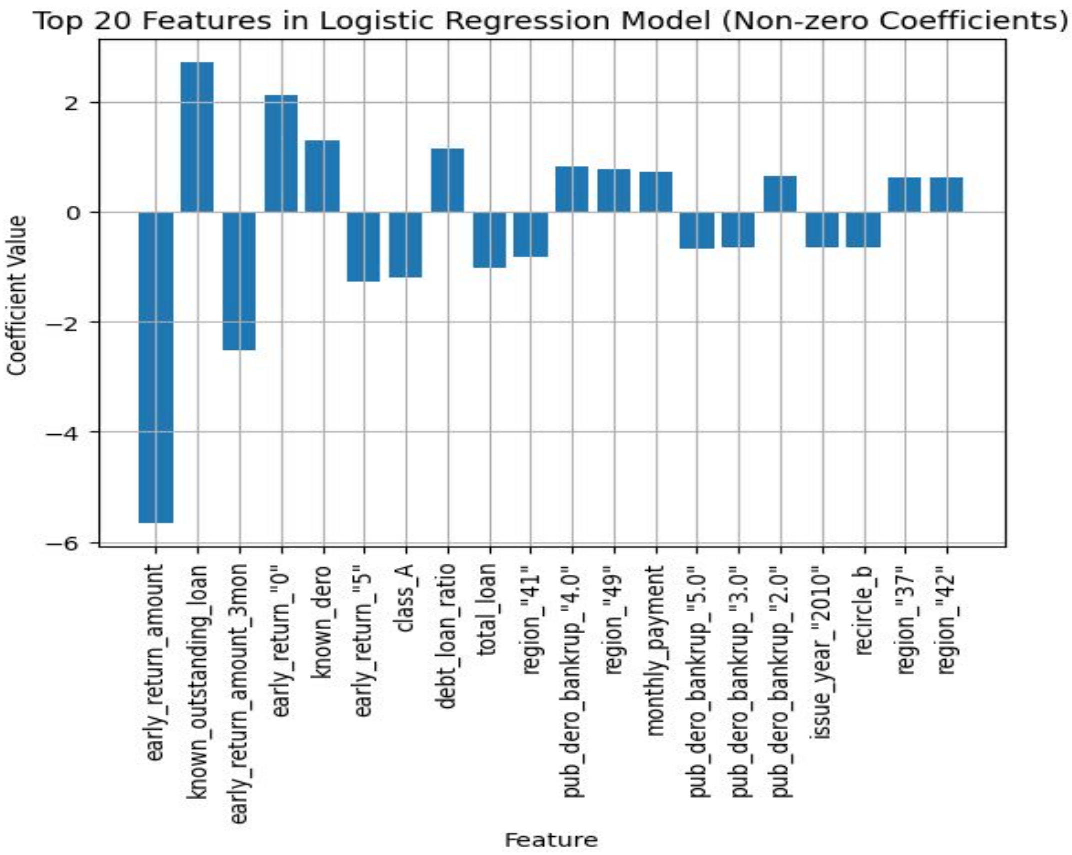


Confusion Matrix

Actual	Predicted 0	Predicted 1
Actual 0	1603	75
Actual 1	234	88

Actual
Predicted

Logistic regression



Top important features:

Early return amount,

Known outstanding amount,

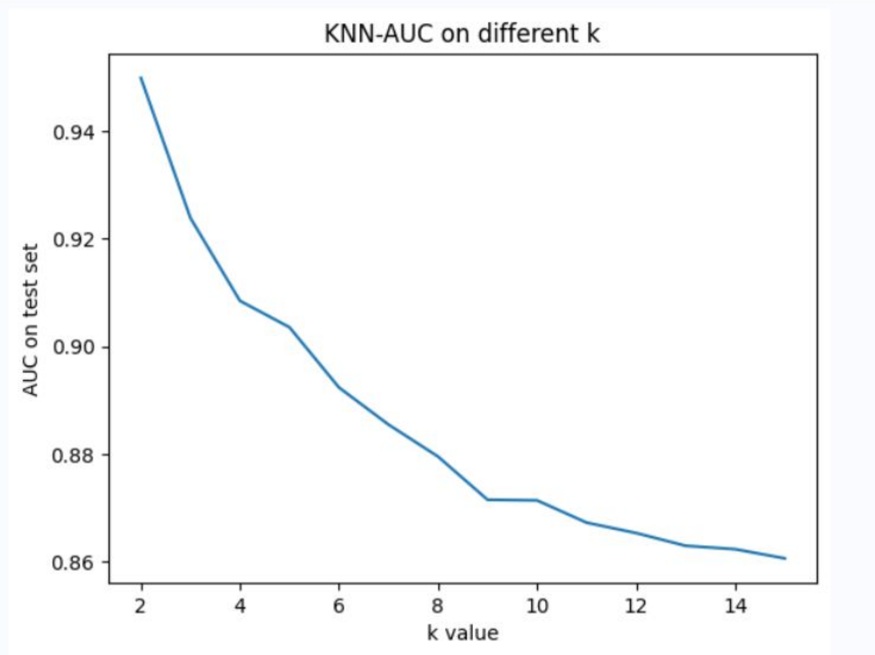
Early return amount in 3 months,

Early return time,

Known dero

KNN

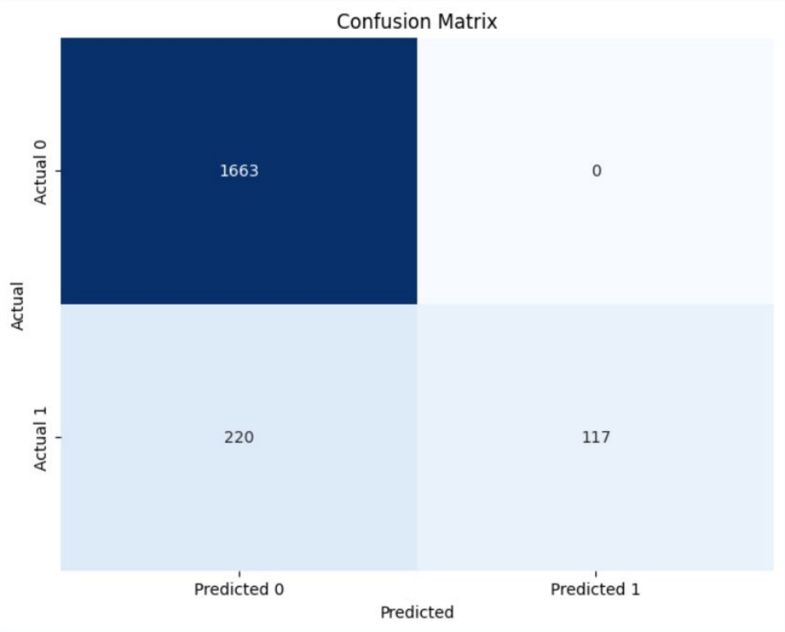
Grid search



AUC decreases when k increases

K = 2 owns highest AUC

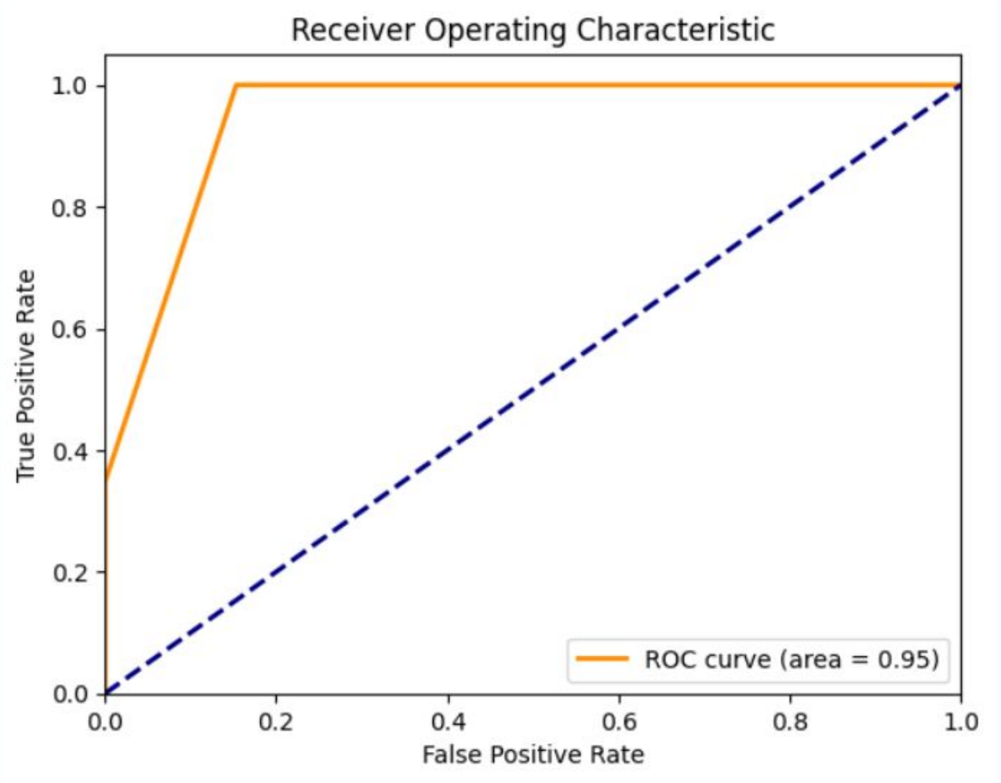
KNN



Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	1663	0
Actual 1	220	117

Actual
Predicted



AUC:0.95, which is a quite high number

XGBoost

Object function: $\text{Obj} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$

A combination of loss function and the regularization term, penalizing the complexity of the model

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$$

Where T is the number of trees, w is the weight of leaf nodes, γ and λ are regularization parameters.

Each time adds a new decision tree to decrease object function.

XGBoost makes prediction through the aggregation of these base decision tree, as the weighted sum of tree’s prediction and its contribution to the decrease of object function.

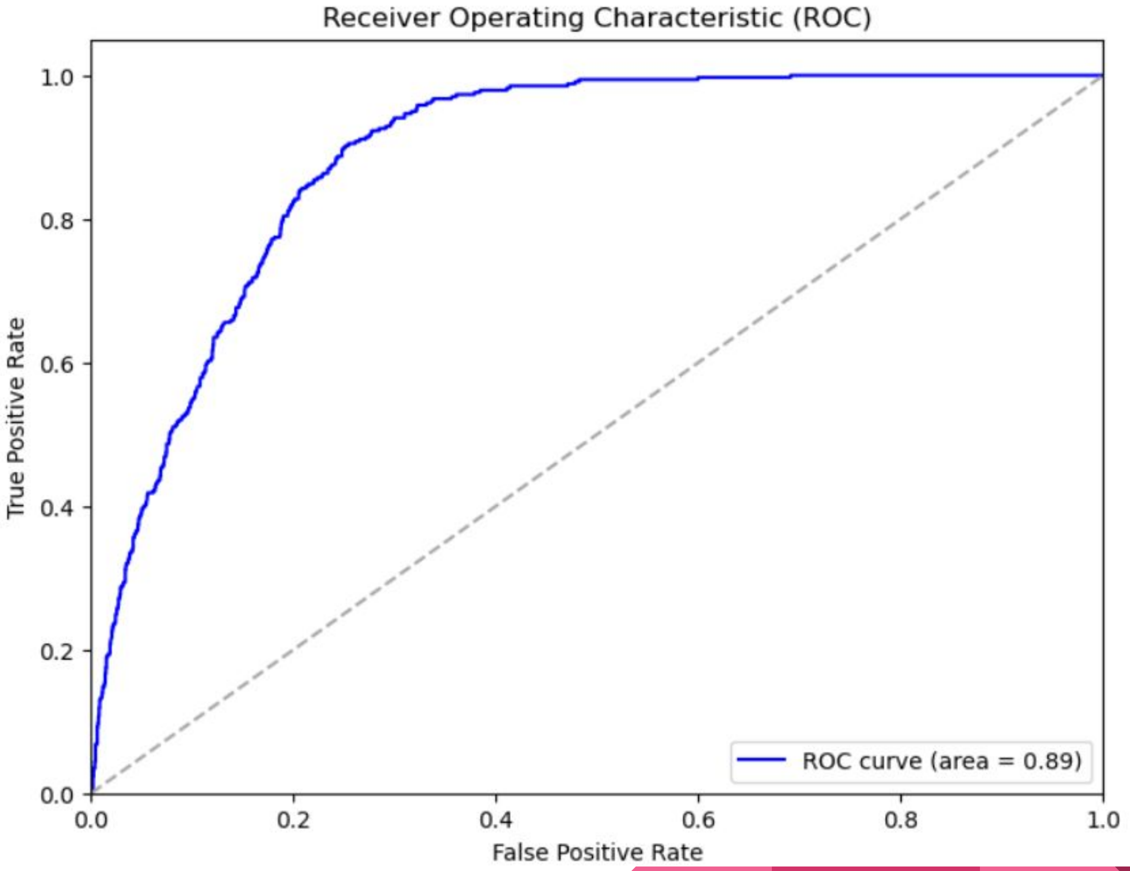
XGBoost

Apply grid search inside train set towards learning rate η and trees' number *n_estimators*

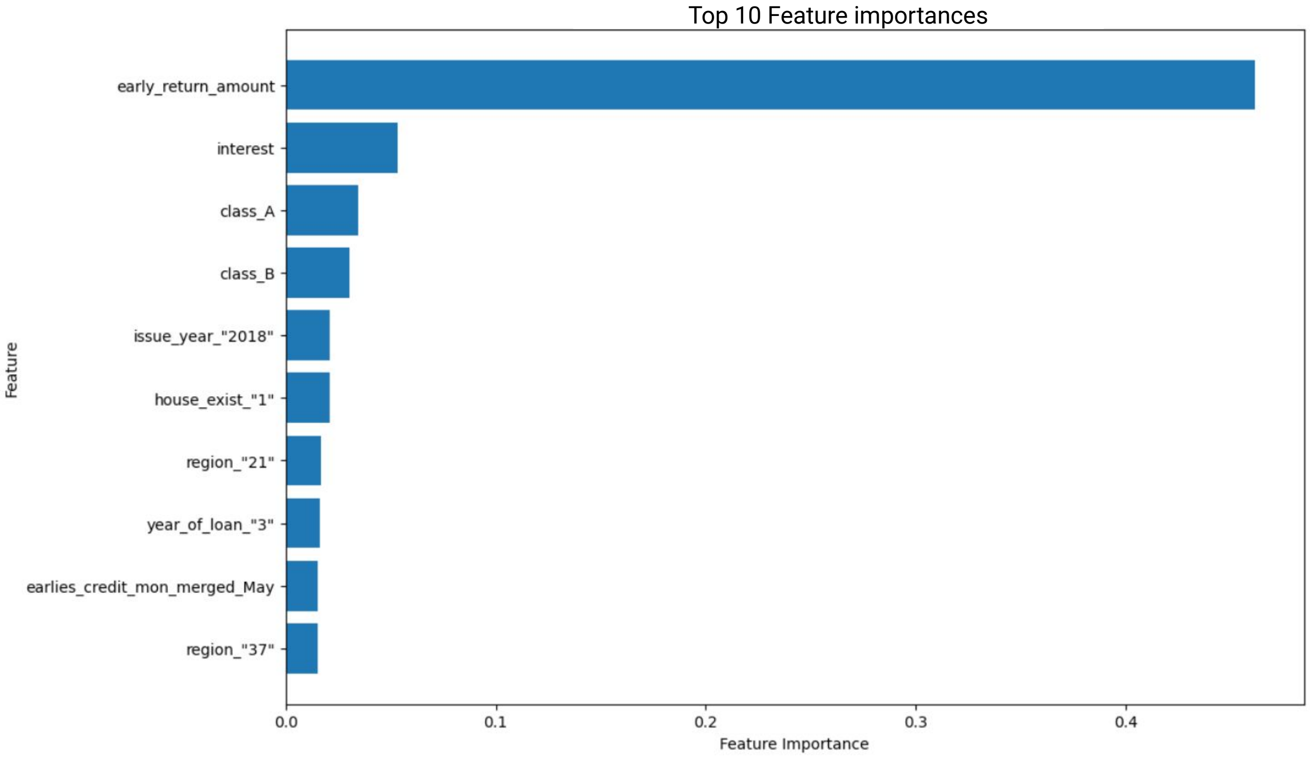
Best estimator: η=0.02 and *n_estimators*=200

Predict on test set

AUC:0.8876



XGBoost



Top important features:

Early return amount,

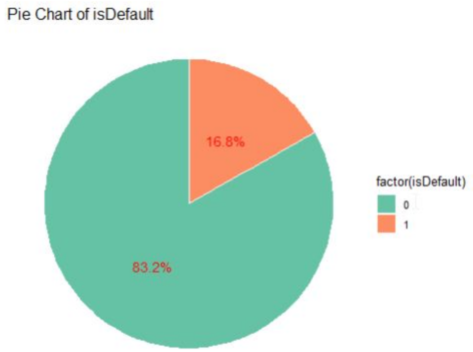
Interest,

Class,

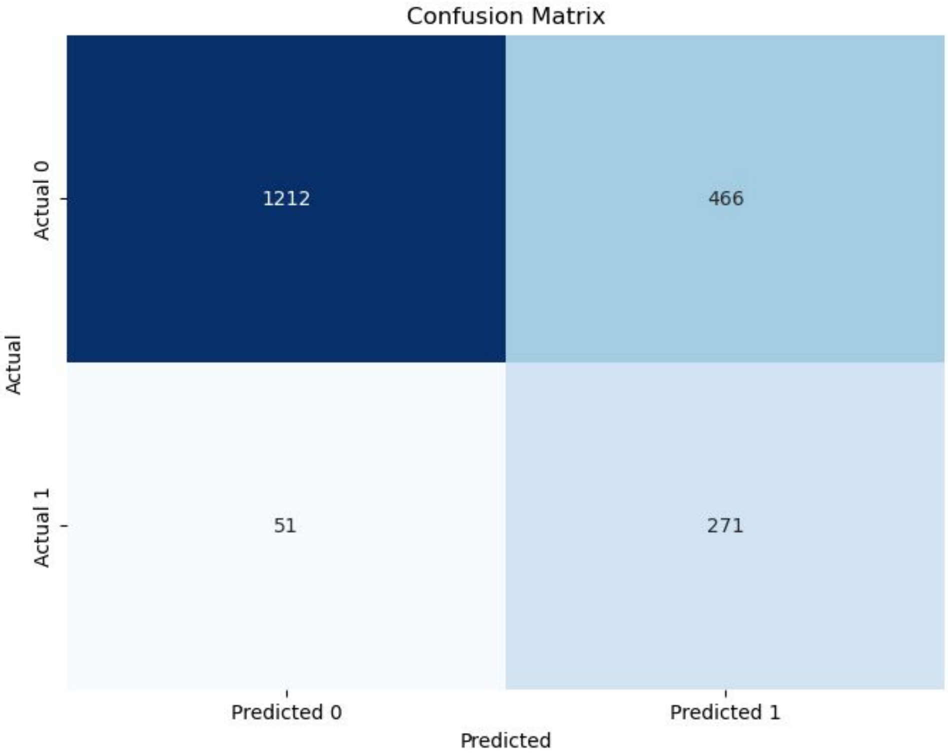
House exist

ROS sampling

- What is ROS sampling?
  - Method used to oversample data by duplicating point for the minority data set.
- Why are we interested in ROS sampling?
  - Our overall dataset contains a significant imbalance in the data and we see a significant difference in performance between predicting default vs predicting not default.
- Drawbacks of ROS
  - There is a significant potential for the model to overfit due to the duplicate data that is being represented in the train runs.



ROS Results (Logistic Regression)

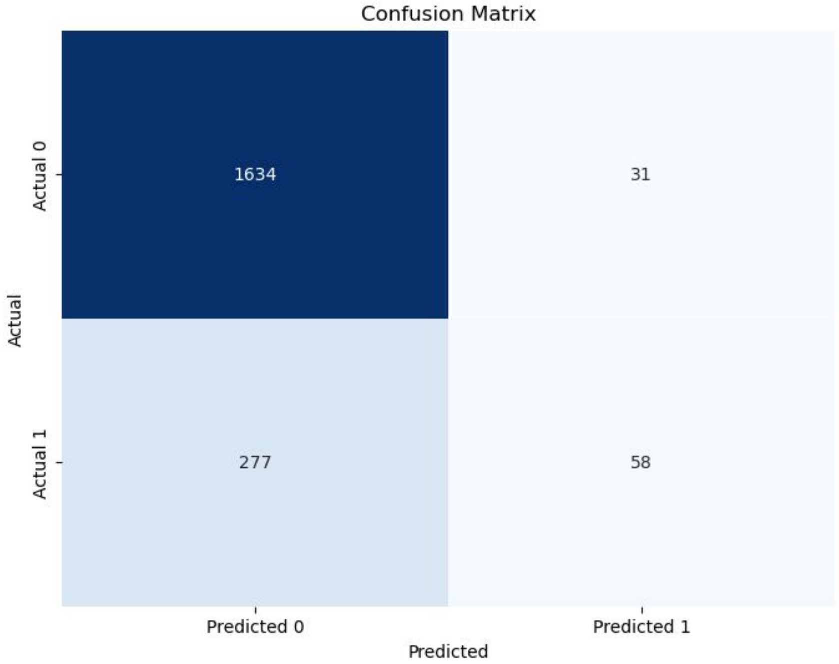


Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	1212	466
Actual 1	51	271

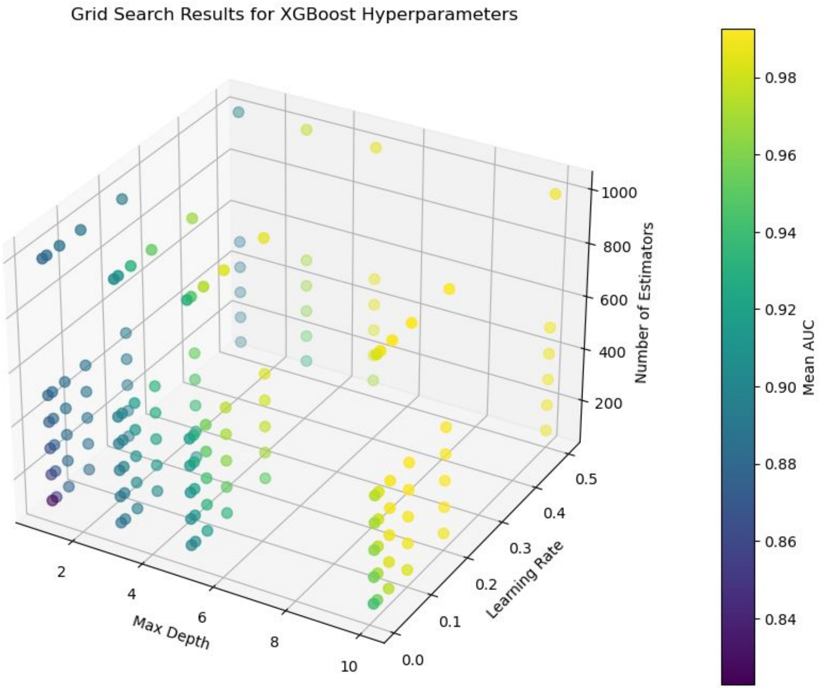
- Saw significantly stronger auc and recall score, but saw a decline in it’s precision
- Saw a significant increase in predictions of default compared to other ROS run changes which was what we hoped, but the accuracy left a lot to be desired.

ROS Results (KNN)



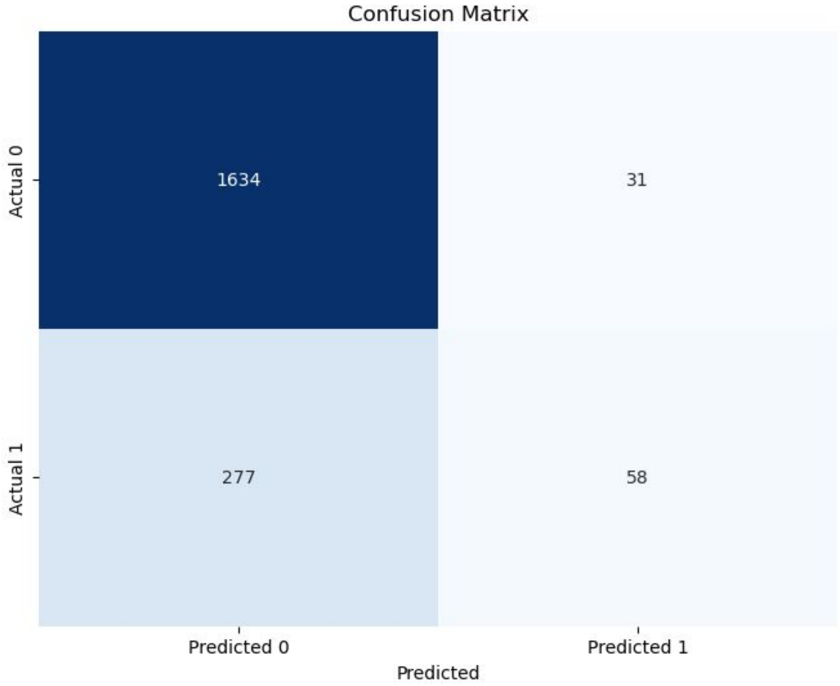
- Underperformed initial model
- Did not see significant difference compared to the first model when comparing the confusion matrix
- Overall no notable change is observed for KNN

ROS results (XGBoost)



- Through our grid search runs, there is clear signs of overfitting when looking at the train data results as the ROS on the boost models saw upwards of .99 AUC score but underperformed severely on the test set.

ROS results (XGBoost)



- But there is a very clear decline in accuracy when looking at XGBoost with ROS.
- When running these models I expected better performance when predicting default due to the increase in data.
- Instead of doing oversampling, maybe doing undersampling would work better.

Conclusion

Take-away:

- The KNN model achieved the best results, reaching an AUC of 0.95. The XGboost model was the next best, with an AUC of 0.89, while the Logistic Regression model ranked last with an AUC of 0.85.
- In terms of speed, the KNN model runs the fastest, Logistic Regression holds the middle ground, and XGBoost is significantly slower than the other two.
- Early return amount is most important both in regression and xgboost.

Future Work:

- Further explore other machine learning approaches that may yield better results.