DATA 1030: Midterm Project Report

Yangyin Ke, Brown University

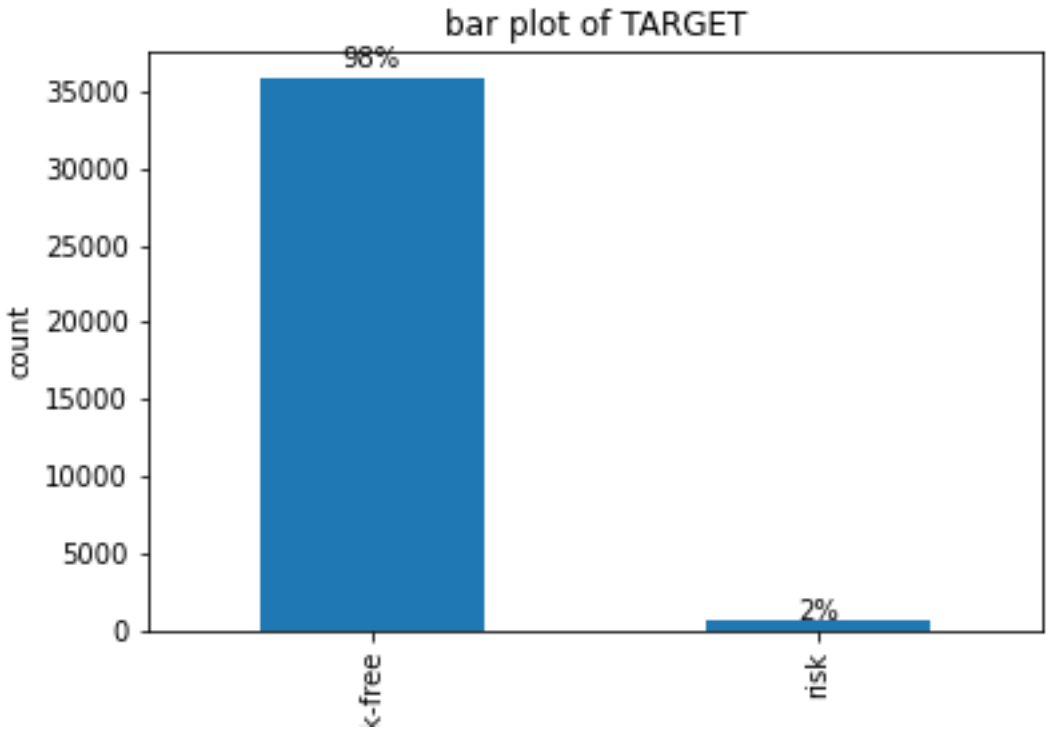
Github Link: https://github.com/yangyinke/1030project

Introduction:

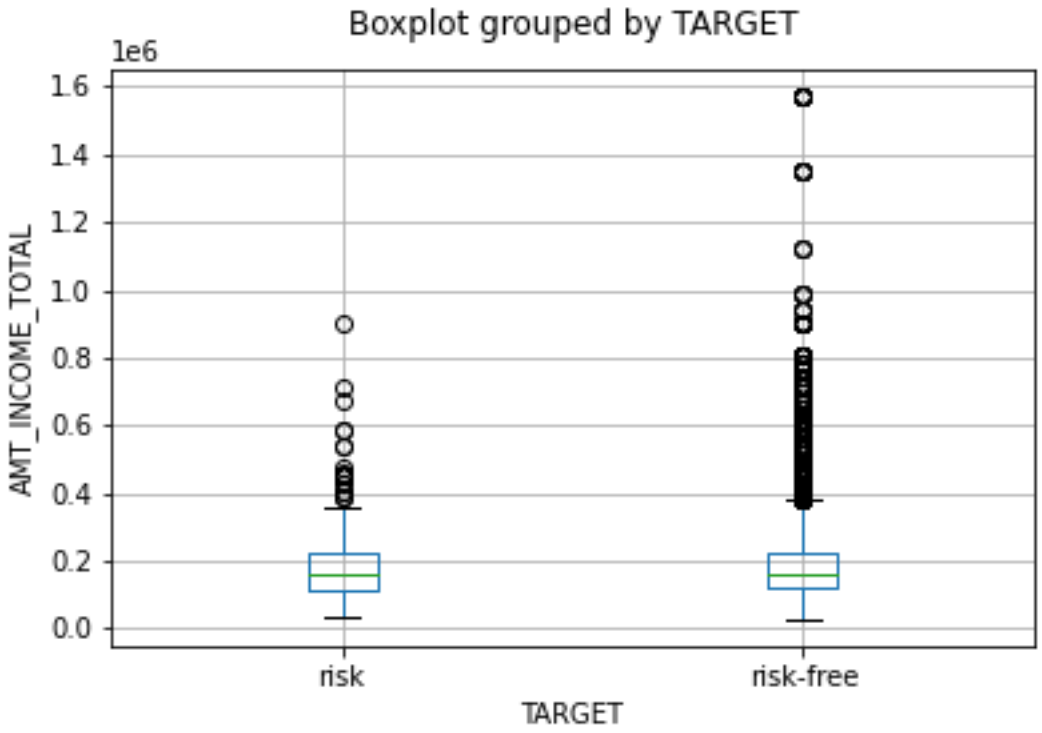
The dataset used in this project is a credit card client information dataset, which includes clients' personal information as well as their payment status (paid off, no loan or overdue). It is derived from combing application record dataset with credit record dataset. The problem to be solved in this project is to predict whether a client is a risk user given his/her personal information. Therefore, the target variable in this project is whether a client is a risk user or not. It is a classification problem because it classifies clients into two groups: risk user and risk-free user. It is an interesting problem because it can be applied to credit card approval process. Credit card institution can predict the credit status of applicants given their personal information and avoid approving the application from risk users.

After inner merging two datasets in this project, the number of data points is 36457 and there are 20 features in total. The dataset used in this project is from Kaggle and is well-documented. One related public project on Kaggle completed by Ramgopal (2020) is to predict whether a client is a good client or a bad client. He considers those overdue for more than 120 days as bad clients while the rest of them as good clients. The prediction is based on clients' personal information. The author finds that the proportion of bad clients jumps significantly in clients with accounts opened over 50 months ago. Therefore, he excluded those data points. Ramgopal used XGBoost model in his training and he says it gave him a good accuracy level. Another project on Kaggle is done by Xiao (2020). He used this dataset to explore credit card approval. Xiao applied several algorithms separately to do the prediction and his random forest algorithm results in best accuracy of 94.88%. He says, however, optimized algorithms like XGBoost and LightGBM are also worth exploring.

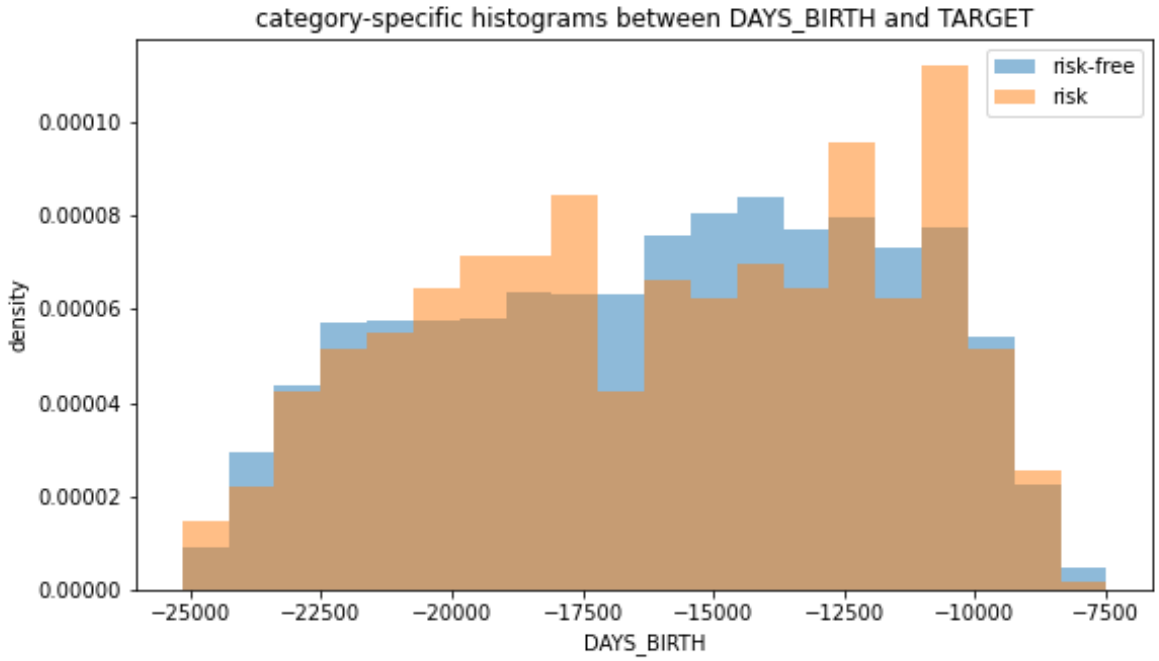
Exploratory Data Analysis



X axis is the risk classification value and y axis represents the number of data points in each status. It shows the distribution of data points under different risk condition. It is obvious that the dataset is imbalanced with 98% of points in risk-free status and only 2% are risk users. Therefore, a stratified strategy should be used when splitting the dataset.



X axis is the risk status and y axis is the amount of total annual income. It shows the distribution of annual income of clients with different risk identifications. It can be concluded that even though there is not much difference between the average annual income of these two classes, risk-free clients are more likely to gain higher annual income.



X axis represents the days of birth counting back from current and y axis is the corresponding population density. Histogram with orange color represents the distribution of risk users' days of birth while blue color represents risk-free users'. The distribution of days of birth is similar in both risk classes, which means that days of birth is likely to be irrelevant to risk status of clients.

Methods

The machine learning pipeline in this project includes exploratory data analysis (EDA), balancing dataset, dataset split, data preprocessing, model training. Particularly for model training, with five machine learning algorithms involved, their parameters were tuned to see which model with which set of parameters gives rise to the best test score. The test scores here are mostly calculated by cross-validation.

Balancing Dataset

The original dataset is so imbalanced that models trained on it can hardly perform well. Even though the parameter "class_weight" of models was tuned and GridSearchCV used "f1_weighted" as scoring, the outcome models get f1_scores less than 0.2. Therefore, balancing the dataset becomes necessary in this case. SMOTE (Synthetic Minority Over-sampling Technique) is used in this project to balance dataset, with new risk users synthesized from original dataset. The output balanced dataset has nearly half risk users and half risk-free users, and it will be splitted to train models in this project.

Splitting Strategy

The dataset in this project is splitted into three parts, including 60% for training, 20% for testing and 20% for validation, which is a safe choice for majority datasets. Stratified splitting and stratified k-fold splitting are used in this project. Considering that the original dataset is significantly imbalanced, with less than 2% of data points as risk users, it is important to ensure each splitted set contains a specific portion of risk users. In this way, it can best mimic future use when deploying the model. Stratified k-fold is used here to do cross validation, so that the test scores can be more reliable.

Data Preprocessing

The dataset is IID, with each row representing a distinct client. There is no correlation between rows. Even though some IDs present more than once in the dataset, they stand for different clients with completely irrelevant information. Therefore, there is no group structure in the dataset, and it is not a time series data.

Preprocessor choices are clarified below:

- One-hot features:
    - Columns: CODE_GENDER, NAME_INCOME_TYPE, NAME_FAMILY_STATUS, NAME_HOUSING_TYPE, OCCUPATION_TYPE
    - Reason: These features are all categorical features which cannot be clearly ordered.
- Ordinal features:
    - Columns: FLAG_OWN_CAR, FLAG_OWN_REALTY, NAME_EDUCATION_TYPE
    - Reason: These features are all clearly ordered categorical features.
- Standard features:
    - Columns: CNT_CHILDREN, AMT_INCOME_TOTAL, DAYS_BIRTH, DAYS_EMPLOYED, CNT_FAM_MEMBERS, MONTHS_BALANCE
    - Reason: These features are all numerical features without boundaries.

46 features are included after performing the Encoders mentioned above.

Machine Learning Algorithms

This project focus on classification problem. Therefore, classification machine learning models, including logistic regression, random forest, k nearest neighbors, xgboost and support vector machine, were attempted.

In the process of training logistic regression model, C, representing regularization strength, was tuned to avoid over-fitting with 10, 100, 1000.

In the process of training random forest model, max_depth (maximum depth of the tree) was tuned with 40, 60, 80. Also, max_features (maximum fraction of features considered at each split) was tuned with 0.3, 0.5, 0.7. N_estimators (number of trees in the forest) was tuned with 50, 60, 70.

In the process of training k nearest neighbor model, n_neighbors (number of neighbors to use) was tuned with 1, 3. Weights (weight function) was tuned with ‘uniform’ and ‘distance’.

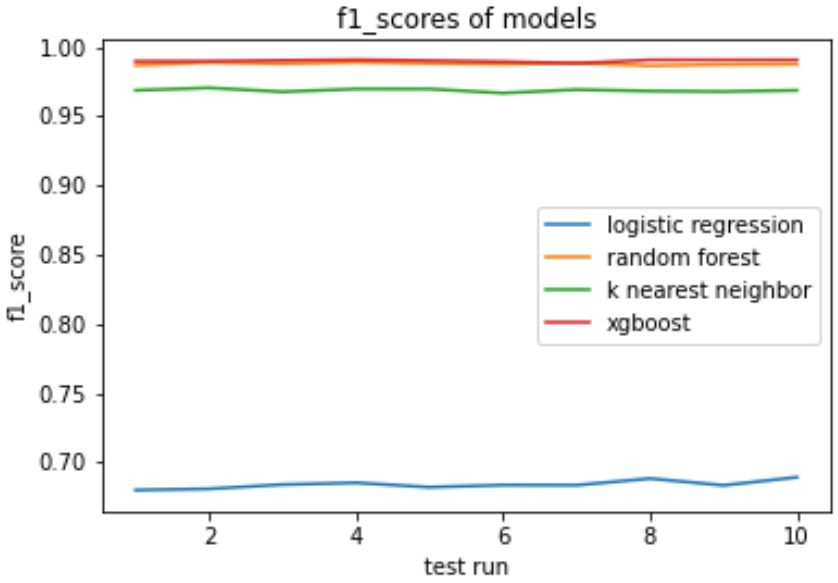
In the process of training xgboost model, n_estimators (number of trees used) was tuned with 1000 and 10000.

In the process of training support vector machine, it takes too long to finish even just one model training. Therefore, this model is not applicable in this project.

Evaluation Metric

In this project, f1_score is used as the metric to evaluate the performance of models. Considering that the purpose of this project is to predict the credit status of applicants, prediction regarding risk users (positive samples) should be more careful. Therefore, fbeta_score providing weighted sum of precision and recall is preferred in this case. Credit card institution will not be happy with predicting a conditionally risk user to be a risk-free user, because these users cannot pay off their credit card balance. Similarly, credit card institution do not want to lose clients due to predicting conditionally risk-free users as risk users by mistake. Consequently, f1_score, assigning same weights to precision and recall, should be used here.

Uncertainties



X axis shows the number of test run and Y axis stands for the f1_score at each run. This plot shows the performance of different models. Three non-deterministic algorithms were used in this project: logistic regression, random forest and xgboost. All are stable with low standard deviation in test scores. In general, random forest and xgboost can better handle uncertainties with test score standard deviation lower than 0.0008.

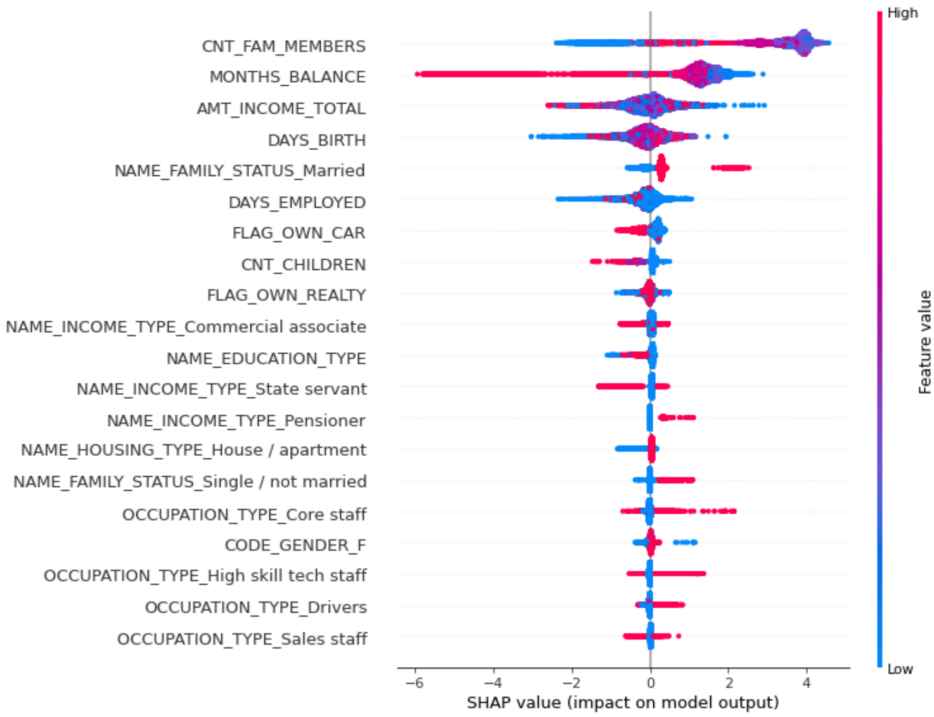
Results

Best Model

Xgboost model works best in this project with highest and most stable f1_score. The mean of its f1_score is 0.99020 and f1_score standard deviation is 0.00074.

The baseline model would be predicting all clients as risk users. Since the model was trained based on a balanced dataset with half risk-free user and half risk user, the baseline accuracy should be 0.50000 and f1_score is 0.66667. Therefore, the best xgboost model gets an f1_score that is 435.36586 standard deviations above the baseline f1_score.

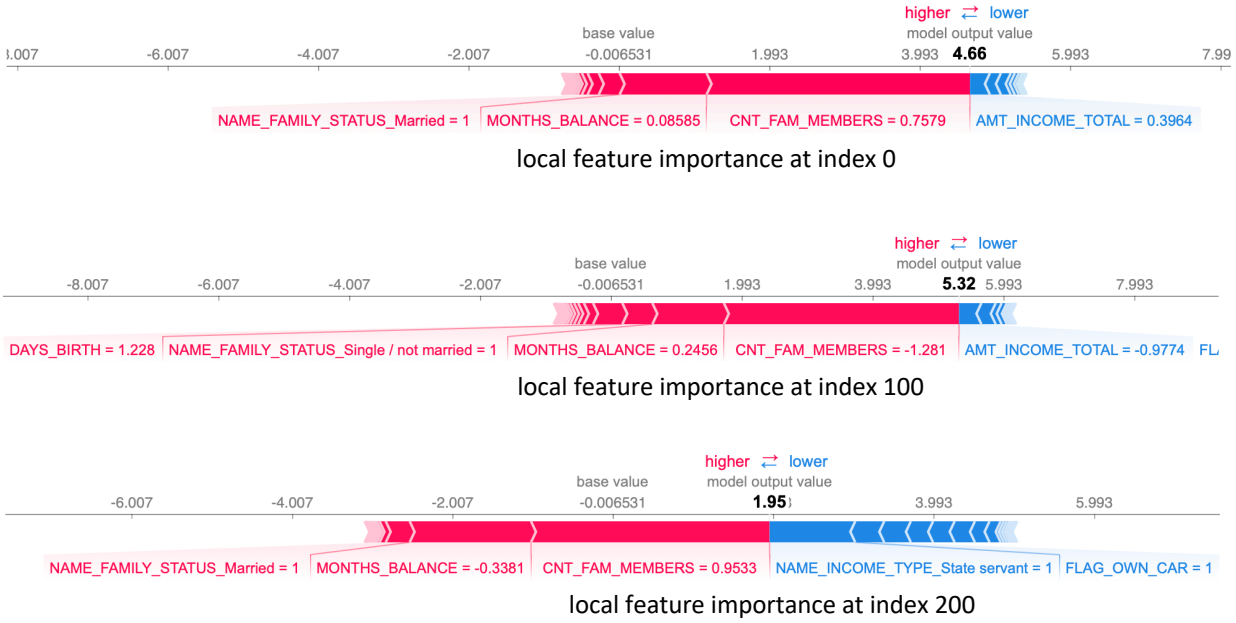
Global Feature Importance



X axis is the SHAP value and y axis stands for feature names. This plot shows the global SHAP values for each feature in the testing set. In terms of global feature importance, count of family members is the most important feature. It is reasonable because more family members always comes with heavier financial burden. Therefore, clients with more family members are more likely to be risk users. The least important feature globally is occupation type It is unexpected because some occupation types in the dataset like “High skill tech staff” and “Core staff” are always associated with high income, which means people working this type of jobs are

more likely to be risk-free. However, this is not the truth here. It is possible that due to 30% of data missing in this column, such feature is not that helpful in prediction.

Local Feature Importance



X axis stands for feature names and their lengths and colors represent their impacts on prediction. In general, these plots explain the importance of each feature in prediction at index 0, 100, 200. Like global feature importance, count of family members is the most significant feature and type of occupation is the least significant feature.

Outlook

The weak spot of this modeling approach is balancing the imbalanced dataset. Considering that the incoming dataset in real life is still imbalanced, even though the model performs well on balanced dataset, it might be hard to be deployed on incoming data. Therefore, the model can be improved by training on an imbalanced dataset with some other smart ways. With 30% of datapoints missing in occupation type, the additional technique I could have used should be reduced_feature_xgboost, which is an advanced tree-based algorithm capable of handling missing values. Alternatively, collecting additional data on clients' occupation type can also help improve model performance.

References

Ramgopal P. (2020). Credit Card Approval Model using XGBoost. Retrieved from https://www.kaggle.com/ramgprajapat/credit-card-approval-model-using-xgboost

Xiao S. (2020). Credit Card Approval Prediction Using ML. Retrieved from https://www.kaggle.com/rikdifos/credit-card-approval-prediction-using-ml

Xiao S. (2020). A Credit Card Dataset for Machine Learning. Retrieved from https://www.kaggle.com/rikdifos/credit-card-approval-prediction