Loan Default Predictor

https://github.com/yxing12/loan-default-predictor

Yunfei (Cynthia) Xing
Data Science Institute

December 6, 2023


01
Project
Recap

Recap

Problem

Loan Default - situation where a borrower fails to repay a loan according to the terms agreed upon.

Negatively impacts the borrower (credit rating) and the financial institution (revenue, reputation).

Leverage ML techniques to predict loan defaults.

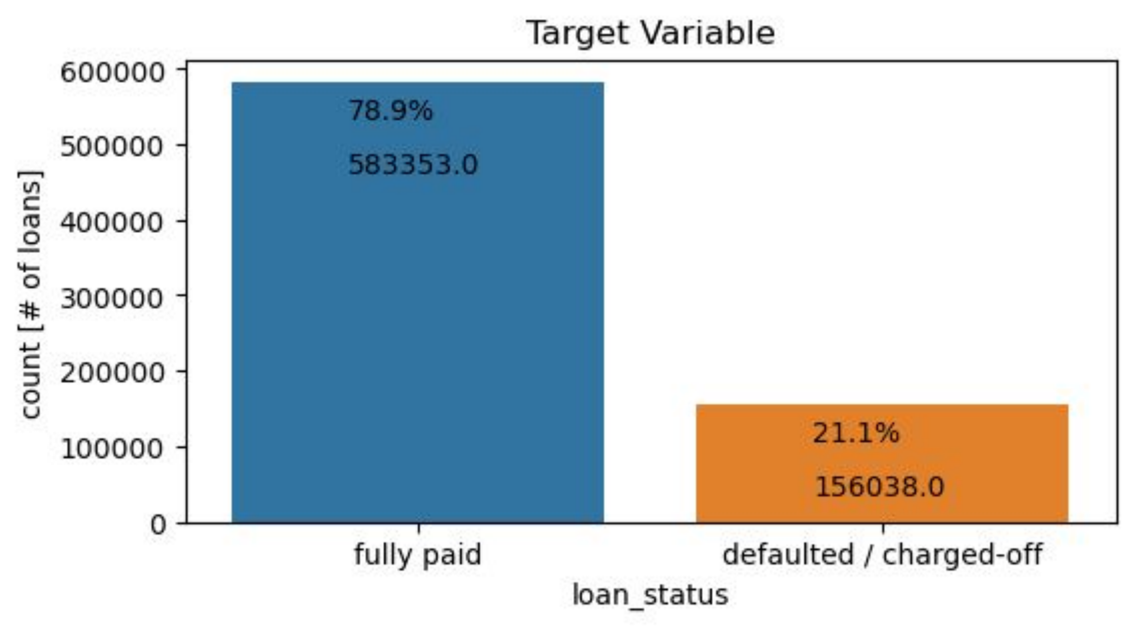
Dataset

- Lending Club's loan data from 2007-2017 adopted from Kaggle
  - Aggregated data from Lending Club website
- Cleaned Dataset shape: (739,391, 32)
- I.I.D: Yes
- **Classification Problem** - categorize the loan into one of the two class (fully paid/defaulted) based on loan specs and borrower financials

Recap - Target Variable

- ~80% loans fully paid ~20% charged off / defaulted

  → **unbalanced classification problem**



* A charge-off is a debt that a creditor has given up trying to collect on after borrower have missed payments for several months.

Recap - Splitting

Basic train_test_split

- Large i.i.d dataset (>700k rows)
- **98-1-1 train-val-test split**

```
train + val shape shape:  (731997, 31) (731997,)
--------------------------
test shape:  (7394, 31) (7394,)

y_other Distribution: Fully-Paid: 78.90%, Defaulted/Charged-off: 21.10%
y_test Distribution: Fully-Paid: 78.90%, Defaulted/Charged-off: 21.10%
```

Preprocessing

- OneHotEncoder
  - categorical
- OrdinalEncoder
  - ordinal
- MinMaxScaler
- StandardScaler
  - continuous

```
# collect the features
cat_ftrs = ['home_ownership','verification_status','purpose',\
            'addr_state','initial_list_status','application_type']
ordinal_ftrs = ['grade','sub_grade']
ordinal_cats = [['A','B' ,'C','D','E','F','G'],
                ['A1', 'A2', 'A3', 'A4', 'A5', \
                 'B1', 'B2', 'B3', 'B4', 'B5', \
                 'C1', 'C2', 'C3', 'C4', 'C5', \
                 'D1', 'D2', 'D3', 'D4', 'D5', \
                 'E1', 'E2', 'E3', 'E4', 'E5', 'F1', 'F2', 'F3', 'F4', 'F5', \
                 'G1', 'G2', 'G3', 'G4', 'G5']]
max_min_ftrs =['loan_amnt','int_rate','installment','fico_score']
std_num_ftrs = ['term', 'dti', 'earliest_cr_line', 'open_acc', 'pub_rec', \
                'revol_util', 'total_pymnt', 'total_rec_int', 'last_pymnt_amnt', \
                'acc_open_past_24mths', 'avg_cur_bal','bc_open_to_buy', 'bc_util', \
                'mo_sin_old_rev_tl_op','mo_sin_rcnt_rev_tl_op', 'mort_acc', \
                    'num_actv_rev_tl', 'pub_rec_bankruptcies', 'log_annual_inc']
```

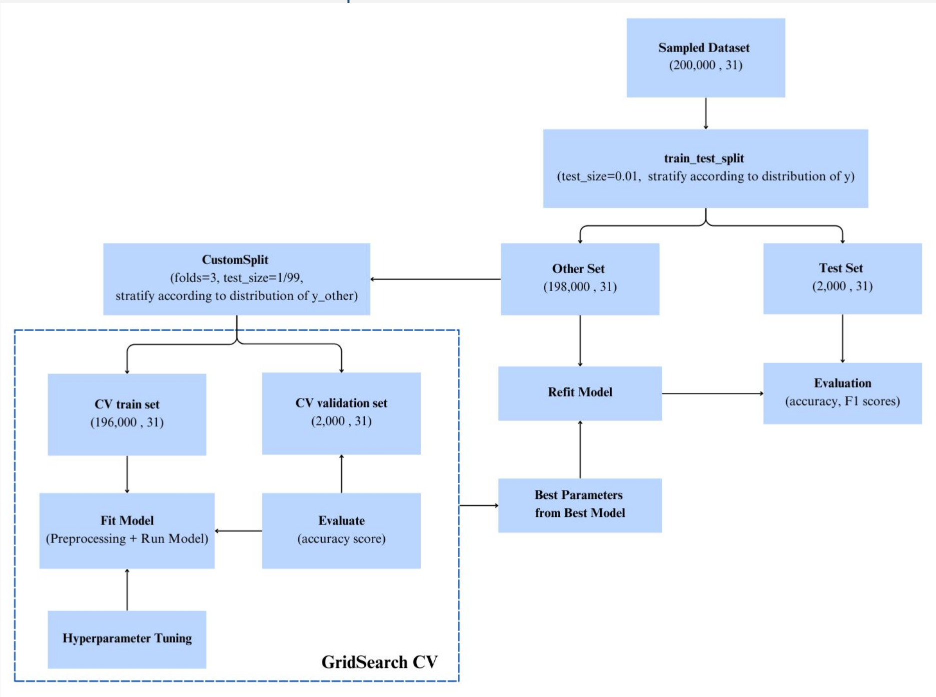
Model Selection & Cross Validation

02

Cross Validation Pipeline

- Define a "CustomSplit" class
    - Functions like StratifiedKFold
    - Configurable number of splits for CV
    - Allows setting a specific proportion for test data (98-1-1)

- CV Pipeline
    - Sampled 200,000 data
    - Iterate through 5 random states
    - Split the data into 'other' and 'test'
    - Create pipeline and GridSearchCV
        - preprocess the data
        - undergo GridSearchCV to train and validate the model
        - scoring = accuracy
    - Find best model, refit on full dataset, and calculate test scores



ML Algos & Hyperparameters

'logisticregression__C': [0.0001, 0.001, 0.01, 0.1, 1]

Logistic Regression

'xgbclassifier__reg_alpha': [1e-1, 1e0, 1e1],

'xgbclassifier__reg_lambda': [1e-1, 1e0, 1e1],

'xgbclassifier__max_depth': [5, 10, 20]

* Redefined Pipeline to implement early_stopping, used early_stopping_rounds=10

XGB Classifier

'randomforestclassifier__max_depth': [1, 3, 10, 30, 100],

'randomforestclassifier__max_features': [0.25, 0.5, 0.75, 1.0]

Random Forest Classifier

'kneighborsclassifier__n_neighbors': [10, 20, 30, 100],

'kneighborsclassifier__weights': ['uniform', 'distance']

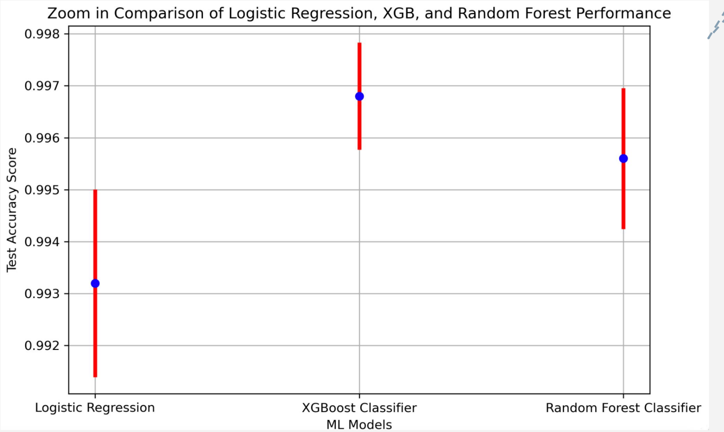
KNeighbors Classifier

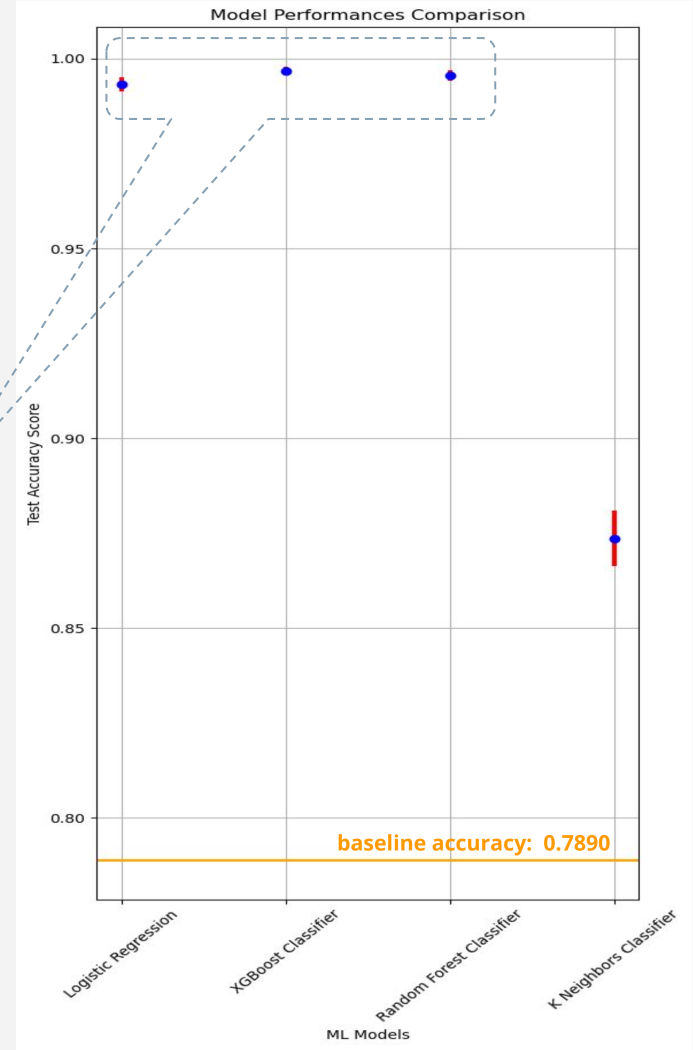
Tuned on a subsample of 200,000 data points for 5 random states

03 Results & Outlook

Results from CV

	Model	Mean Test Score	Std Test Score	#stds Above Baseline Accuracy
0	Logistic Regression	0.9932	0.001806	113.0
1	XGBoost Classifier	0.9968	0.001030	202.0
2	Random Forest Classifier	0.9956	0.001356	152.0
3	K Neighbors Classifier	0.8736	0.007317	12.0

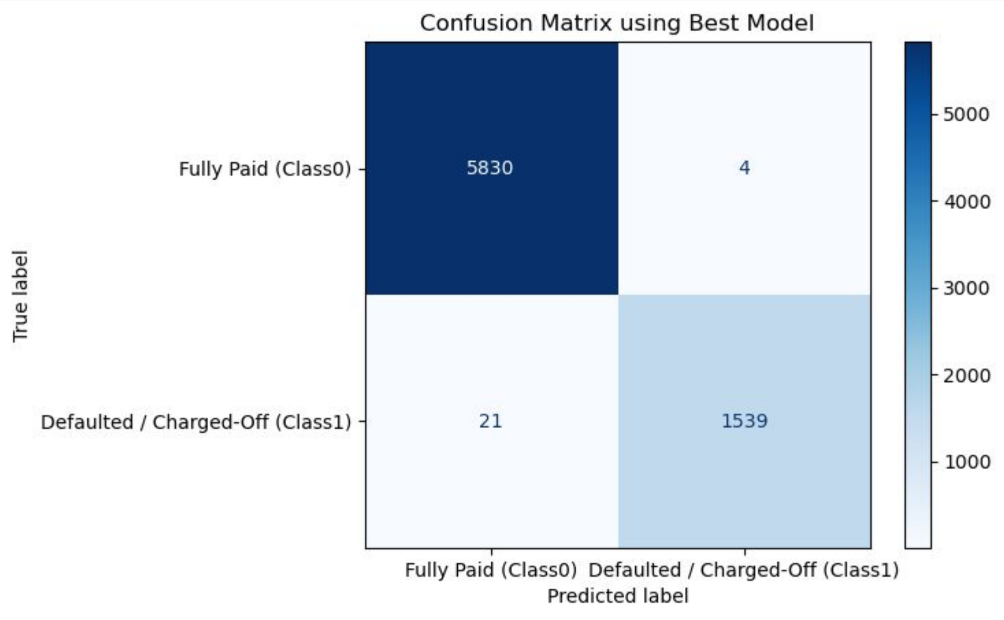




- KNN struggles in high-dimensional spaces (the curse of dimensionality) and with non-linear relationships between features
- KNN can be significantly impacted by noise and imbalances in the dataset

Best Model

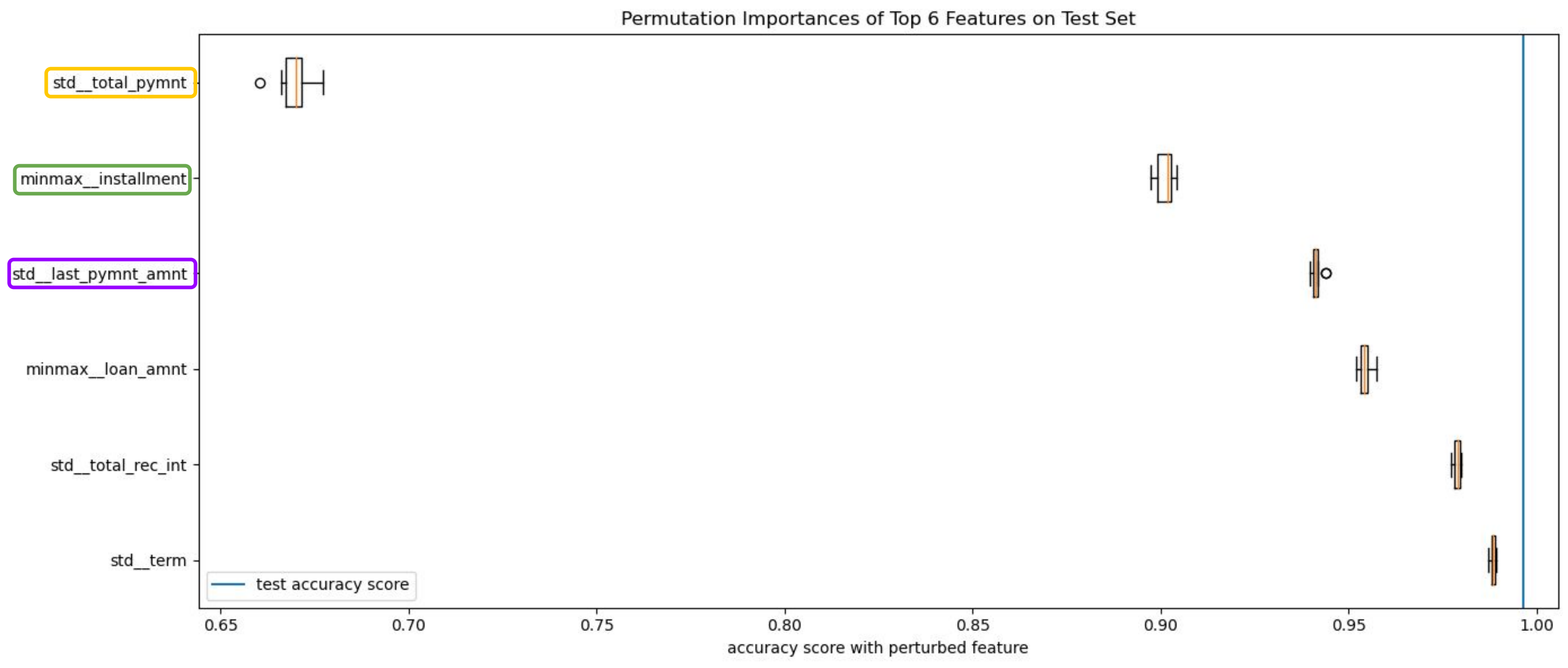
XGBClassifier (random_state=42, colsample_bytree=0.9, subsample=0.8, max_depth=20, reg_alpha=0.1, reg_lambda=1.0, early_stopping_rounds=10)



- Refit best model on full dataset
- 98-1-1 split ratio
- cm threshold = 0.2

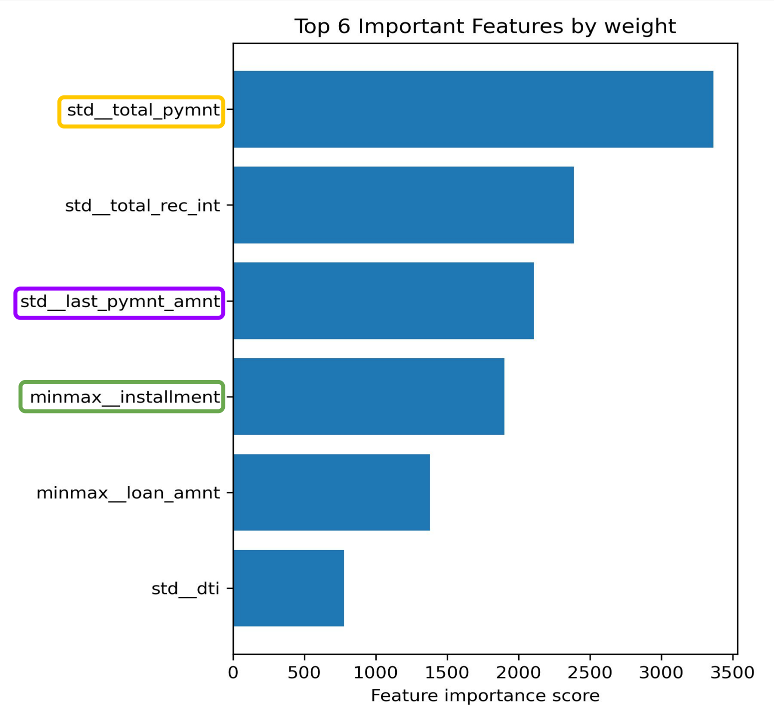
```
Validation accuracy: 0.9981065728969435
Validation precision: 0.9980670103092784
Validation recall: 0.992948717948718
Validation F1 score: 0.9955012853470437

Test accuracy: 0.9966188801731133
Test precision: 0.9974076474400518
Test recall: 0.9865384615384616
Test F1 score: 0.9919432806961006
```

Global Feature Importance

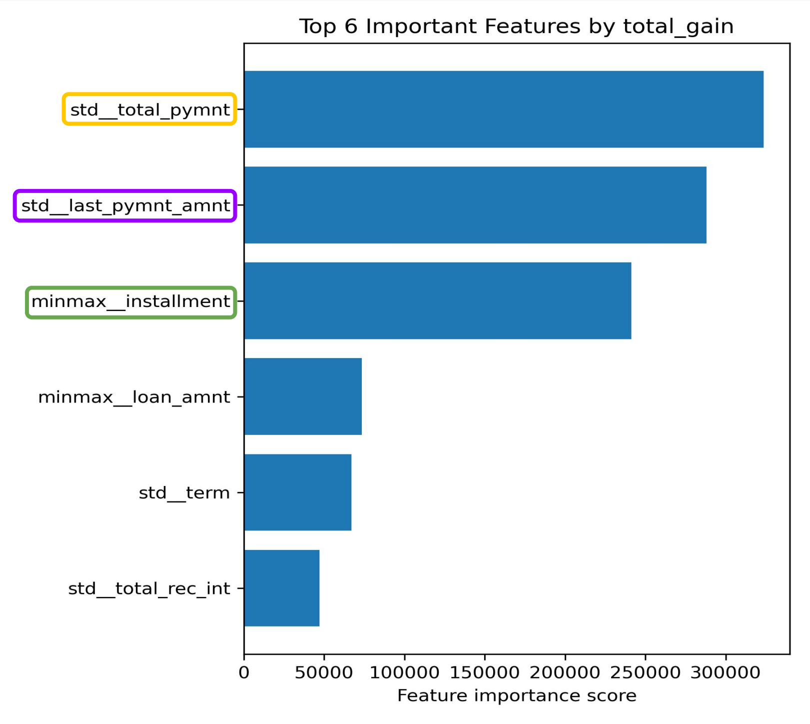


estimates predictive influence based on the drop in model performance due to random perturbation of a feature

Global Feature Importance

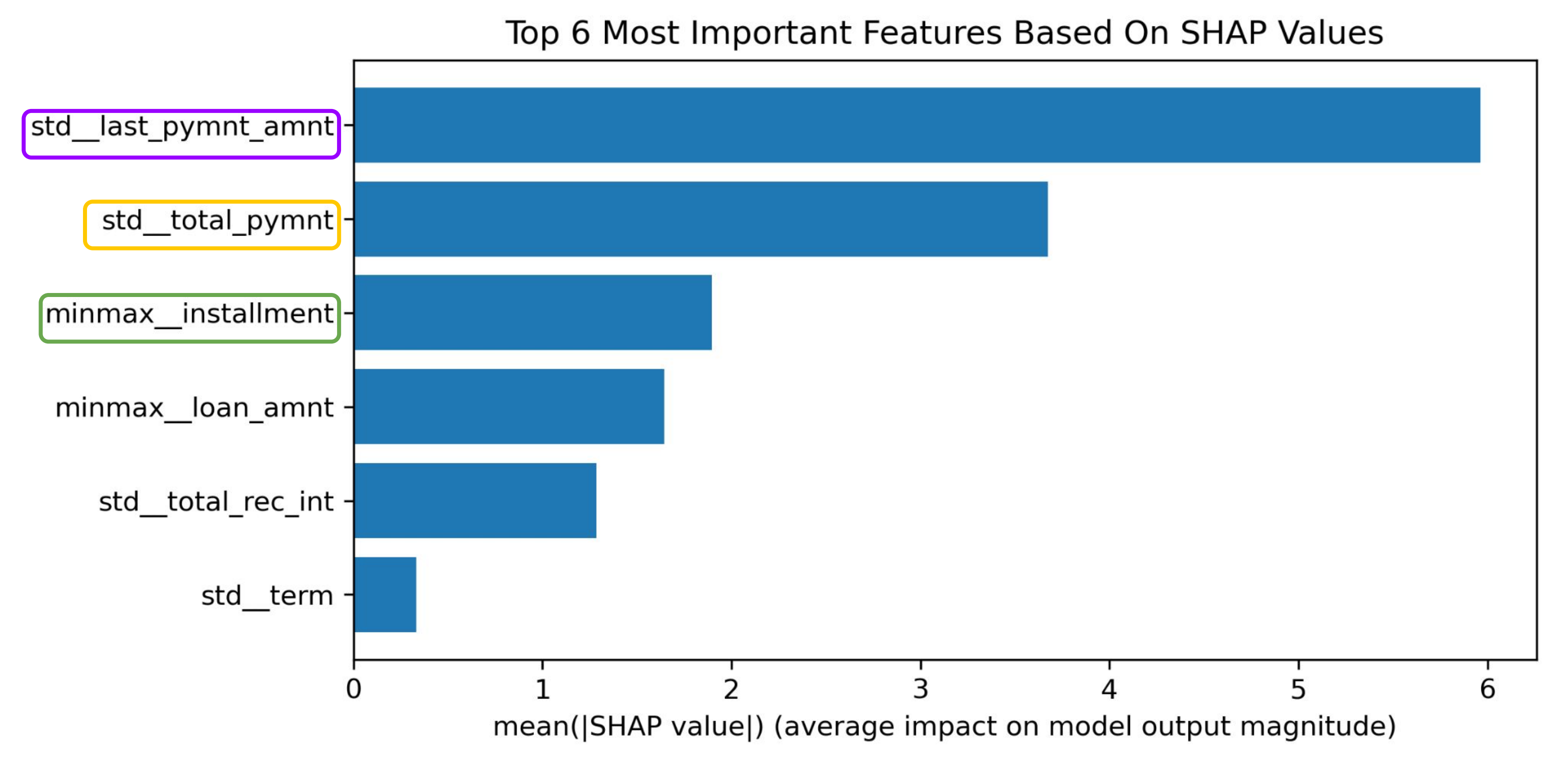


Weight - number of times a feature is used to split the data across all trees.



Total Gain - relative contribution of the corresponding feature to the model's accuracy

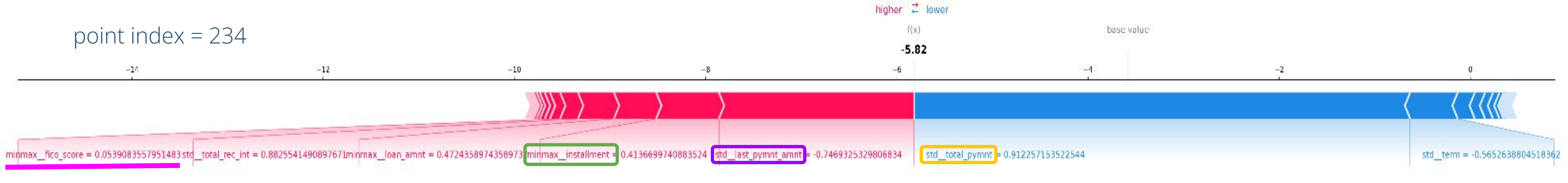
Global Feature Importance



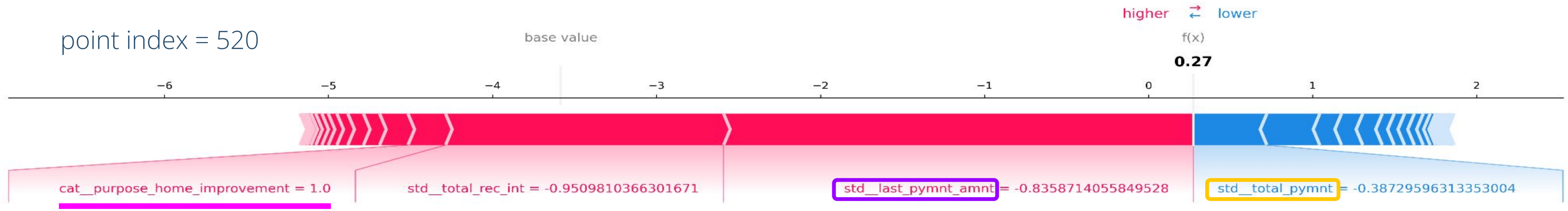
estimates predictive influence based on Shapley values from game theory

Local Feature Importance

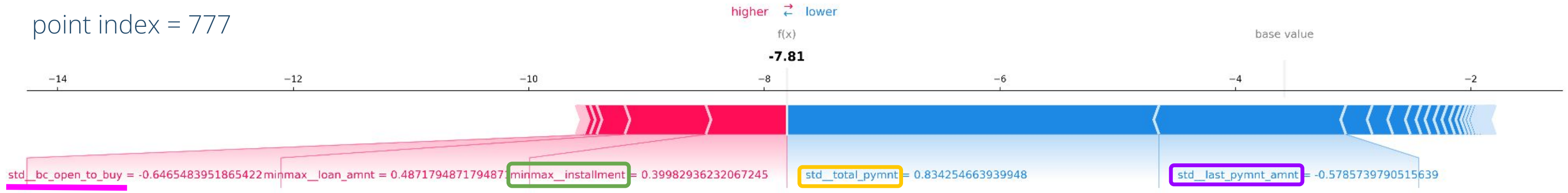
point index = 234



point index = 520



point index = 777



gauge the contribution of a feature to the predictability of a certain point

Outlook

Given more time and more computing power, the project could be improved by:

Increasing CV folds

Due to limitation in computing power, I only used 3 folds in CV. Increasing to 5 or 10 folds would be ideal.

Comprehensive Dataset & Hyperparameter Tuning

Using the full dataset for CV, along with tuning a more thorough hyperparameters could significantly refine the model's generalizability.

Try out more models and techniques

Due to the large size of the dataset, I was unable to run SVM because it was taking too long. Also, I'm curious about exploring non-ML techniques, such as Neural Networks, to see how they can influence model performance and interpretability.

Misclassification Reduction

I would like to explore more ways to minimize false negatives and false positives, which is especially important in loan default prediction. For instance, false negatives can cause credit issues for both borrowers and financial institutions.

Thank you!

Questions?