Model Insights Summary

Github link: https://github.com/AdrianCross2021/loan_ml_modelling/tree/main

EDA on raw data (see eda_cleaned_data.xlsx and eda_raw_data.xlsx in analysis folder)

General Notes

- id is a unique value
- There are currently 8 categorical and 21 numerical/bool columns
- int_rate and int_rate3 are 100% correlated - only one of these should be included
- ~5% of rows are mostly empty and should be removed

Feature Engineering

- To remove
  - All values are null - num_rate, numrate, interest_rate, wtd_loans
  - int_rate2 - gives the same information as int_rate and int_rate3
- Extract numerical values from categorical variables
  - term - pull out 60 or 36 from unique values
  - emp_length - pull out years of employment (assume 10+ is 10, <1 year as 0.5)
  - earliest_cr_line - get days since the earliest_cr_line
- One hot encode
  - Keep all values - home_ownership, purpose, addr_state
- New columns
  - Is_loan_complete - True if Fully Paid, Charged Off or Default, all other loans are assumed incomplete
  - Is_good_loan - defined in target section

Target

- Produce 2 models with 2 different targets
- Model 1 - use the loan_status on all data to determine whether a loan is ‘good’ or ‘bad’ as a binary target
  - bad - 'Late (31-120 days)', 'Charged Off', 'Late (16-30 days)', 'Default'
  - good - 'Current', 'Fully Paid', 'In Grace Period'
- Model 2 - Calculate the predicted profitability of a loan and then predict the value with a regressor model
  - Profitability calculated as (total_pymnt - funded_amnt)
    - A predicted profitability > 0 would indicate a good loan, otherwise a bad loan
  - Only completed loans are fed into the model

EDA on cleaned data

- There are now 3 non numerical columns and 21 numerical/ boolean columns
- The target is unbalanced with the is_good_loan being 95.8% being True
- 12% of the loans are considered complete
- There are 2 columns with null values, mnths_since_last_deling (57% null) and emp_length (4% null) - xgboost can use null values so these are left in
- "total_pymnt","total_rec_prncp", "total_rec_int", "out_prncp", "is_loan_complete" columns are removed to prevent data leakage as these exist after the loan has already been given

Model breakdown

Model 1

- Is a classification model which uses loan status as a boolean target metric
- Hyperparameters are tuned to maximize the recall value

Model 2

- Is a regression model which predicts profit (total_pymnt - funded_amnt) for completed loans only
- Hyperparameters are tuned to maximize the mean absolute error

- Classification model analysis results (confusion matrix, recall etc.) are based on the definition of a positive predicted value being good and a negative predicted value being bad

Advantages of model 1 over model 2
- Model 1 can use incomplete loans which means there are 9.5k data points (vs 1.2k for model 2)
- Model 1 directly takes a loan status from the data which (depending upon business use) can be better than predicting profitability

Advantages of model 2 over model 1
- Output of model is a real cash value which can be used to directly look at revenue impact
    - Uplift is a good example of this where it can show how this model could directly improve revenue
- Uses profitability as a success metric as opposed to loan_status which can misclassify a bad loan
    - E.g. In model 1 a bad loan can have a late payment, but could still be profitable so this could actually be considered a good loan

Model success summary (see confusion matrix and model metrics below)
- Model 1 shows a low recall value of 40% (tuned to this to prevent false negatives), from the confusion matrix you can see that a lot of bad loans are classified as good loans (54 false negatives vs 35 true positive predictions)
- Model 2 shows a better classification of bad loans with 36 true positive values 13 false negatives, however a lot of good loans are classed as bad loans also
    - The uplift value is the most reflective of real value, this shows that if this model was implemented in a way that bad predicted loans are rejected then revenue would drop 30%
    - In further model tuning this is a good metric to watch (depending on business use case)

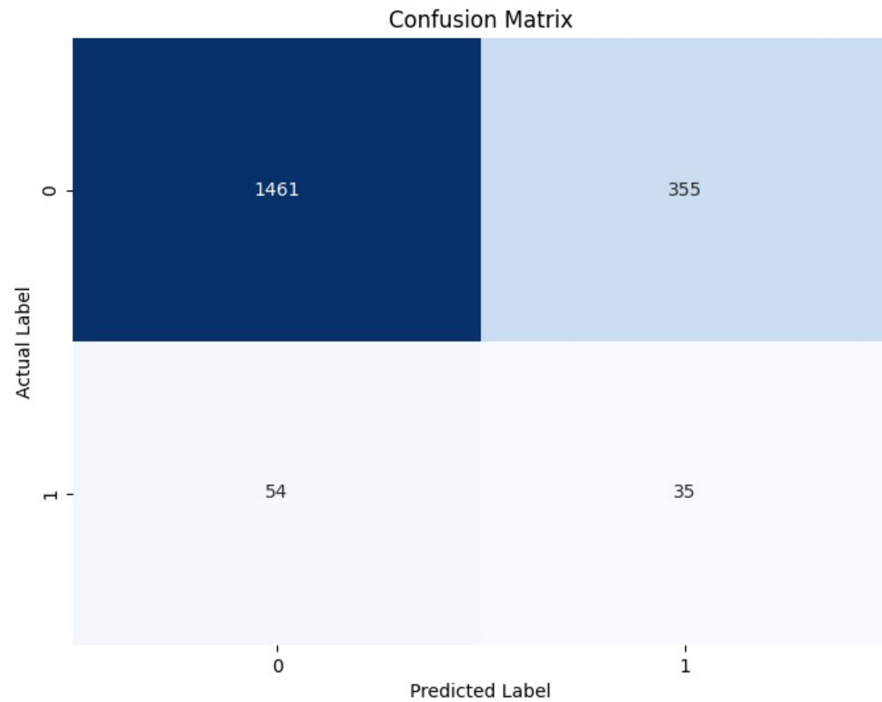
SHAP output analysis (see graphs below)
note: a high SHAP value in model 1 corresponds to a low SHAP value in model 2
- Most predictive columns for both models are interest rate, annual income, debt to income ratio.
    - A high interest rate is indicative of a bad loan
    - A high annual income is indicative of a good loan
    - A high dti is indicative of a good loan, except for some customers with a high dti and very low predictions for model 2
- Most states have a low predictions, given the number of data points these states could be grouped into regions to aid in predictability

Implications of work
- Risk management - when customers apply for loans these models can be used to predict likelihood of repayment
    - Loans can then be denied for high risk customers
    - Loan terms can also be adjusted to minimize risk - for example make predictions on a customer with different adjustable action features (interest rate, repayment amount etc.) and the lowest probability of a bad loan can be chosen
- Efficiency - The model can be deployed in a way that the risks can be assessed quickly with less manual risk assessment work, increasing efficiency and improving customer experience
- Continuous improvement - looking at model insights (SHAP values) predictive features can be further investigated for indications of risk
    - E.g. dti is a highly impactful feature so more granular data around a customers debt and income could be investigated as they are indicative of risk factors (debt amount, income sources etc.)

Potential Improvements
- Deploy model either via api or in timed batches
- Create a baseline model to compare to
- Wrap model runs in docker file
- Pull in more data based on predictive fields from SHAP analysis
- Perform more advanced hyperparameter tuning with more options & parameters
- Explore more advanced target metrics, depending on company requirements and how the model would be implemented, for example include labor costs in loan cost calculations

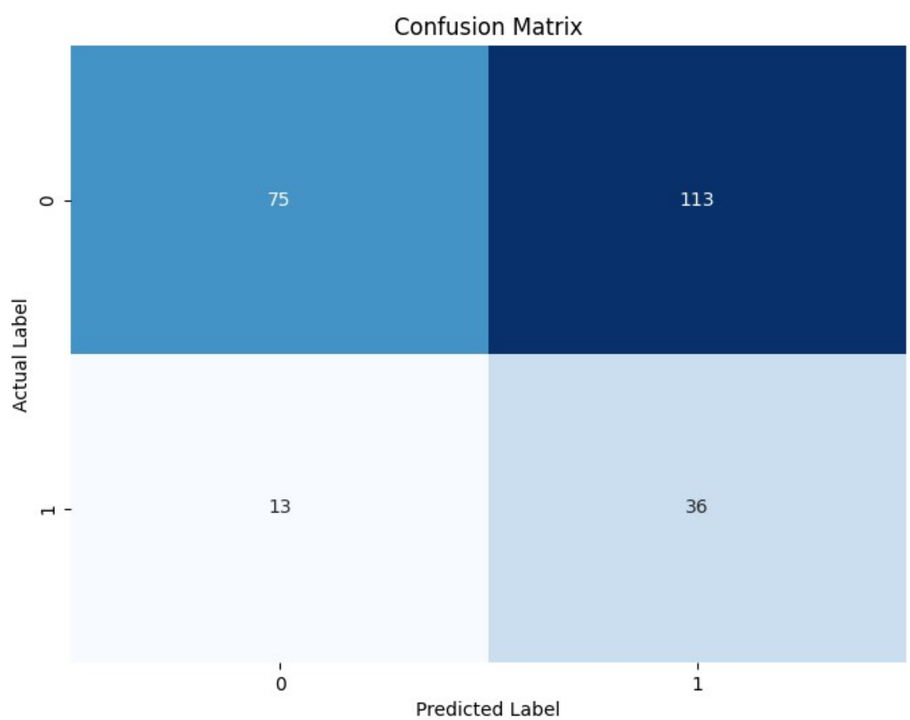
Model 1 metrics

Metric	Score
Accuracy	0.7853018373
Precision	0.08974358974
Recall	0.393258427
F1 Score	0.1461377871

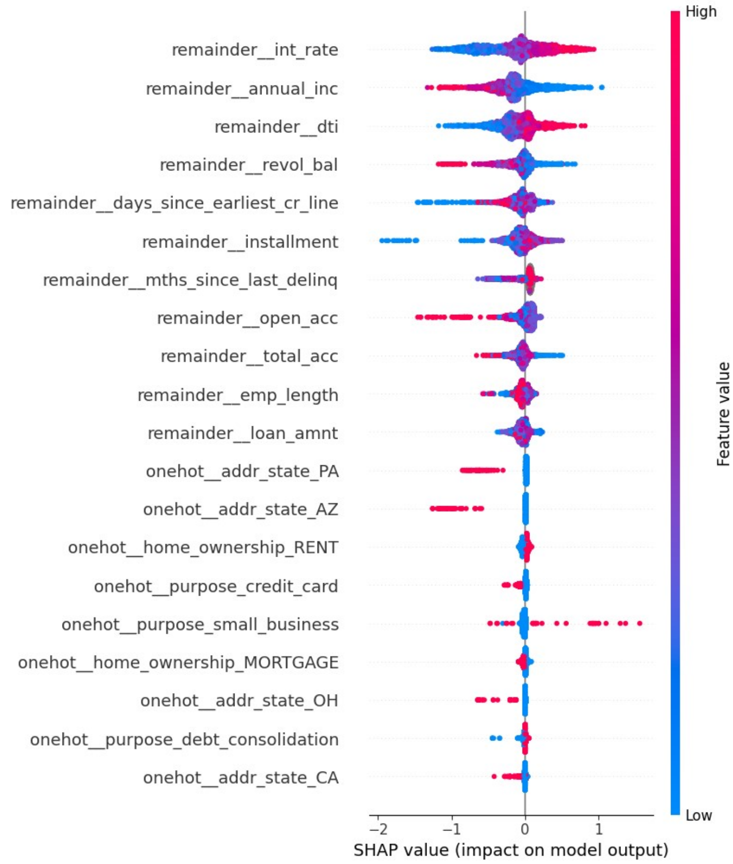


Model 2 metrics

Metric	Score
Accuracy	0.4683544304
Precision	0.2416107383
Recall	0.7346938776
F1 Score	0.3636363636
mean squared error	43915306.99
root mean squared error	6626.86253
mean absolute error	3968.078662
uplift	-0.2952137454



Model 1 SHAPS



Model 2 SHAPS