Data Exploration

Descriptive statistics

The average credit score of all student loan applicants.

6681.97

The average credit score of student loan applicants who received initial credit approval.

748.96

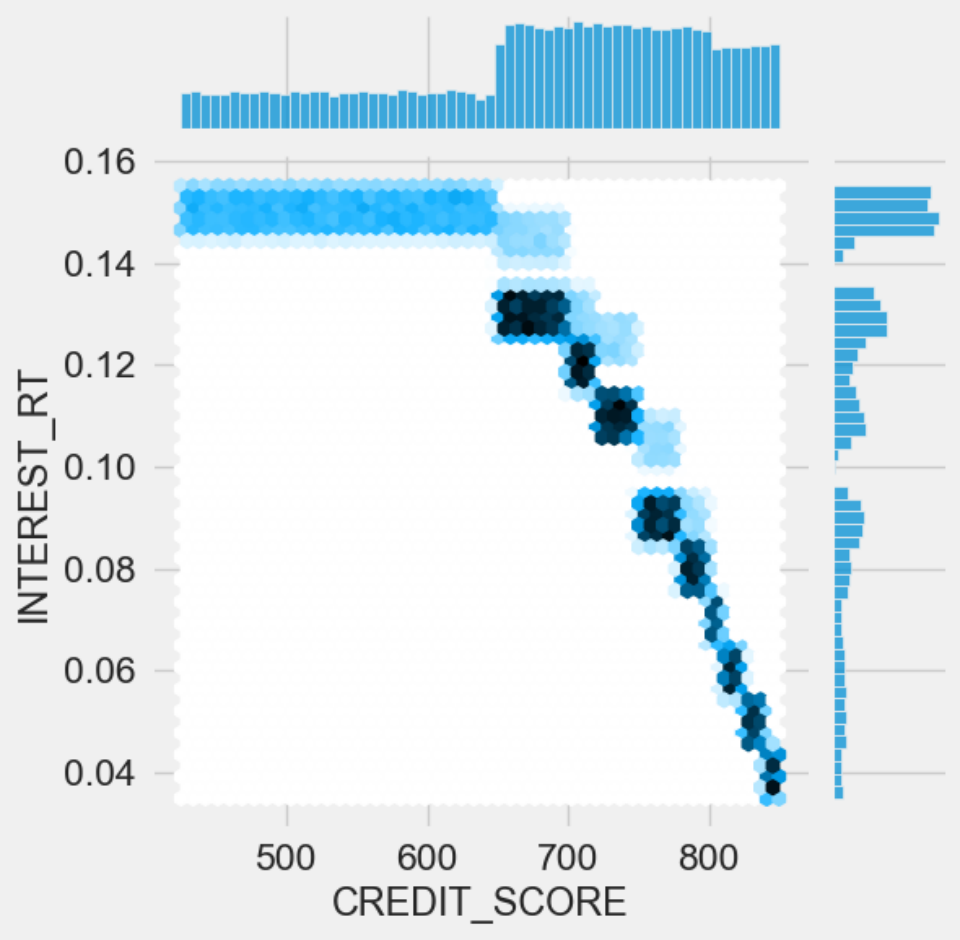
Statistics for the income column

- Count: 99903.00
- Mean: 82702.53
- Median: 72400.00
- Min: 10000.00
- Max: 300000.00
- Standard deviation: 58393.92
- Interquartile Range (75th percentile - 25th percentile):
  - 25%: 36900.00000
  - 50%: 72400.00
  - 75%: 110000.00000

Correlation coefficient between interest rate and credit score

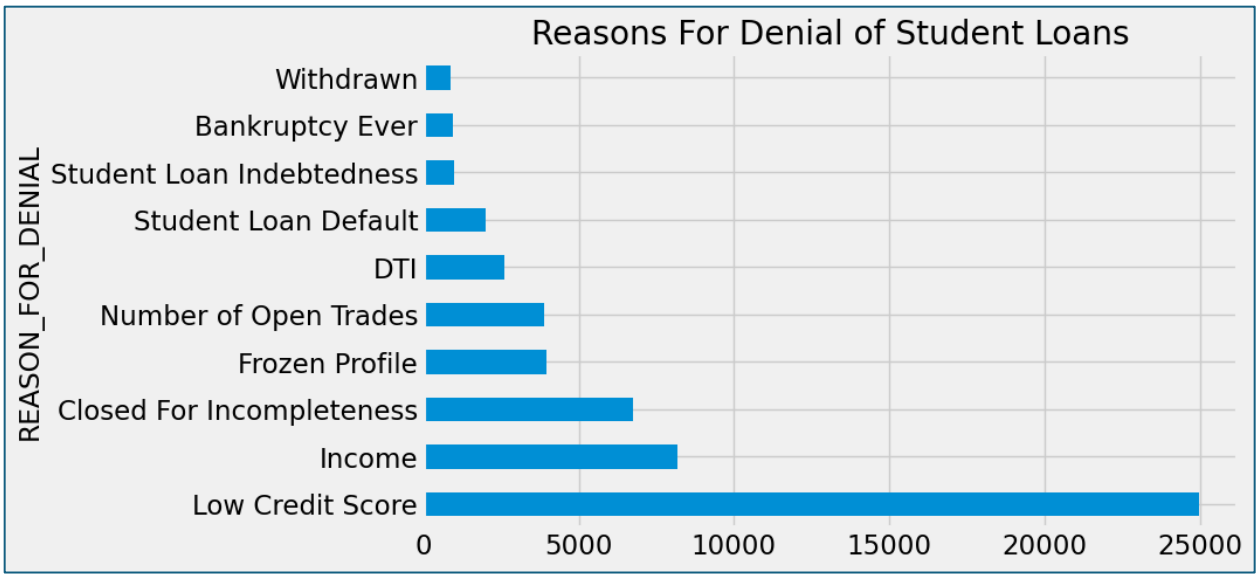
-0.87

The correlation of -.87 shouldn't surprise us here, as credit score goes up, interest rate goes down. Here is a hexbin plot to visualize the relationship. As the color gets darker, the points are increasing in density, we see a crystal clear relation in decreasing interest rate and increasing credit score.



Visualization

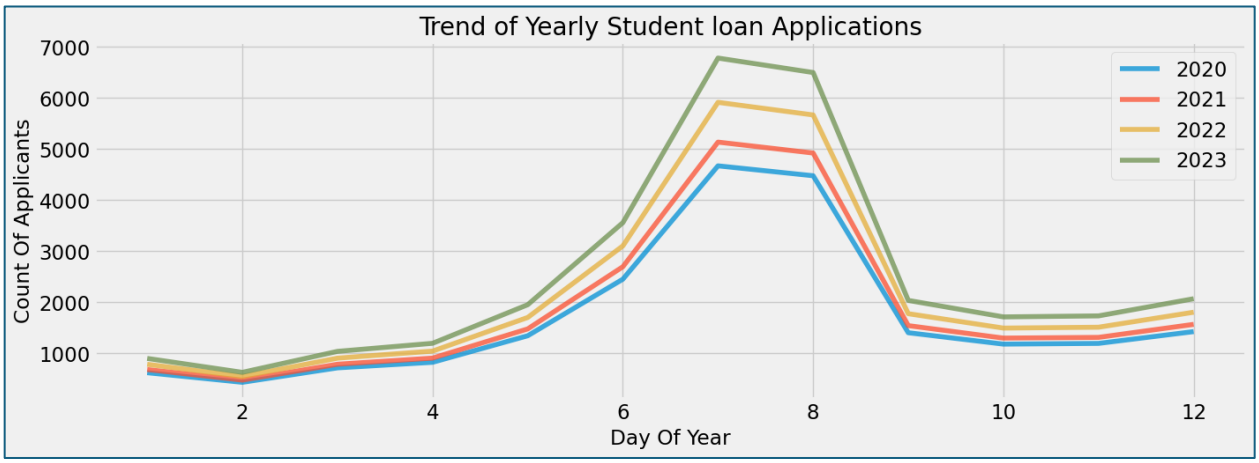
Horizontal bar chart of the different reasons a borrower has been denied a loan



top 3 most common reasons

1. Low credit score
2. Income
3. Closed for incompleteness

Trend of yearly student loan applications



The plot is plotting every yearly seasonal cycle by month. We see strong monthly seasonality around June, peaking in July and august, this makes perfect sense due to the application process. Students are likely applying around this time going into their senior year, or even some last-minute laggards applying a month or two before college.

Feature Engineering

Debt to Income

Debt to income ratio was straight forward in its calculation and can be found in the attached Jupyter notebook.

Principal Component Analysis

Principal Component analysis is a decomposition method used to extract the orthogonal components from a given feature. This can be thought of like taking the toppings off a pizza, PCA aims to extract only the unique parts of features and is ideal to use in cases like this, where multi-collinearity is present and harming model performance, but dropping all multi-colinear features would further harm model performance. I don't use it often due to the large loss in interpretability, it is hard to explain "component 1" to stakeholders.

Predictive Modeling

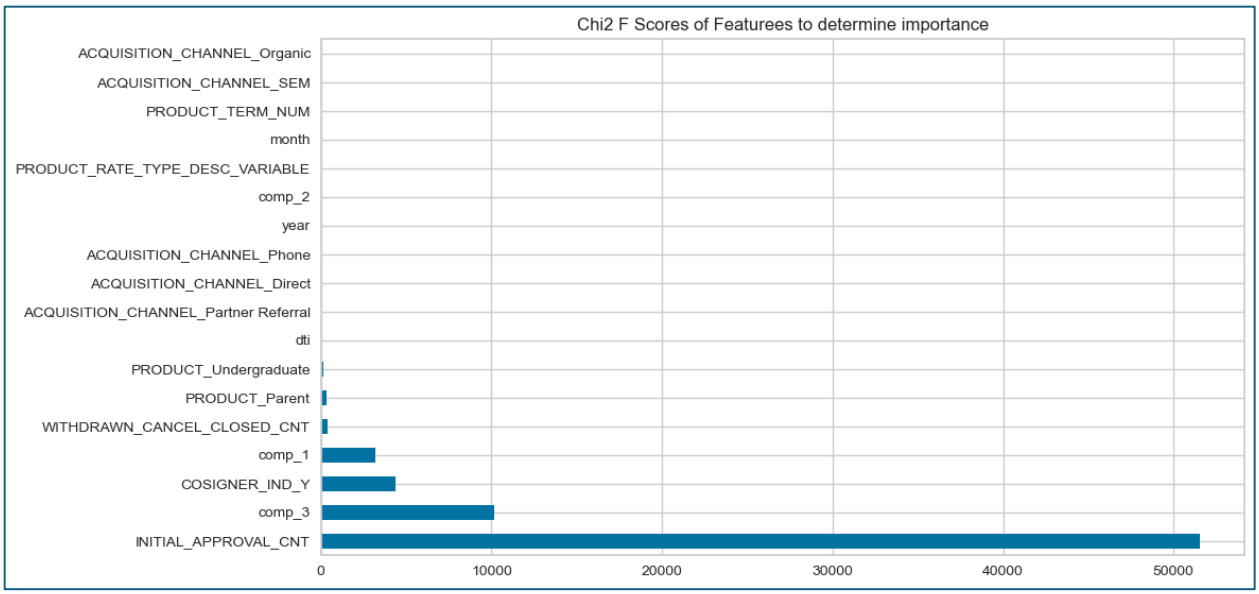
predictive model that predicts whether an application will turn into a 'Booked Loan.'

For our model, we built an AdaBoost Classifier that achieved a 89% F1 Score on the test and 90% f1 score on the training set. This out of the box performance led us to use this model with no additional hyper-parameter tuning. Adaboost is a tree-based boosting ensemble.

Evaluation of model - why the bank should trust this model.

This model showed better performance on unseen data than on the training set, Adaboost is very robust to outliers and gears the ensemble toward better predicting them, which could help the bank highlight anomalies. It outperformed its linear counterpart by 8%, which is an incredible improvement and worth the loss in interpretability.

The most important features of this model



The most important features are plotted here as the chi2 F statistic, the F statistic indicates the magnitude of the features statistical significance. I was hesitant looking at these results, but after sanity checking, 20% of initially approved loans did not end in a positive approval, therefore we can include initial approval count as a feature in the model. As you can see it is surpassing all other features by a huge margin.

Some other interesting notes is the importance of parent driven applications, as well as the cosigner being present in the application. All other features did show significance, but the scores are much smaller. It seems like acquisition channel overall was not very significant. The PCA components were helpful, but in the end are much less interpretable than we would like, therefore we can't speak much to those, only referencing the features from which they were extracted.

Feature Importance Scores in tabular form:

INITIAL_APPROVAL_CNT	51558.88
comp_3	10190.83
COSIGNER_IND_Y	4393.29
comp_1	3181.63
WITHDRAWN_CANCEL_CLOSED_CNT	390.51
PRODUCT_Parent	355.85
PRODUCT_Undergraduate	136.50
dti	23.17
ACQUISITION_CHANNEL_Partner Referral	4.13
ACQUISITION_CHANNEL_Direct	2.68
ACQUISITION_CHANNEL_Phone	1.84
year	0.88
comp_2	0.61
PRODUCT_RATE_TYPE_DESC_VARIABLE	0.12
month	0.09
PRODUCT_TERM_NUM	0.02
ACQUISITION_CHANNEL_SEM	0.00
ACQUISITION_CHANNEL_Organic	0.00

The feature importance scores below are found in tabular form. To further elaborate I thought that month, comp2 and acquisition channel would exhibit far more importance. Specifically, I feel month would've had a higher importance, perhaps a binary "fall month" or "application season" feature would've worked better considering this data was seasonal and the pattern was observed across years. To reiterate above, initial approval, component 3 and cosigner presence dominated the model.

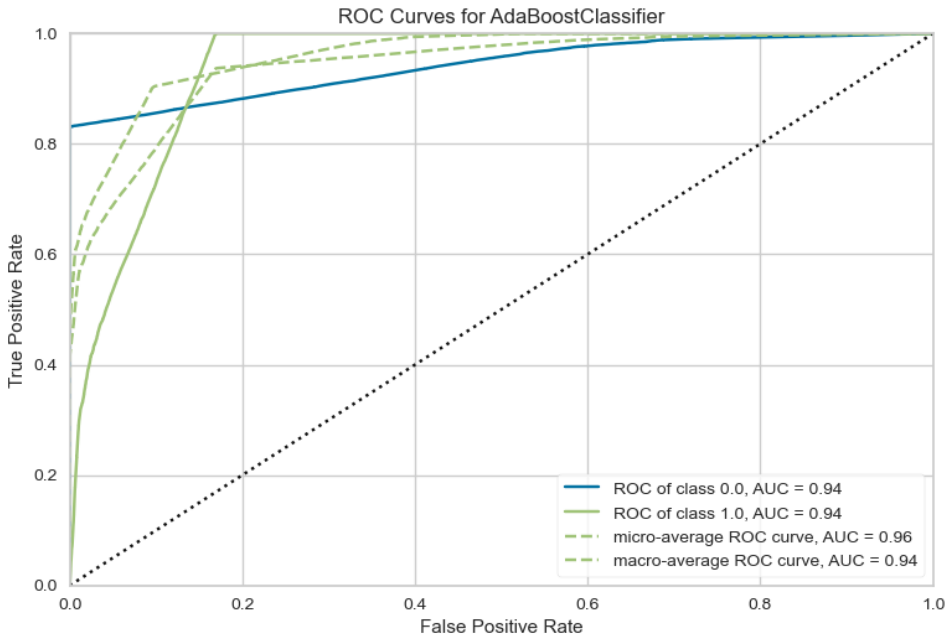
FINAL MODEL PERFORMANCE METRICS:

	Negative Class	Positive Class	accuracy	macro avg	weighted avg
precision	1.00	0.82	0.9	0.91	0.92
recall	0.83	1.00	0.9	0.91	0.90
f1-score	0.91	0.90	0.9	0.90	0.90
support	39808.00	30125.00	0.9	69933.00	69933.00

The above table displays a classification report of my Adaboost model. The Accuracy, F1 Score (macro and weighted avg) and the precision/recall across classes. As you can see, we have a balanced classifier that performs well on both classes, slightly underperforming on the positive class. The f1 score is my target metric of choice as it is the harmonic mean of Precision and Recall.

To briefly explain precision and recall, precision it tp/tp+fp, or the total correct predictions/ total predicted positives. Recall is tp/tp+fn or the total correct predictions divided by the true positives. So we missed some predictions on the positive class, causing an f1 score of 90%.

ROC AUC Curve Plotted



As seen above the ROC AUC curve of the model is very sharp and has a very high TPR in relation to FPR. The ROCAUC aims to measure False Positive Rate in Relation to TPR, the steeper the curve, the better.