Home Loan Prediction Report

Table of Contents

Executive summary:

The study, driven by the interplay between financial stability and homeownership, leverages a Dream Housing Finance Company dataset to explore various socio-economic factors affecting loan approvals. For some additional context, Dream House Finance Company is a consulting company that helps potential home buyers get suitable loans. Initial model benchmarks revealed skewed data, prompting the use of a median for data filling and removing biased features. Subsequent preprocessing, including resampling with SMOTE, was employed to balance the data and enhance model accuracy. Decision trees emerged as the most effective model post-adjustment. This project provides insights into home loan approvals and offers valuable implications for business managers and future research directions in home financing.

Introduction:

In today's dynamic financial and real estate sectors, grasping the elements that affect home loan approvals is paramount. Our project, "Predictive Analytics in Home Loan Approvals," taps into the capabilities of machine learning to scrutinize this vital issue with a specific lens on Orange County's distinctive housing market.

The deep-seated link between financial security and the pursuit of homeownership, a critical aspect of the American dream, drives this study. Our analysis will explore the factors influencing loan approval decisions to discover trends and insights beneficial to prospective homeowners. We are not just focusing on the accuracy of machine learning models like decision trees, logistic regression, and Naive Bayes; we are also keen to understand the underlying socio-economic factors these models reveal.

Using an extensive dataset from the Dream Housing Finance Company, sourced from Kaggle, our investigation will cover 13 independent variables, encompassing demographic, financial, and property details. The core of potential home buyers gets loan approval, indicated by the dependent variable "Loan Status" in the dataset.

Our project transcends academic theory, representing an exploration of the practical applications of machine learning in home loan financing. We aim to provide a detailed understanding of the loan approval process through scenario simulations and sophisticated analytical methods. We aim to unravel complex queries with empirical evidence, setting the stage for ongoing research and practical implementations in this essential field.

Data Summary

The Dream Housing Finance Company home loan dataset contains 614 observations and 13 independent variables, with 7 of the independent variables having null values. The data type

column indicates a mix of categorical and numerical variables as object and integer or float, respectively.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
```

Data Description

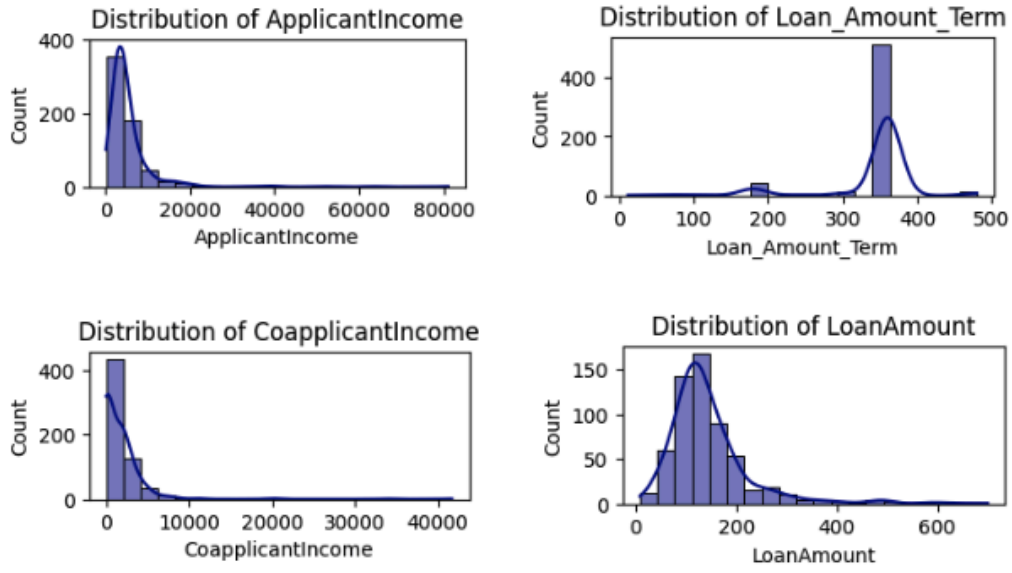
Below is a detailed description of each independent variable:

Independent Variable	**Description**
Gender	Gender consists of male and female, which we converted into a binary variable where male = 1 and female = 0.
Married	Marital status consists of yes and no, which we converted into a binary variable where yes = 1 and no = 0.
Dependents	The applicant's number of dependents ranged from 0 to 3.
Education	The two levels of education are graduate and not graduate, which we converted into a binary variable where graduate = 1 and not graduate = 0.
Self-Employed	Self-employed status is indicated by yes or no, which we converted into a binary variable where yes = 1 and no = 0.
Applicant Income	Applicant's monthly income has a 25% quartile of $2,877, a median of $3,812, and a 75% quartile of $5,795.
Co-applicant Income	If the applicant chooses to have another person on the application as a co-applicant, then there will be a value in the co-applicant income

	column. The co-applicant income ranges from $0, as having no co-applicant, and $2,297.25.
Loan Amount	The loan amount is in thousands, ranging from $9k to $700k. The average loan amount is $146,412.00.
Loan Amount Term	The loan term is in months, starting at 12 months to 360 months.
Credit History	Credit history indicates whether the applicant meets the criteria or not. This is set as a binary variable 0 and 1, where 0 means the applicant doesn't meet the credit history criteria and 1 means the applicant does.
Property Area	The property area indicates the property's location for which the applicant is getting a loan. This is split into urban, semi-urban, and rural. We converted rural to 0, semi-urban to 1, and urban to 2.
Loan Status	Loan status shows if the applicant received a loan or not. It is stated as 'Y' (yes) and 'N' (no), which we converted into a binary variable where Y = 1 and N = 0.

Data Visualization

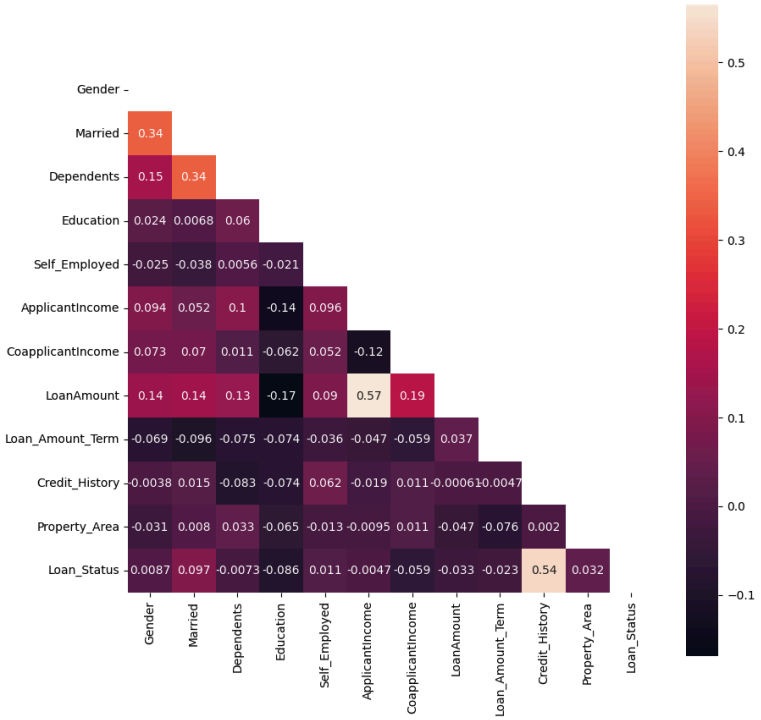
Looking at the numerical data through histograms, the distribution of applicant income, co-applicant income, and loan amount is highly skewed to the left. At the same time, the distribution of loan amount terms is skewed to the right.



The bar charts below show the distribution of our categorical data, which consists of gender, marital status, dependents, education, self-employed, property area, and loan status. We can see that there are many more males than females and married than not married, implying that more males are the primary applicants for a home loan and are seeking a loan as a couple. The majority do not have any dependents and are not self-employed. The spread of the property area is reasonably even, with semi-urban being the most common area. Lastly, loan status, our target variable, needs to be more balanced, with a high yes to a low number of no.

The heatmap displays the correlation between multiple features while color-coding the strength of the correlation. We can see that the applicant's income and loan amount are positively correlated. Applicants with higher incomes can obtain a higher loan amount. Similarly, the heatmap indicates a positive correlation between our target variable, loan status, and credit

history. As an applicant meets the credit history requirement, there is a higher tendency to get a loan approved.



Data Preparation

We had to complete some data cleaning to utilize the data for our analysis. Starting with categorical data, we had to convert it into binary variables, as described in the data description table above. Then, we dropped the Loan_ID and Gender column. The Loan_ID was dropped because it was not helpful in our case and would cause errors due to the data type being a string. Regarding the Gender column, we decided to drop it because someone's gender should not be a factor impacting one's loan status. There are missing values for 7 of our variables, and we had to decide whether to drop those rows or replace them with something else. It was not a good idea to drop those rows because our dataset is small. As shown from the histograms above, all numerical data is heavily skewed. When the data is skewed, replacing missing values with the median of the entire feature column is best. This is because the median is less sensitive to outliers than the

mean. After completing our data cleaning process, below is what our data looks like. No more null values exist, and categorical variables have been converted appropriately.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Married            614 non-null    int64
 1   Dependents         614 non-null    int64
 2   Education          614 non-null    int64
 3   Self_Employed      614 non-null    int64
 4   ApplicantIncome    614 non-null    int64
 5   CoapplicantIncome  614 non-null    float64
 6   LoanAmount         614 non-null    float64
 7   Loan_Amount_Term   614 non-null    float64
 8   Credit_History     614 non-null    float64
 9   Property_Area      614 non-null    int64
 10  Loan_Status        614 non-null    int64
dtypes: float64(4), int64(7)
```

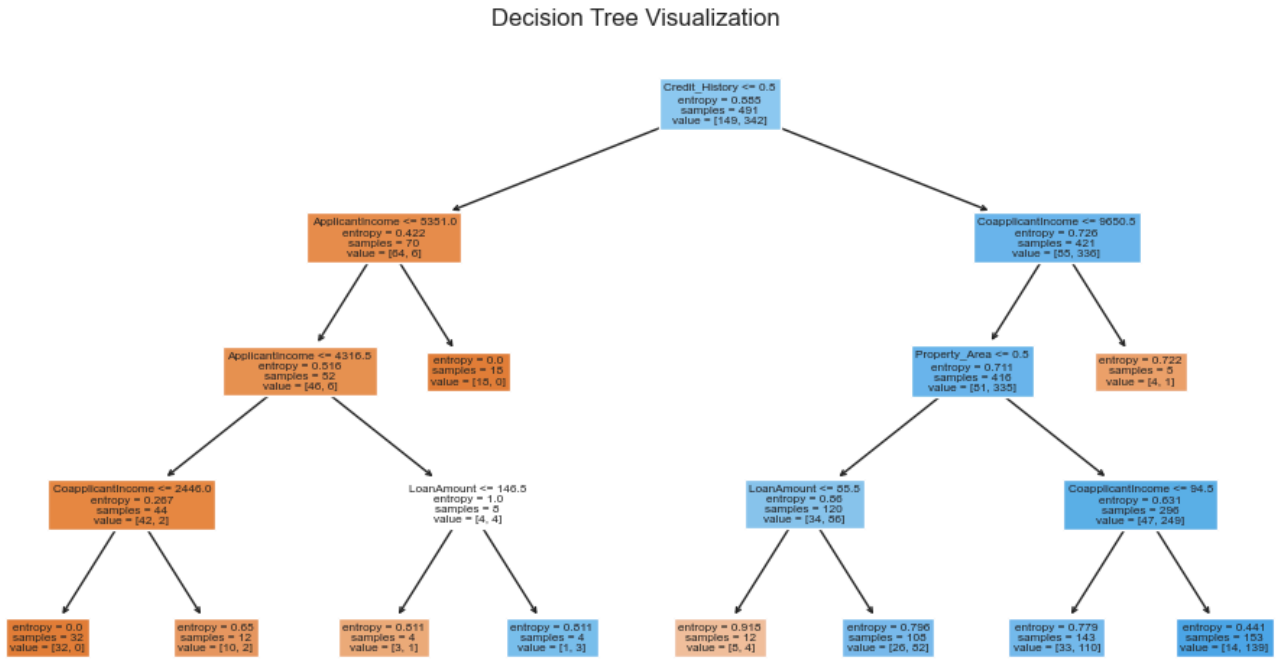
	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	0	0	0	0	5849	0.0	128.0	360.0	1.0	2	1
1	1	1	0	0	4583	1508.0	128.0	360.0	1.0	0	0
2	1	0	0	1	3000	0.0	66.0	360.0	1.0	2	1
3	1	0	1	0	2583	2358.0	120.0	360.0	1.0	2	1
4	0	0	0	0	6000	0.0	141.0	360.0	1.0	2	1
5	1	2	0	1	5417	4196.0	267.0	360.0	1.0	2	1
6	1	0	1	0	2333	1516.0	95.0	360.0	1.0	2	1
7	1	3	0	0	3036	2504.0	158.0	360.0	0.0	1	0
8	1	2	0	0	4006	1526.0	168.0	360.0	1.0	2	1
9	1	1	0	0	12841	10968.0	349.0	360.0	1.0	1	0

Analysis

Benchmark

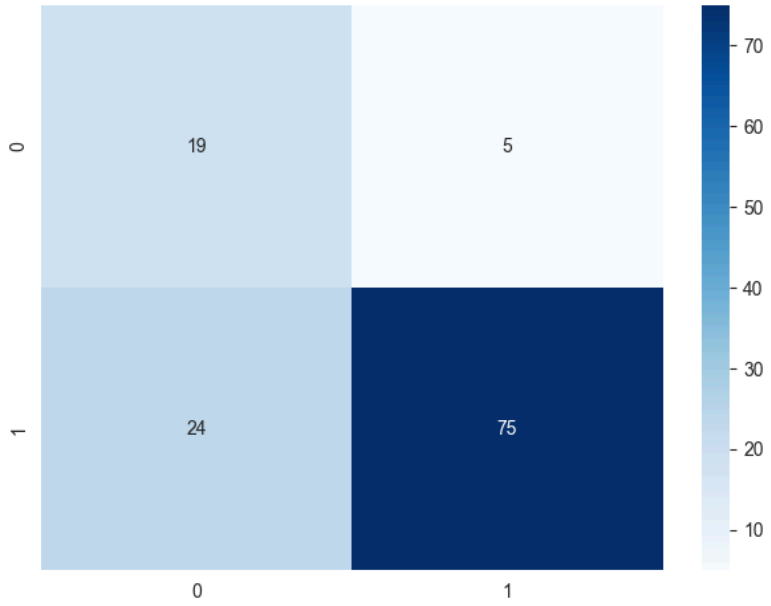
For our analysis, the three classification models we will be running are Decision Tree, Logistic Regression, and Naive Bayes. The data was split based on an 80/20 split, where the models were trained with a random 80% subset of our data, and 20% were used for testing. Decision trees have easy interpretability and visualization. They tend to mimic human decision-making, as many have analogized. Decision trees handle nonlinear relationships

between features and class variables well. Lastly, decision trees can handle mixed data types, such as numerical and categorical data, without preprocessing, such as scaling or normalization. Below is a visualization of our decision tree as a benchmark.



We obtained an accuracy of 76% and a f1 score of 74% from the decision tree classification model. We have provided a confusion matrix below, and we can estimate the stratified accuracies with the confusion matrix. Therefore, for those whore are being denied, the stratified accuracy is 79%, and the for those that are being accepted, the stratified accuracy is 75%.

```
Decision Tree - Accuracy on Validation Set: 0.7642276422764228
Decision Tree - Classification Report on Validation Set:
              precision    recall  f1-score   support

           0       0.79      0.44      0.57        43
           1       0.76      0.94      0.84        80

    accuracy                           0.76       123
   macro avg       0.77      0.69      0.70       123
weighted avg       0.77      0.76      0.74       123
```

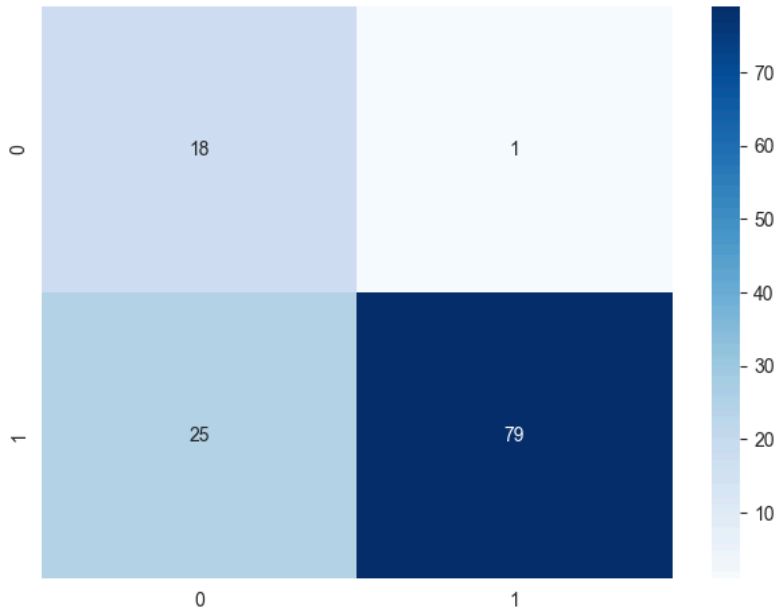


Logistic regression handles binary classification problems well. We have a binary classification problem here, whether we approve a loan or not. Logistic regression also gives probability estimates that can make it more suitable for scenarios, such as understanding the importance of likelihoods of events such as our home loan approval and what factors contribute to this probability. Unlike decision trees, logistic regression can prove well if a linear relationship exists.

We have provided both a classification report and a confusion matrix. Based on our analysis, logistic regression obtained an accuracy score of 79% and an f1 score of 76%. When

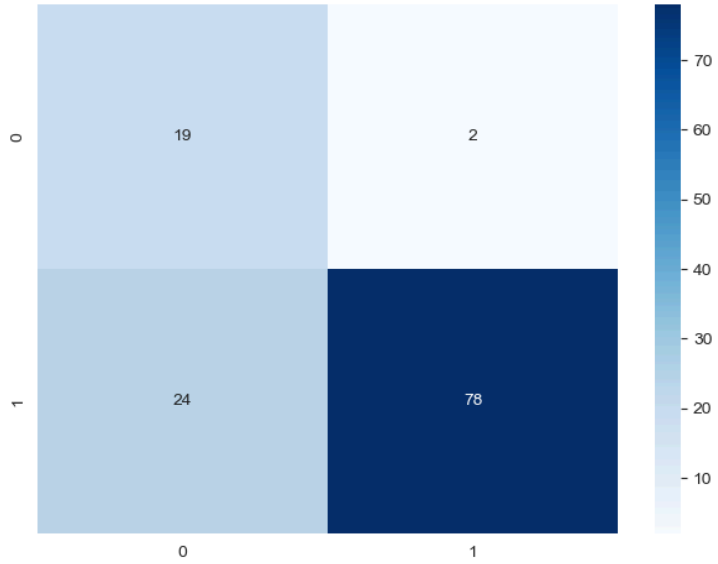
estimating the stratified accuracies, we calculated that the stratified accuracy for those benign rejected is 94%, and the stratified accuracy for those accepted is 76%.

```
Logistic Regression - Accuracy on Validation Set: 0.7886178861788617
Logistic Regression - Classification Report on Validation Set:
              precision    recall  f1-score   support

           0       0.95      0.42      0.58        43
           1       0.76      0.99      0.86        80

    accuracy                           0.79       123
   macro avg       0.85      0.70      0.72       123
weighted avg       0.83      0.79      0.76       123
```



Lastly, for Naive Bayes, the model works well with classifying strings. It is computationally cheap and trains faster compared to other complex models. It can handle large data sets with over 500 rows, which could be helpful in our analysis. Lastly, Naive Bayes can also handle many features. Although we only have 12 features, this could still be useful to us.

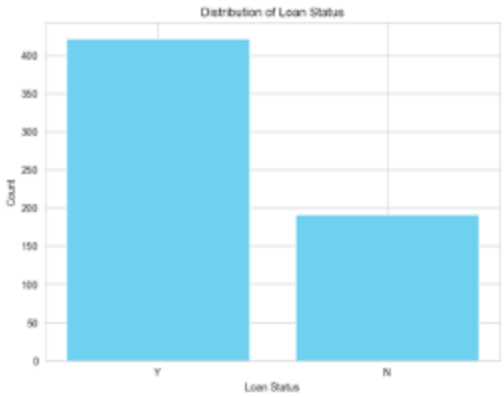
We have provided a confusion matrix below with our findings and classification report. With our analysis, Naive Bayes has an accuracy score of 70% and a f1 score of 81%. We can also estimate that the stratified accuracy for those who would get rejected is 90%, and the stratified accuracy for those who get accepted is 76%.



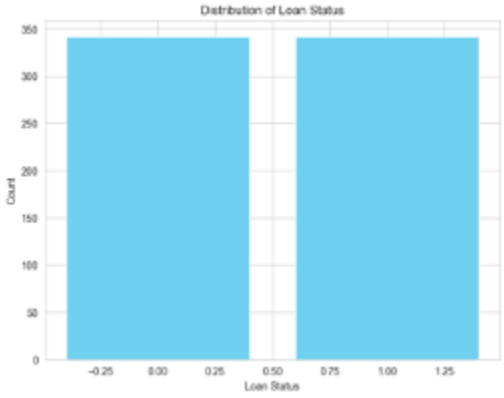

```
Naive Bayes - Accuracy on Validation Set: 0.7886178861788617
Naive Bayes - Classification Report on Validation Set:
              precision    recall  f1-score   support

           0       0.90      0.44      0.59        43
           1       0.76      0.97      0.86        80

    accuracy                           0.79       123
   macro avg       0.83      0.71      0.73       123
weighted avg       0.81      0.79      0.77       123
```

Balancing Data

Before balancing the data, our data was skewed as the distribution of Loan_Status had more “Yes” than “No.” We have provided a bar chart to visualize the distribution of Loan Status before resampling the dataset.



Based on the data, we can see that the number of people who got their loan approved is 422, and the number of those denied is 192. Unbalanced data causes an overfitting problem in our models as they become biased when classifying individuals. From the previous confusion matrix, we can see how our models had more true positives than true negatives. This is because our data was skewed to true positives.



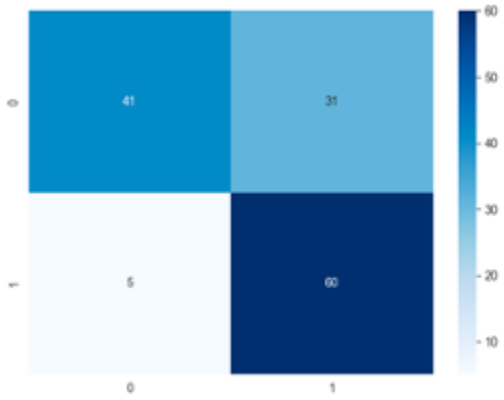
To fix this overfitting problem, we needed to balance our data. To do that, we oversampled our dataset by using SMOTE. When using SMOTE, we set our sampling strategy to ‘auto’ and the random state to 42. As a result, our resampled target contains 342 instances, and our classification attributes also contain 342 instances.

We then split our data into a training set and a testing set where the training set contained 80% of the data, and the remaining 20% was assigned to the validation set. As we did previously,

we will be running Naïve Bayes, Decision Tree, and Logistic Regression and analyzing the model best suited for classifying whether or not individuals can obtain a loan.

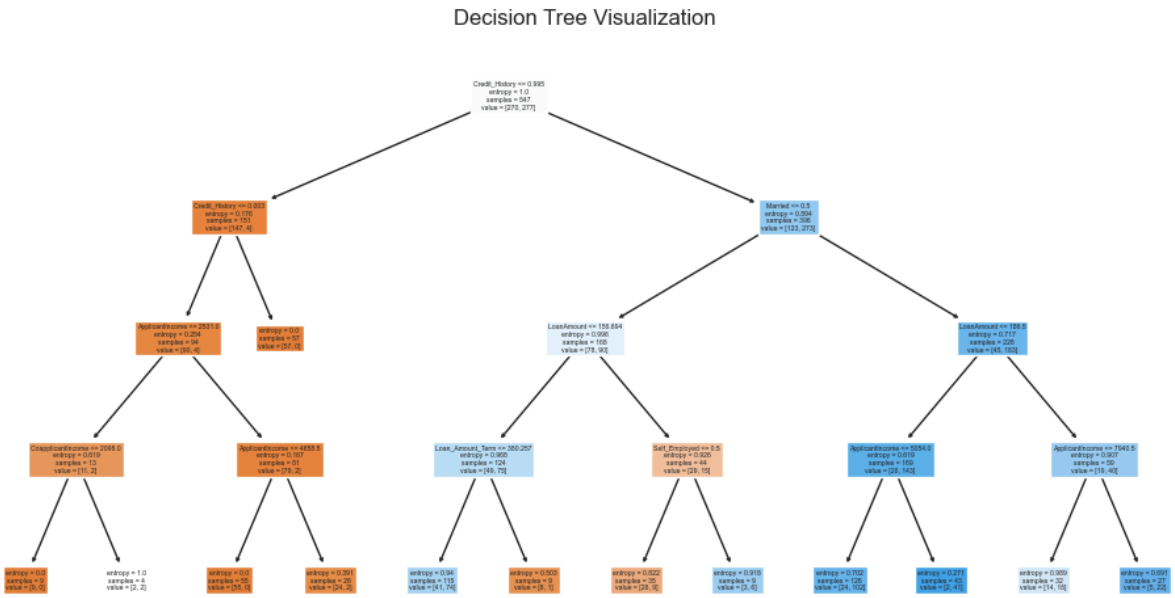
Decision Tree

The first model we will explore with the resampled data is a decision tree, and with this classification model, we obtained an accuracy score of 74% and an f1 score of 73%. Based on the confusion matrix, we can estimate the stratified accuracies of the classifiers, and we calculate that the accuracy for those who would get rejected is 57%. The accuracy for those who would get the loan is 92%.



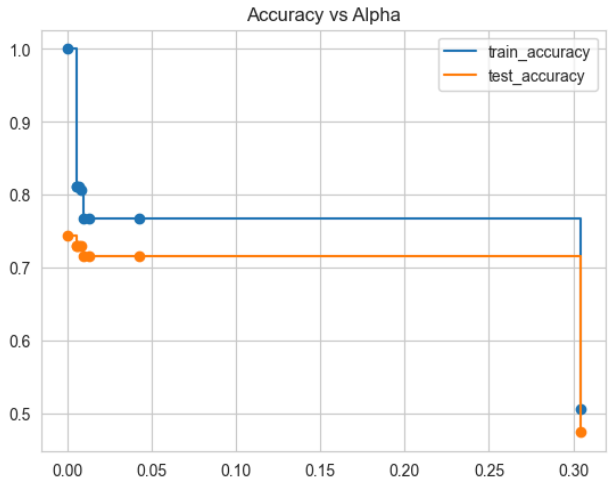
	precision	recall	f1-score	support
0	0.89	0.57	0.69	72
1	0.66	0.92	0.77	65
accuracy			0.74	137
macro avg	0.78	0.75	0.73	137
weighted avg	0.78	0.74	0.73	137

A visualization of our decision tree is provided below, and we can see that the best attributes for the classification model are Credit History, Married, Applicant's Income, Co-applicant Income, Loan Amount, and Self-employed.

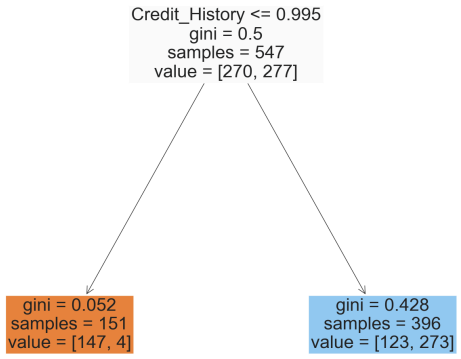


As we proceeded with our analysis, we were looking for a way to improve our model, so we decided to post-prune the decision tree. While post-pruning the decision tree, we obtained a list of alphas that would be optimal for the model. We decided to run an alpha of 0.04241243. We then ran the model and a similar accuracy score and f1 score to the non-pruning decision tree model.

	precision	recall	f1-score	support
0	0.89	0.57	0.69	72
1	0.66	0.92	0.77	65
accuracy			0.74	137
macro avg	0.78	0.75	0.73	137
weighted avg	0.78	0.74	0.73	137



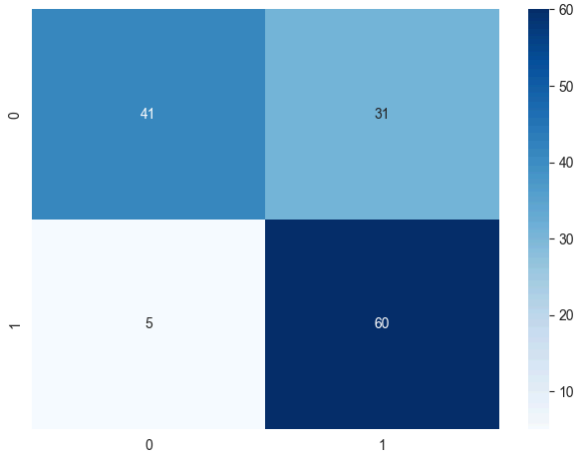
On the other hand, when developing the decision tree while post-pruning it, the visualization changed and indicated that credit history is the only attribute being considered in the model. In theory, credit score is the ultimate factor that determines whether individuals will be able to qualify for a loan or not.



Another way that we modified the decision tree is by pre-pruning it and adjusting its parameters. We ran GridSearchCV to find optimal parameters within a parameter grid. When establishing the parameter grid, we establish the ‘max_depth’ as 2, 4, 6, 8, 10, and 12. We set the ‘min_samples_slit’ as 2, 3, and 4. The ‘min_samples_leaf’ as 1 and 2. Lastly, we set the

'criterion' as entropy. When running the GridSearchCV, it stated that the best 'max_depth' is 4, the 'best min_samples_leaf' is two, the best 'min_samples_splits' is two, and the best 'criterion' is entropy. Based on this decision tree, we obtained an accuracy score of 74% and an f1 score of 73%, the same as the post-pruning process.

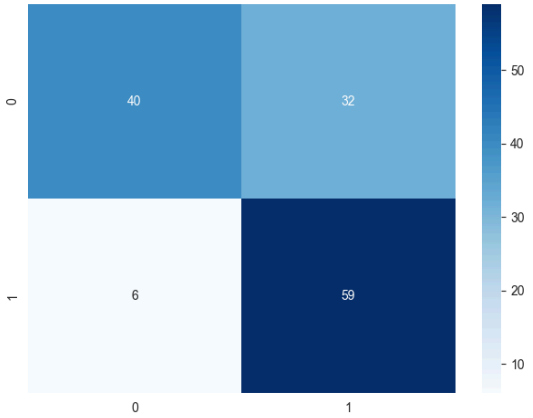
	precision	recall	f1-score	support
0	0.89	0.57	0.69	72
1	0.66	0.92	0.77	65
accuracy			0.74	137
macro avg	0.78	0.75	0.73	137
weighted avg	0.78	0.74	0.73	137



Logistic Regression

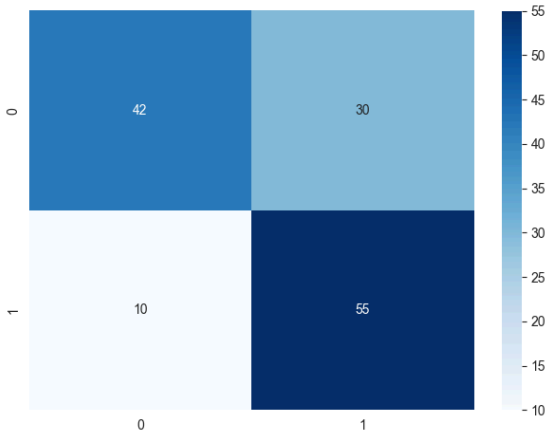
The second classification model we will explore is logistic regression. When running logistic regression, we set the max iteration equal to 1000 and the rando state equal to 42. We obtained an accuracy score of 72% and a f1 score of 72% with those two parameters. We estimate the stratified accuracy to be 56% for those who would be rejected and 90% for those who would be accepted.

	precision	recall	f1-score	support
0	0.87	0.56	0.68	72
1	0.65	0.91	0.76	65
accuracy			0.72	137
macro avg	0.76	0.73	0.72	137
weighted avg	0.76	0.72	0.72	137



In our effort to modify the classification model, we ran GridSearchCV with CrossValidation to determine the best hyperparameters. As a s result, the best parameter was to set the penalty to L2. We then ran the modified logistic regression and obtained an accuracy score of 71% and a f1 score of 70%. When estimating the stratified accuracies, we calculated the accuracy for those being denied to be 58%, and the accuracy for those to be accepted us 84%. When comparing it to the logistic model that was not modified, there was a decrease in all accuracies.

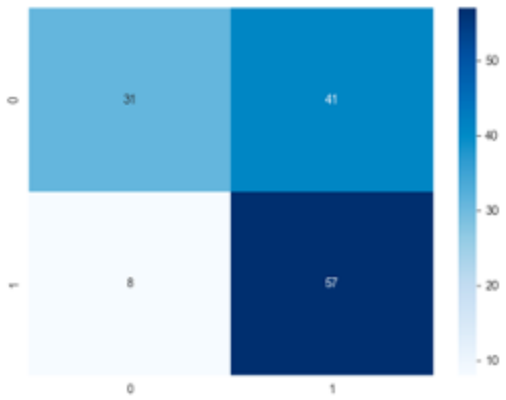
	precision	recall	f1-score	support
0	0.81	0.58	0.68	72
1	0.65	0.85	0.73	65
accuracy			0.71	137
macro avg	0.73	0.71	0.71	137
weighted avg	0.73	0.71	0.70	137



Naïve Bayes

The third and last classification model we will explore is Naïve Bayes. When running Naïve Bayes, we obtain an accuracy score of 64% and an f1 score of 63%. We also provided a confusion matrix to visualize the true positives and true negatives. With the confusion matrix, we can also estimate the stratified accuracy of the classifiers. We calculate that the accuracy for those who would get rejected is 43%, and for those who get accepted is 88%.

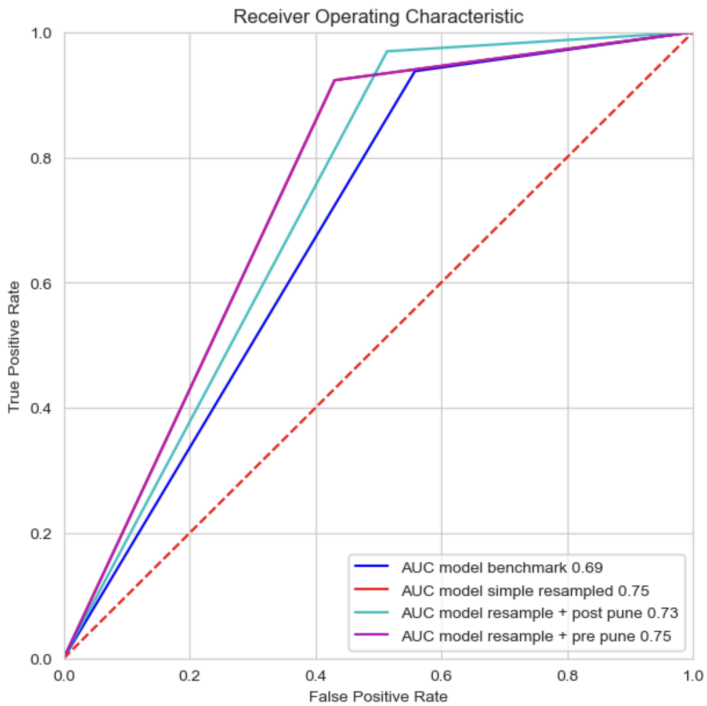
	precision	recall	f1-score	support
0	0.79	0.43	0.56	72
1	0.58	0.88	0.70	65
accuracy			0.64	137
macro avg	0.69	0.65	0.63	137
weighted avg	0.69	0.64	0.63	137



Takeaways:

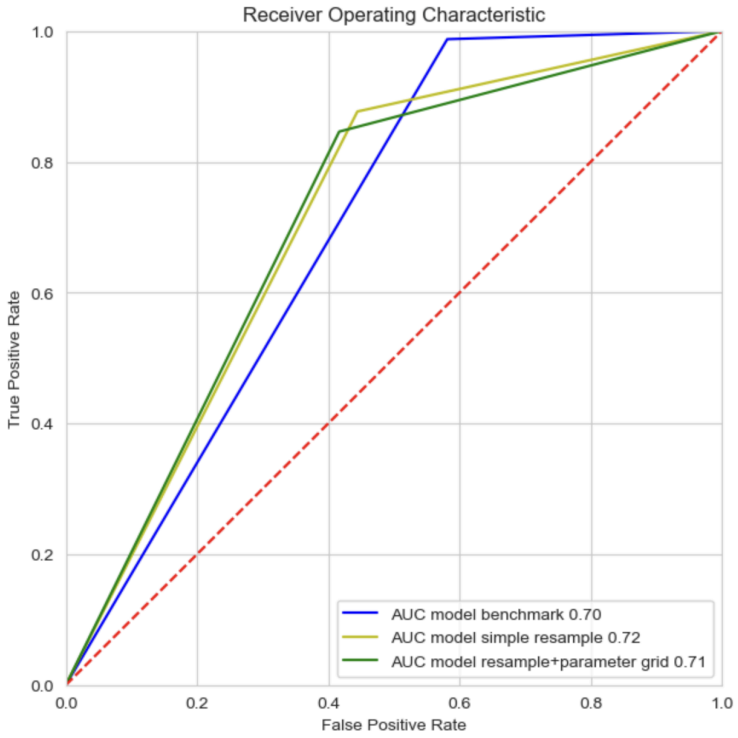
The Decision Tree model is the best model among all the attempts. This classification model gives us a visualization of how to interpret the home loan status and provides us with the optimal attributes when classifying individuals. We can see an improvement when comparing it to our benchmark decision tree. The table below shows that the F1 score has dropped from 0.76 to 0.74 after balancing the data. However, the AUC has improved from 0.69 to 0.75. It is important to note that the f1 score has dropped due to our benchmark's unbalanced data.

Method	**F1 Score**	**AUC**
Decision Tree (no processing)	0.76	0.69
Decision Tree (after SMOTE)	0.73	0.75
Decision Tree (SMOTE + Post Prune)	0.73	0.73
Decision Tree (SMOTE + Pre Prune)	0.74	0.75



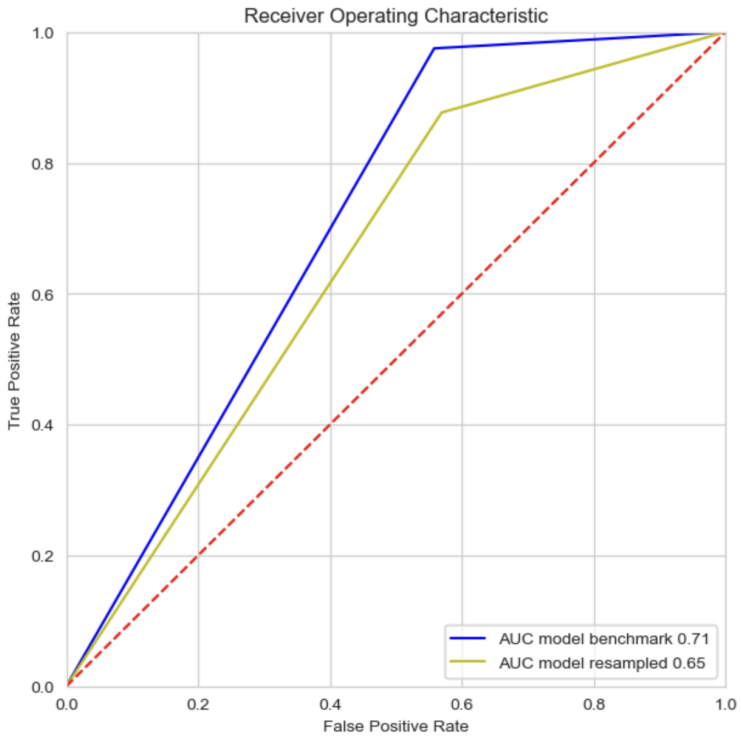
For Logistic Regression, the overall accuracy has decreased from the benchmark of 0.76 to 0.70. Nevertheless, a closer examination of the AUC margin reveals a slight improvement, moving from 0.70 in the benchmark to 0.71 after balancing the data.

Method	**F1 Score**	**AUC**
Logistic Regression (no processing)	0.76	0.70
Logistic Regression (after SMOTE)	0.72	0.72
Logistic Regression (SMOTE + Parameter Grid)	0.70	0.71



For the Naïve Bayes model, we could not improve the model by balancing the data set. For benchmarking, our F1 score is 0.79, and AUC is at a margin of 0.71. However, after balancing the data, the metric went down: the F1 score was at 0.63, and the AUC margin was at 0.65.

Method	F1 Score	AUC
Naïve Bayes (no processing)	0.79	0.71
Naïve Bayes (after SMOTE)	0.63	0.65



Conclusion

In conclusion, to fix the overfitting problem we had with the original data set, we had to over-sample the data by using SMOTE to run classification models using a more balanced data set. While we tried to modify our model, there was no additional accuracy improvement by pre- or post-pruning the decision tree. Overall, we had three objectives we wanted to achieve.

First, we wanted to see which model would best predict home loan approval between decision trees, Naive Bayes, and logistic regression. While our accuracies are lower than our benchmark conducted before processing the data, we can conclude that the best classification after processing the data is a decision tree.

Second, we wanted to look at an archetype of a person interested in buying a home; for example, an unmarried MSBA graduate with good credit trying to buy a home in Orange County suburbs, making $90,000 annually without a co-applicant. Using our last decision tree created, this person would likely be denied. They would make it down to applicant income, second from the bottom right-most, and barely miss the roughly $7,900 income threshold for a higher chance at approval.

Lastly, we wanted to look at the highest information gain in our variables predicted home loans. According to our decision tree, our top three are credit history, marriage status, and applicant income. All, including marriage status, are financial status determinants to a lender. With the cost of living skyrocketing and housing prices following this trend, we are excited about how our model could evolve with more data and help prospective home buyers reach their home ownership goals.