Business Problem:

A Large bank wants to obtain more customers by sending out pre-approval offers for loans. However, they want to target the potential customers with a high chance of being qualified for a loan. In order to do this, they want a model to predict loan approvals based on current and previous customer data.

Background/History:

The use of Data Science in determining credit worthiness is relatively new. Banks started to use credit algorithms around the 1950s and prior to that; applicants would be interviewed by a credit officer that would determine their credit worthiness, resulting in discriminatory practices (Shin & Alvarado, 2022). Around the 50's is when credit reporting started but was improved upon and in 1989 when FICO scores were created (Shin & Alvarado, 2022). However, Discriminatory practices still occurred until 1970-1974 when the fair credit reporting and fair lending acts came along (Shin & Alvarado, 2022). Data science started playing a larger role in finance around the 80s for credit card offers, making determinations on credit card applications and quickly worked its way into more finance fields thereafter.

Dataset:

The data will be loan approval prediction dataset from Kaggle.com: https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset/data

The dataset consists of 13 columns:

- Loan Id: The loan ID corresponding to the applicant's data in the following columns.
- Number of dependents: The number of dependents the applicant has.
- Education: Whether the applicant is a graduate or not.
- Self Employed: If the applicant is self-employed or not.
- Income annum: The annual income of the applicant.
- Loan amount: The loan amount the applicant is applying for.
- Loan term: The term of the loan applied for.
- Cibil Score: The credit score of the applicant.
- Residential assets: The amount of residential assets the applicant has.
- Commercial assets: The amount of commercial assets the applicant has.
- Luxury Assets: The amount of luxury assets the applicant has.
- Bank Assets: The amount of bank assets the applicant has.
- Loan status: Whether the loan was approved or not.

As the loan ID is only an index of the of the rows, this wouldn't have served any purpose to building the model and thus it was removed. Education was reviewed separately as well to see if there was a disparity among graduates and non-graduates receiving a loan approval. The data appeared to be equal among these as seen in figure 1.

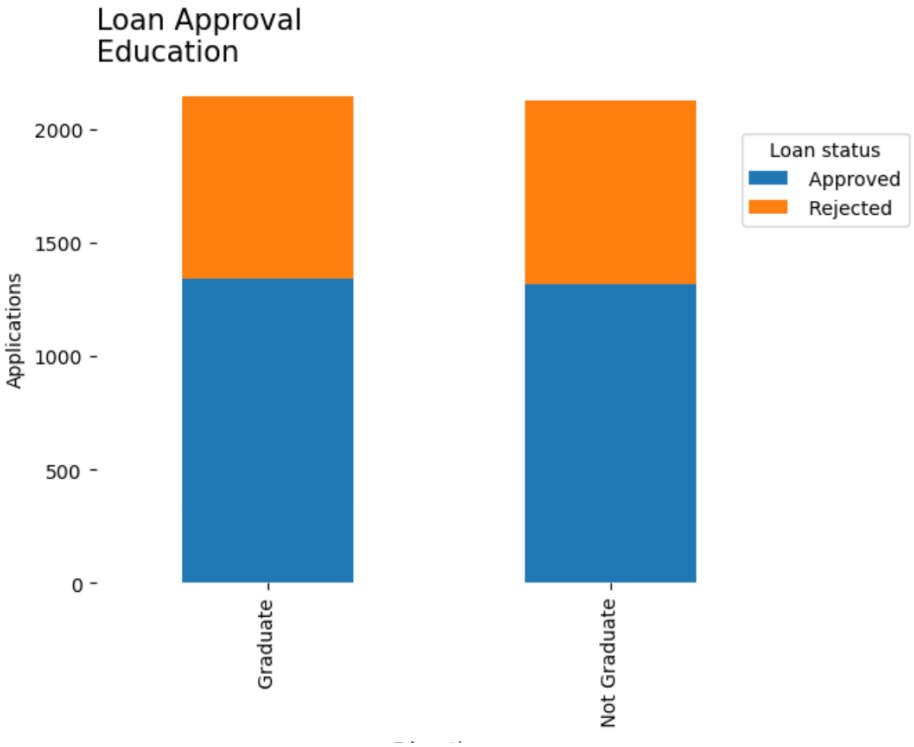


Figure 1: seemingly balance approvals and denials of loans based on education.

However, to prevent the model from favoring one over the other, this was removed from the dataset as well. The next step in preparing the data was to fix the column names. All of the column names had a space in the front and this needed to be removed. If one were to pick up the data and change the code, they may not see the extra space. The data set was then checked for NA values, mixed values, and data types. There were no NA values and to check for the mixed data types I used the describe function. If there were any data types like string this wouldn't have worked. There were no mixed data types in the data set; afterwards the data types were checked to make sure they were all integers. The next step was to get the dummies values for the categorical columns. I applied the get dummies function over the data set using the integer type to get 0 representing false and 1 representing true.

After all of the above was completed, I ran the columns through a loop to generate histograms of the data. I wanted to ensure the data was normalized. The majority of the data appeared to be uniform, but there were 4 columns of concern. These consisted of the residential assets, commercial assets, bank assets, and the loan approval. I pulled the first three separately as they were right skewed or resembling right skew. After applying a

few transformations to the columns, I settled on applying the square root to each in order to make the data appear more normal. Examples of this change can be seen in figure 2.

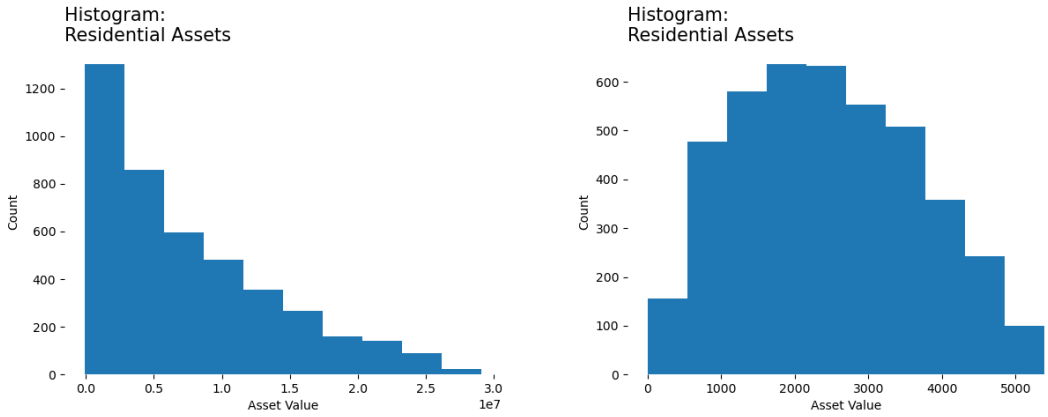


Figure 2: Residential Assets prior to square root function on the left, after on the right.

As for the loan approval column, the amount of loans approved were double that of the rejected loans. This could cause the model to make assumptions that it should lean more towards approving a loan. In order to correct this, I had followed a process that I had found on Project Pros. I had separated the approvals from the rejections and took a random sample from the approvals that would be equal to the number of rejections. Prior to doing this, I obtained the mean for each column and obtained the mean after the transformation. I compared these to ensure that there was no significant change. The last step with processing the data was to check for multi-collinearity. This was done by making a correlation matrix and only keeping the values above the diagonal. I made a list of all the columns to remove, by checking for columns that had greater than 90% correlation. If they met this requirement they were added and I ended up with two. The loan amount and luxury assets. At the moment I had left this in to see if it would have a significant effect on the models.

Methods:

As the data set consists of a binary target variable, I chose to use Linear Regression and Random Forest. Linear Regression was the first choice that came to mind as it is the standard for this type of dataset. This is an effective model for finance as well as ideal for predicting the outcomes as true or false (Yasar, Lawton, & Burns, 2024). As for the reasoning behind Random Forest, this as well is great for modeling the data with binary target columns. However, this will also allow for us to provide a mapping to customers or stake holders to follow and see how the flow of data is sorted to a specific result. Also, the Random Forrest model does have another benefit over the Logistic Regression model. This is that the Random Forest model will help prevent model overfitting, but with higher accuracy the model can become slower (Donges, 2024).

Analysis:

In all the, Liner Regression and Random Forest models were run twice. Once contained all features and the second removed a multicollinearity variable along with a couple that have low importance to the model. The initial Linear Regression model had a total of 52 false positive and false negative predictions with 594 accurate predictions. This model also had an accuracy of 91.95% on the testing data and 92.56% on the training data. The difference between the values doesn't appear significant enough to cause concern for overfitting. The remaining values for precision, recall, f1 score and ROC were all 0.92. With these scores the closer to 1 the better and these results indicate that model had excellent performance.

The Random Forest model had better results. Of 646 loan applications, only 12 had been predicted inaccurately. The accuracy, precision, recall, f-1 score, and ROC were all at 0.98. This may appear to be overfit, but Random Forest models are made to avoid this issue. With these results, the model would perform excellently but may be slower than the Liner Regression model. Once this was completed, I checked the importance of each feature as seen in figure 3.

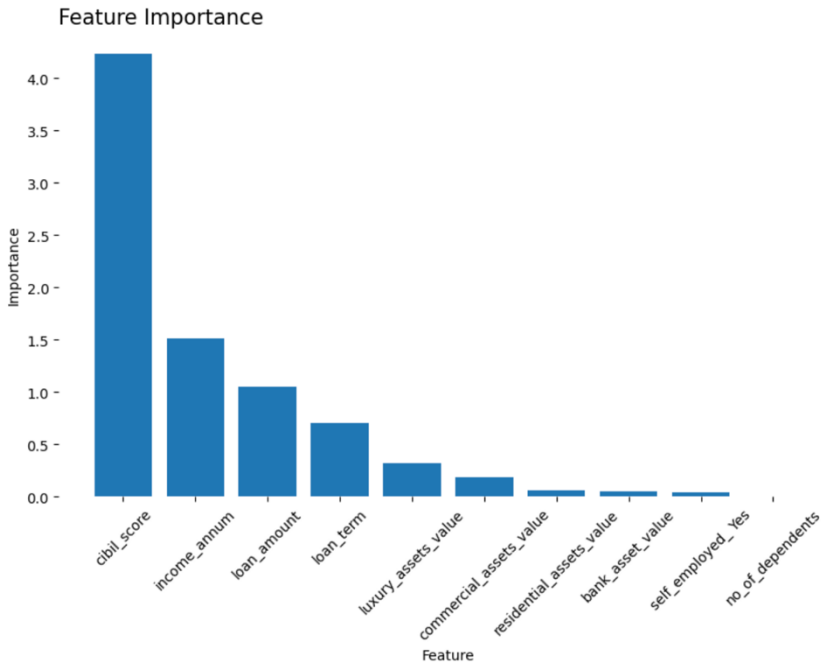


Figure 3: Feature importance.

Based on feature Importance chart, I removed the number of dependents and self-employment status from the data set as they provided little value. I have also removed 1 of the columns that was causing multicollinearity. Between loan amount and luxury asset values, I decided to remove the luxury assets as this had a lower significance to the model. Once this was completed, I ran the models and tests for the models again and there was no change in the results.

Conclusion:

The use of data science in finance is relatively new as its use became more common around the 80's. The data set itself was for the most part clean with no missing values and had 5 columns of concern residential assets, commercial assets, bank assets, the loan approval, and education. Education and index were dropped as basing a loan approval off of education can be seen as discriminatory and index provided no value. The remaining 4 columns were either normalized or balanced. The original models had great results with Logistic Regression having 92% accuracy and Random Forest having 98% accuracy. The features were evaluated on their usefulness. The 2 columns without much contribution and one column causing multicollinearity were removed causing no change to the accuracy of the models except for the shape of the final Random Forest as seen in figure 4.

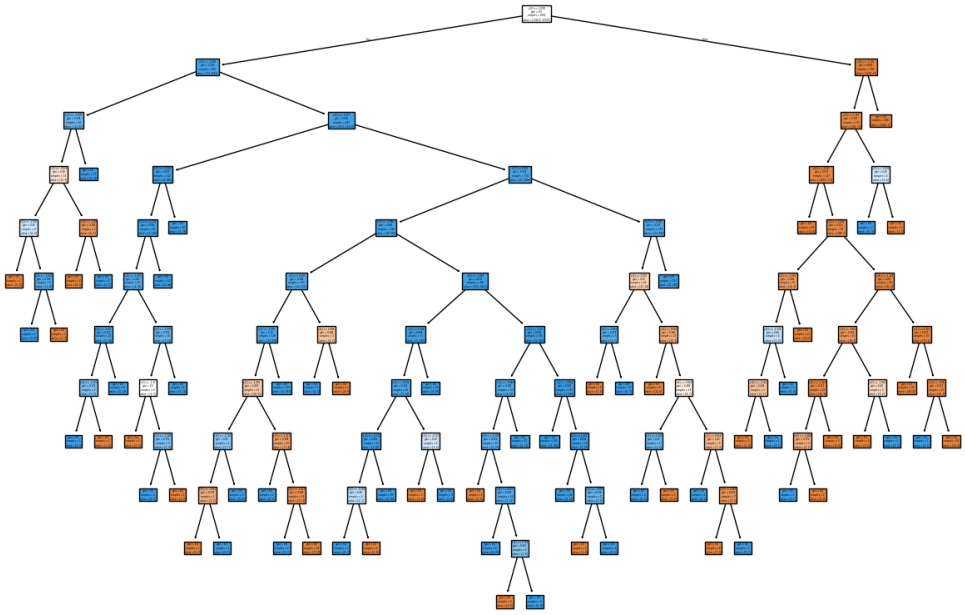


Figure 4: Resulting decision tree.

Assumptions:

I made the assumption that education level could be a source of discrimination. This is because not everyone always has an opportunity to attend college. Also, not all individuals with a higher education are financially responsible. This could have inadvertently caused discrimination within the model or at least some level of inaccuracy. I've also assumed that there were no duplicated rows in the data as well as that reducing the loan approvals to match the loan denials was the best practice. Lastly, I assumed that

if two columns had a high correlation, one could be removed to eliminate this effect on the model. All credit scores were left in with bank assets as this could be a source of outliers.

Limitations:

As far as predicting loan approvals; the applicant's credit score, assets and debt play a large role in determining the loan approval odds. The data itself didn't contain debt and the only risk factor it contained was the credit score. I feel that the model can be improved if there were columns that were used to calculate how risky the applicant would be when offered a loan.

Challenges:

I found it a bit difficult when determining if the models were overfit. I had found a resource indicating that you can tell by the accuracy scores. If the accuracy score of the training data was significantly lower than the accuracy of the testing data then there was a sign the data was overfit. This wasn't an issue for logistic regression but the accuracy of the random forest model made it difficult to tell if it was overfit by using this method.

Future uses:

The loan approval model can be adjusted in many ways. Besides just determining who should get a pre-approved offer, it can be used by the bank to give a quick response on applications that were submitted. Also, this can be applied to credit card applications and even mortgages loans. Companies like Credit Karma can use a model like this as well to send out offers to customers from multiple lenders as long as they know the banks policies for minimal credit score and other risk factors.

Recommendations:

Prior to model deployment it may be best to communicate with the risk departments for loans to determine what they use if more than the credit score to identify risk. If debt is used, then the bank could run a soft credit check on the applicants it plans to send the offer to. This will allow them to get a better idea of how risky it would be to lend to that individual. This information can then be applied to the model as well to get more accurate results.

Implementation plan:

Prior to model deployment, I'd like to confirm the accuracy of the model by connecting additional data sources to it. Currently, the data only has around a thousand rows, which I feel is too little for us to know if it would work as expected when live. I would also need to establish rules for what the model takes in for data, nothing should contain any personal identifying information as to avoid leaking this information. Proper security

measures should be taken as well to ensure the data is safe. Once more data is acquired, the data can be trained and tested on a pilot program. If the accuracy holds, the chosen model can be deployed to a live environment. With regards to which one is chosen, that should be decided upon by the stake holders/ bank as long as they've been briefed and the pros and cons of each model.

Ethical Assessment:

The data itself for the models does not contain any personal information, so loan approvals aren't determined on characteristics of a customer. Any column that could lead to bias has been removed. This includes education level and self-employment status. If an individual has a higher education, it doesn't necessarily mean that they're more financially responsible. The bank assets were left in the model as this can account for outliers or reasons why loans were approved with bad credit. Also, this had little impact on the results of the model so it may only have an insignificant amount of bias from this column. The data was sourced responsibly as it was shared on Kaggle which allows users to make models or work with the data that was provided. All in all, the model is ethically sound.

References

Donges, N. (2024). Random Forest: A complete guide for machine learning. Retrieved from https://builtin.com/data-science/random-forest-algorithm#:~:text=One%20of%20the%20biggest%20advantages,assigns%20to%20the%20input%20features.

How to drop out highly correlated features in python? -. (2023). Retrieved from https://www.projectpro.io/recipes/drop-out-highly-correlated-features-in-python

Kai. (2023). Loan-approval-prediction-dataset. Retrieved from https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset/data

Shin, D., & Alvarado, J. (Eds.). (2022). The history of Credit Score algorithms and how they became the lender standard. Retrieved from https://www.marketplace.org/episode/2022/07/05/the-history-of-credit-score-algorithms-and-how-they-became-the-lender-standard

Yasar, K., Lawton, G., & Burns, E. (2024a). What is logistic regression?: Definition from TechTarget. Retrieved from https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression#:~:text=Logistic%20regression%20is%20ideal%20for,marketing%2C%20finance%20and%20data%20science.

Yasar, K., Lawton, G., & Burns, E. (2024b). What is logistic regression?: Definition from TechTarget. Retrieved from https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression#:~:text=Logistic%20regression%20is%20ideal%20for,marketing%2C%20finance%20and%20data%20science.

Appendix

Data prep:

Count of approvals and denials:

```
data['loan_status'].value_counts()

loan_status
 Approved    2656
 Rejected    1613
Name: count, dtype: int64
```

Checking for missing values and mixed data types:

```
data.isna().sum().sum()

0
```

```
data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
no_of_dependents	4269.0	2.498712e+00	1.695910e+00	0.0	1.0	3.0	4.0	5.0
income_annum	4269.0	5.059124e+06	2.806840e+06	200000.0	2700000.0	5100000.0	7500000.0	9900000.0
loan_amount	4269.0	1.513345e+07	9.043363e+06	300000.0	7700000.0	14500000.0	21500000.0	39500000.0
loan_term	4269.0	1.090045e+01	5.709187e+00	2.0	6.0	10.0	16.0	20.0
cibil_score	4269.0	5.999361e+02	1.724304e+02	300.0	453.0	600.0	748.0	900.0
residential_assets_value	4269.0	7.472617e+06	6.503637e+06	-100000.0	2200000.0	5600000.0	11300000.0	29100000.0
commercial_assets_value	4269.0	4.973155e+06	4.388966e+06	0.0	1300000.0	3700000.0	7600000.0	19400000.0
luxury_assets_value	4269.0	1.512631e+07	9.103754e+06	300000.0	7500000.0	14600000.0	21700000.0	39200000.0
bank_asset_value	4269.0	4.976692e+06	3.250185e+06	0.0	2300000.0	4600000.0	7100000.0	14700000.0

Data types in the data set:

```
data.dtypes

no_of_dependents            int64
income_annum                int64
loan_amount                 int64
loan_term                   int64
cibil_score                 int64
residential_assets_value    int64
commercial_assets_value     int64
luxury_assets_value         int64
bank_asset_value            int64
self_employed_ Yes          int64
loan_status_ Rejected       int64
dtype: object
```

Mean of columns prior to balancing loan approval:

```
data.groupby('loan_status_Rejected').mean().T
```

loan_status_Rejected	0	1
no_of_dependents	2.474774e+00	2.538128e+00
income_annum	5.025904e+06	5.113825e+06
loan_amount	1.524725e+07	1.494606e+07
loan_term	1.039759e+01	1.172846e+01
cibil_score	7.034620e+02	4.294681e+02
residential_assets_value	2.437822e+03	2.479176e+03
commercial_assets_value	1.983607e+03	1.964551e+03
luxury_assets_value	1.501660e+07	1.530694e+07
bank_asset_value	2.080913e+03	2.099672e+03
self_employed_ Yes	5.037651e-01	5.034098e-01

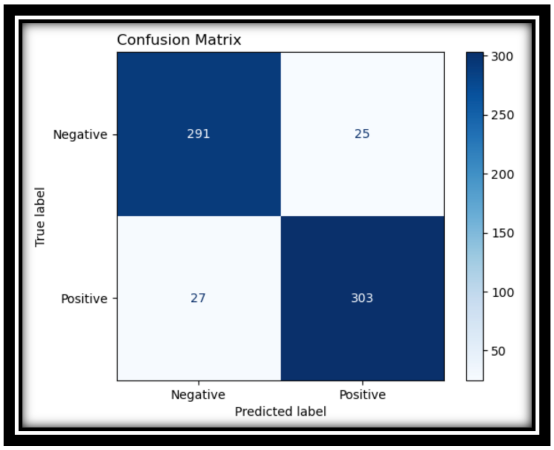
Mean of columns after the balance:

```
new_data.groupby('loan_status_Rejected').mean().T
```

loan_status_Rejected	0	1
no_of_dependents	2.472412e+00	2.538128e+00
income_annum	5.023435e+06	5.113825e+06
loan_amount	1.524668e+07	1.494606e+07
loan_term	1.043397e+01	1.172846e+01
cibil_score	7.005518e+02	4.294681e+02
residential_assets_value	2.426523e+03	2.479176e+03
commercial_assets_value	1.976787e+03	1.964551e+03
luxury_assets_value	1.505077e+07	1.530694e+07
bank_asset_value	2.076978e+03	2.099672e+03
self_employed_ Yes	5.127092e-01	5.034098e-01

Modeling:

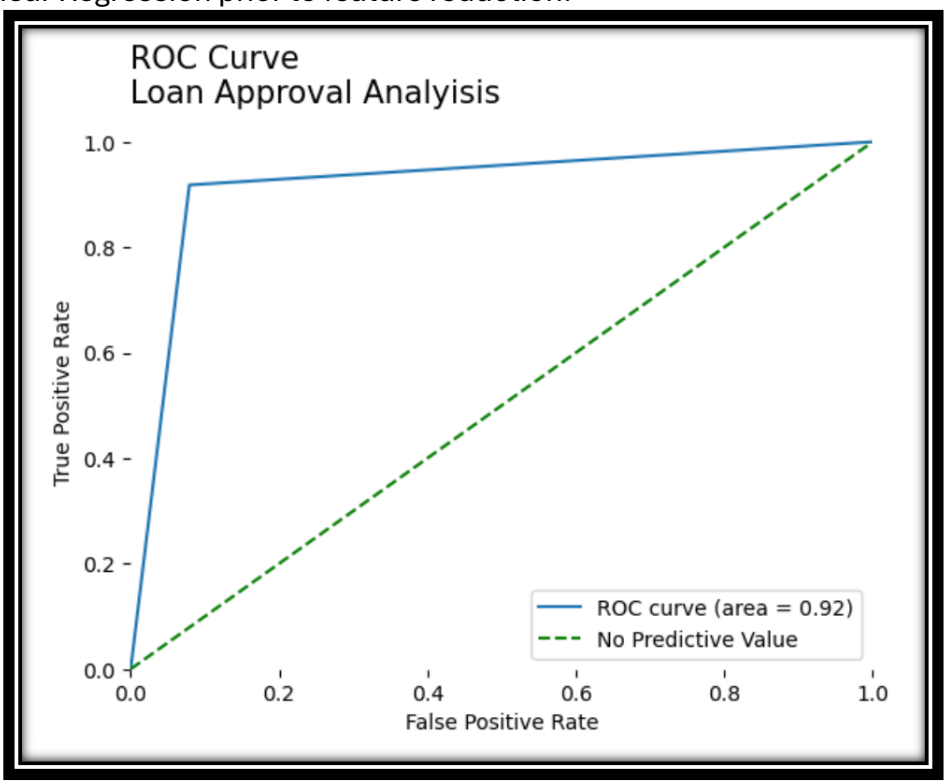
Confusion matrix for Linear Regression before feature reduction:



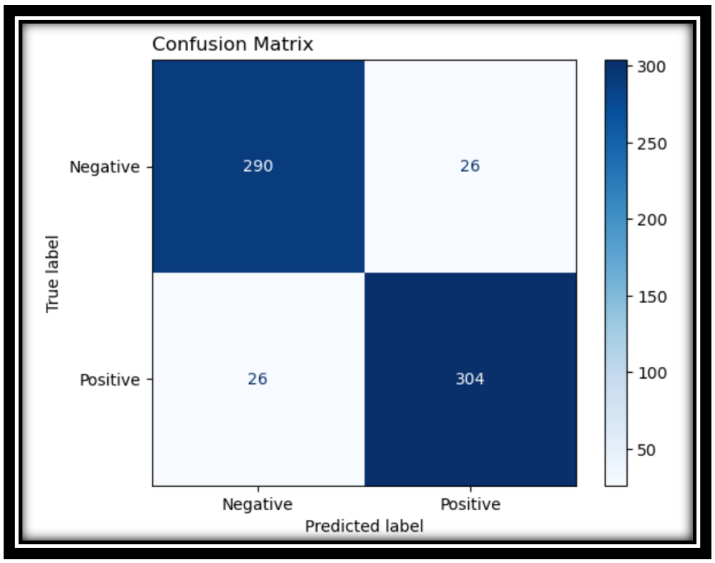
Accuracy report of Linear Regression before feature reduction

	precision	recall	f1-score	support
0	0.92	0.92	0.92	316
1	0.92	0.92	0.92	330
accuracy			0.92	646
macro avg	0.92	0.92	0.92	646
weighted avg	0.92	0.92	0.92	646

ROC for Linear Regression prior to feature reduction:



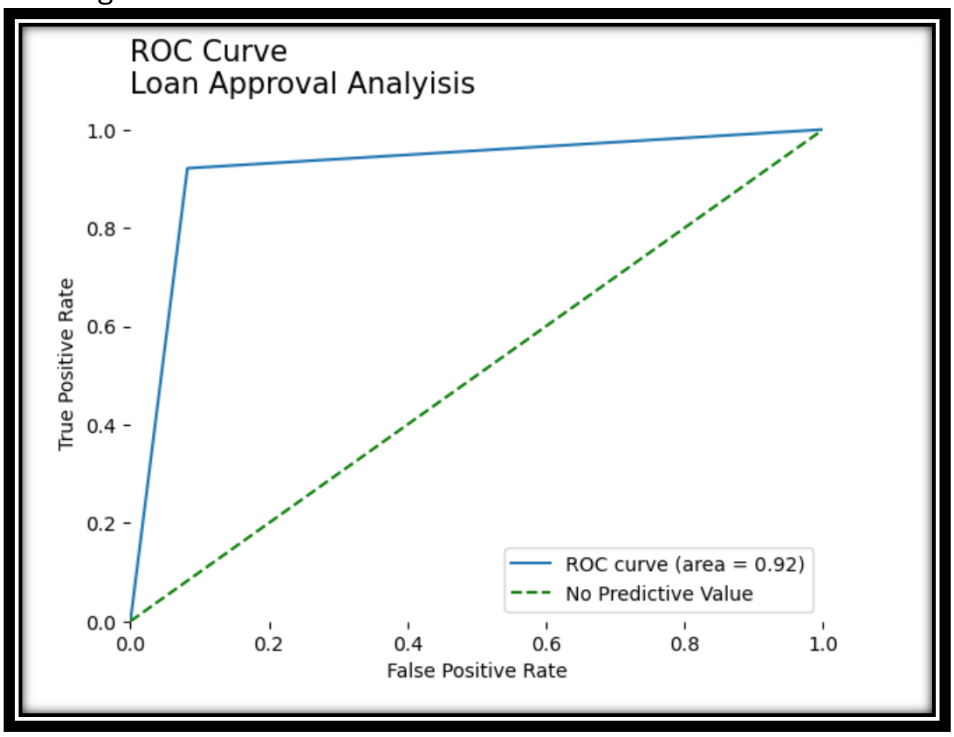
Confusion matrix for Linear Regression after feature reduction:



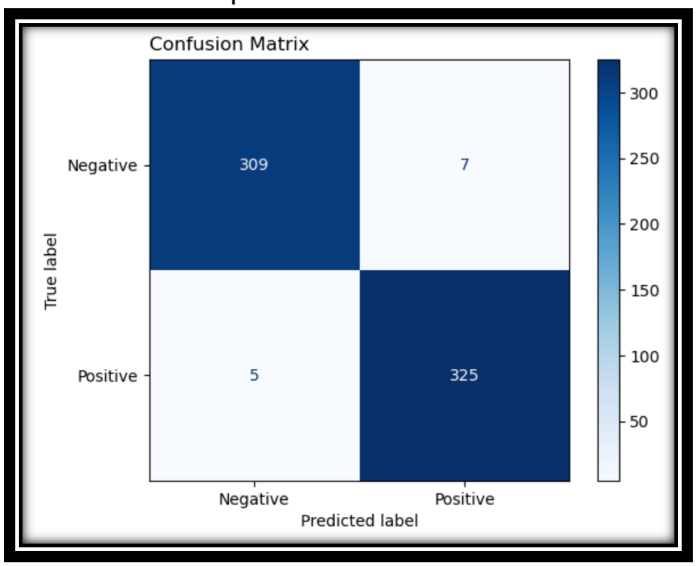
Accuracy report for Linear Regression after feature reduction:

	precision	recall	f1-score	support
0	0.92	0.92	0.92	316
1	0.92	0.92	0.92	330
accuracy			0.92	646
macro avg	0.92	0.92	0.92	646
weighted avg	0.92	0.92	0.92	646

ROC for Linear Regression after feature reduction:



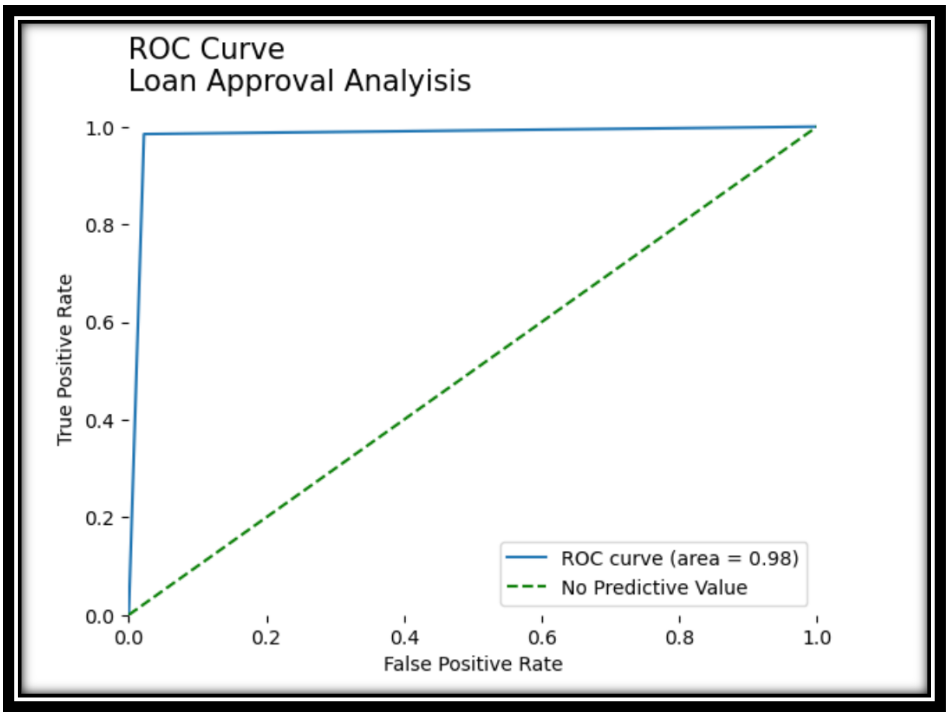
Confusion matrix for Random Forest prior to feature reduction:



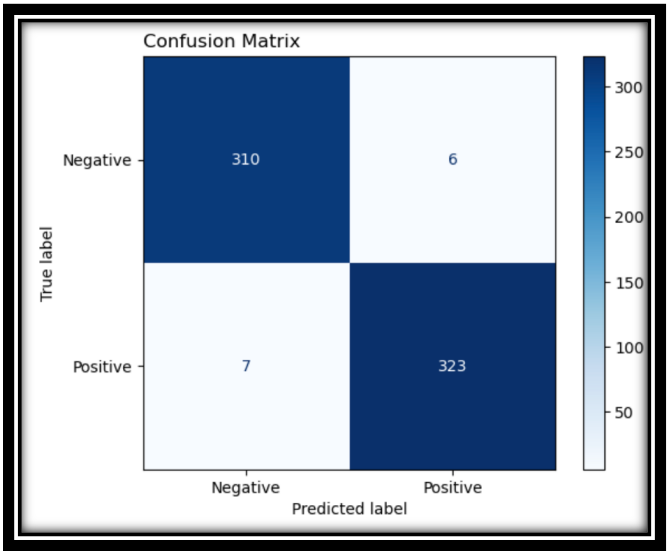
Accuracy report for Random Forest prior to feature reduction:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	316
1	0.98	0.98	0.98	330
accuracy			0.98	646
macro avg	0.98	0.98	0.98	646
weighted avg	0.98	0.98	0.98	646

ROC for Random Forest prior to feature reduction:



Confusion matrix for Random Forest after feature reduction:



Accuracy report for Random Forest after feature reduction:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	316
1	0.98	0.98	0.98	330
accuracy			0.98	646
macro avg	0.98	0.98	0.98	646
weighted avg	0.98	0.98	0.98	646

ROC for Random Forest after feature reduction: