Springboard Data Science
Capstone Project 3
Loan Risk Prediction

Paul Kim

July, 2022

1. Introduction

1.1)Background

In 2022, Americans owed $178 billion in personal loan debt and it is the highest in the last 17 years and the amount is continuously growing every year. However as the number of loan applicants increase, the number of denials of the applications from the bank or third party has increased as well. In 2021, 16.1 percent of loan applications were denied. There are multiple features lenders consider when they offer loans to clients. The main question they ask themselves as a lender is ‘Can the applicant pay the loan back within time?’ Having said that, credit score continues to be one of the most important features of the client’s financial status among the many factors that are used to measure the credit score of the applicant.

1.2) Problem Statement

As it is stated above, not all of the loan applicants are approved. In such a case, the applicants whose applications are rejected by the banks or third party lenders initially, usually prepare another application for reconsideration. However, once the first application is rejected, the detailed reasons for denials are not fully disclosed to applicants. Therefore, even if they prepare the required documents to prove their financial stability again, the chances of getting denied by the lenders remain high. As a result, the applicants end up wasting more time and money for unassured results. Therefore, this project aims to identify applicants with high risk of rejection according to the financial features with machine learning algorithms. Also, features that significantly affect the result of loan application compared to other features can be determined to have applicants prepared with applications that give higher chances.

1.3) Overview

The project’s goal is to develop machine learning models that successfully predict clients with risk of denial from the loan applications using their financial features. The developed models can be used by institutions where the client’s required documents are prepared. Also they can be used by e third-party lenders to attract clients who are not likely to be approved for loans from the banks. Lastly, it can be used to provide more detailed and precise counseling to clients on the reasons for denial and interventions that might lead to increase the probability of approval.

In this project, samples with financial features and loan application results were collected and different machine learning algorithms were built to identify the best performing model; and the model developed using the XGBoost classifier (with hyper-parameter tuning) achieved 84.23 percent of ROC-AUC score with 89.13 percent of mean accuracy score.

After analyzing the impact of features on our best model using the SHAP library[1] the most important features included whether the client owns a vehicle or not, the period the client lived in his/her house, the occupation of the client and the income and years of experience from the client's job.

Implementation details can be found in the notebooks and deliverables in the GitHub repository link below.
(https://github.com/paulkimDSN/Capstone-3-)

2. Approach

2.1) Data Acquisition & Wrangling

The raw dataset named 'Loan Prediction' that contains financial status and result of loan application of the samples was retrieved from the website Kaggle[2] in a csv format.

The dataset has 13 columns and 252,000 rows. Of these 13 columns, 7 columns have data type 'int64' and 6 columns have data type 'object'. Each row represents an applicant, and columns represent different financial features for applicants. The column names and the attribute information are listed below.

1) Id: Unique Identifier of the sample
2) Income: Annual income of the sample
3) Age: Age of the sample
4) Experience: Total years of working experience of the sample
5) Married / Single: Marital status of the sample
6) House_Ownership: House ownership status of the sample

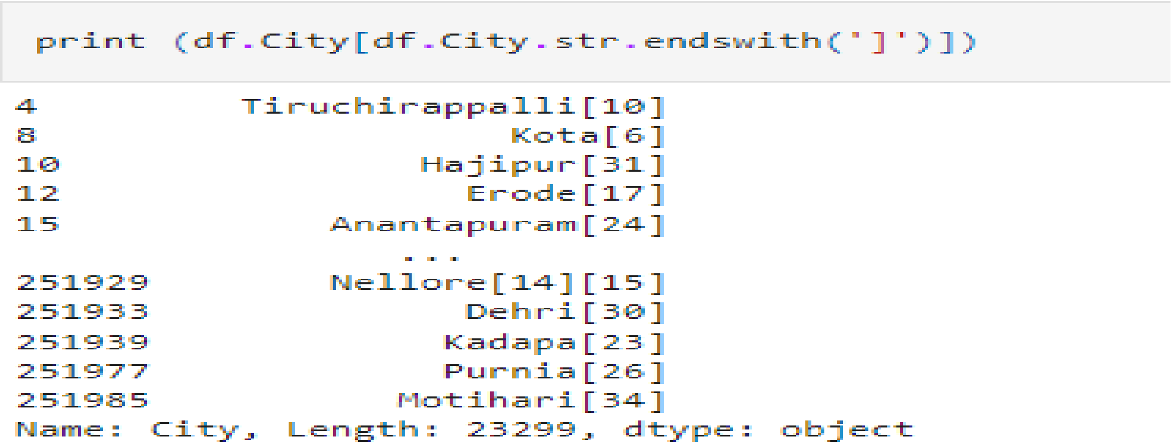
[1] https://shap.readthedocs.io/en/latest/index.html

[2] https://www.kaggle.com/datasets/subhamjain/loan-prediction-based-on-customer-behavior?select=Test+Data.csv

7) Car_Ownership: Vehicle Ownership status of the sample
8) Profession: Profession/Occupation of the sample
9) City: City the sample is currently residing in
10) State: State of the city the sample is currently residing in
11) Current Job yrs: Years of working experience in the current job of the sample
12) Current_House_yrs: The period of the sample residing in the current address
13) Risk_Flag 0: Loan Application Approved / Risk_Flag 1: Loan Application Rejected

After the dataset was loaded into a dataframe, initial inspection was conducted. The shape and data types of the dataset are inspected and numerical features are analyzed. Now, the null values of the dataset were checked. Fortunately, the dataset did not have any missing values in the columns.

Next, the column names are listed for any errors or misspelling and decided to change all the names to upper case letters for consistency. The same process was done for unique values and inspected as well. While inspecting the values of columns 'City' and 'State', we realized some of the city and state names contained random square brackets with numbers inside them at the end of values, as shown below.



```
print (df.City[df.City.str.endswith(']')])
```

```
4          Tiruchirappalli[10]
8                     Kota[6]
10                 Hajipur[31]
12                   Erode[17]
15              Anantapuram[24]
                       ...
251929          Nellore[14][15]
251933                   Dehri[30]
251939                  Kadapa[23]
251977                  Purnia[26]
251985               Motihari[34]
Name: City, Length: 23299, dtype: object
```

After we double checked that the numbers do not mean anything and it is not necessary for our project, a function that removes the square brackets and numbers inside them and replaces the values back to original columns was defined. The defined function was used for columns ‘City’ and ‘State’ and now the column values only represent the city and state names.

Finally, now the values are checked and cleaned, the categorical values are changed to upper case letter as well for consistency while visualizing the dataset. Now, the dataframe is ready for analysis and visualization, it has been exported to a file named ‘loan_data.csv’ for further use.

2.2) Exploratory Data Analysis

Exploratory data analysis was conducted to answer questions for machine learning development in the future steps. First, Exploratory Data Analysis was conducted on a jupyter notebook using different python packages including numpy, Pandas, Seaborn and Matploblib.

Another part of EDA was completed using Tableau and the Tableau link for the project can be found in this link.

First, we started with general visualization of the dataset to understand the overall picture of it.

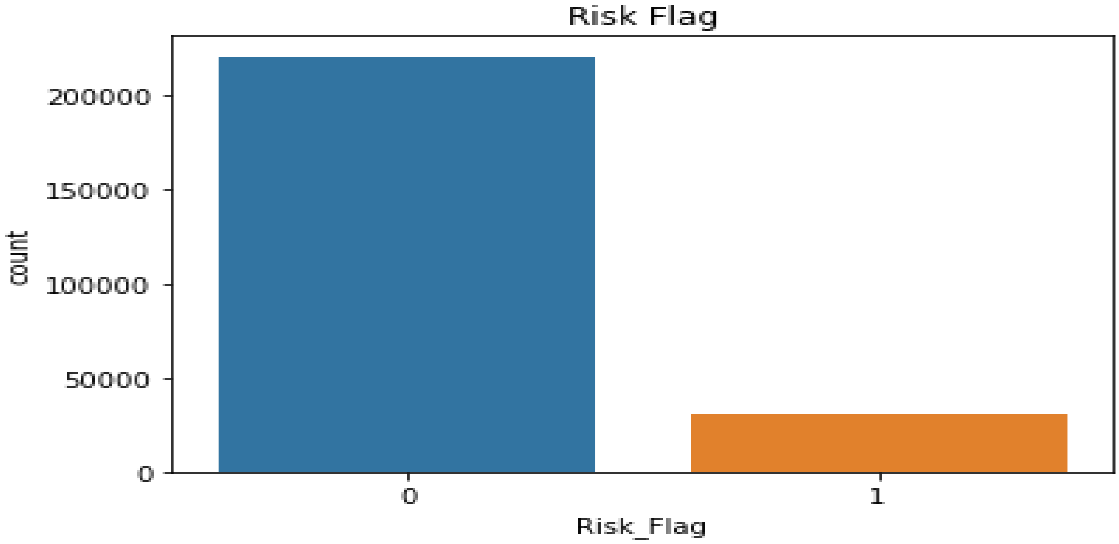


Figure 1:Bar plot: Distribution of Risk_Flag population

As the bar plot shows above, the proportion of the clients’ within the dataset was calculated. 87.7 percent of the sample was clients labeled with ‘No-Risk’ and only 12.3 percent of the sample size was labeled ‘Risk’. Therefore, most of the clients from the sample have been approved for loan. Since the column ‘Risk-Flag' is the target variable of the project therefore, the imbalance classification must be dealt with to produce a better performing model in the later steps of the project.

Next, distribution of age group in the dataset was visualized

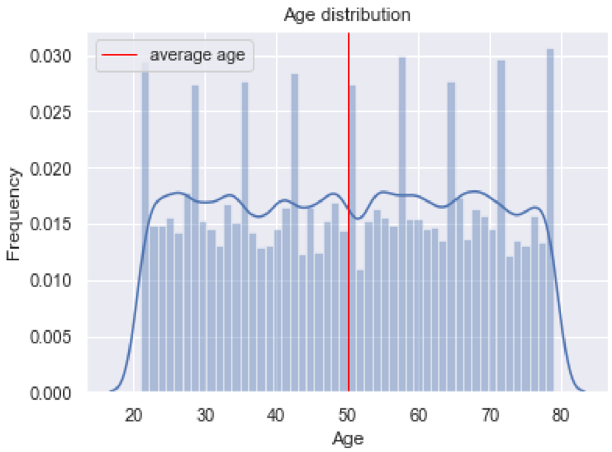


Figure 2:Age group distribution

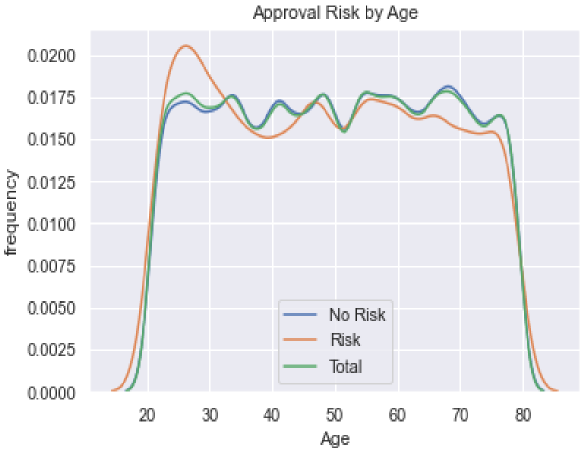
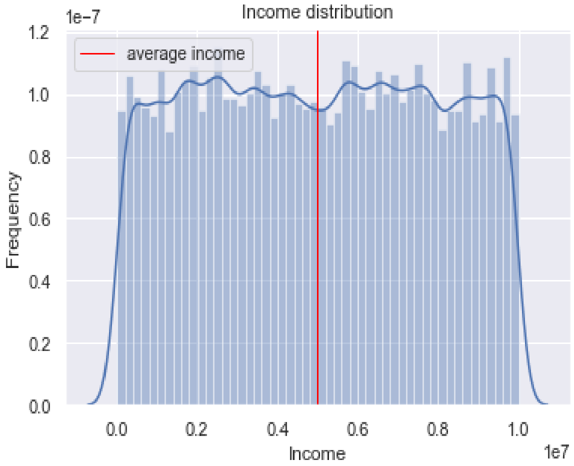


Figure 3: Age distribution by risk group

As figure 2 shows, the dataset was evenly distributed in terms of age group. The bar chart shows an average of 49.95, with minimum of 21 and maximum of 79.

Figure 3 shows where the risk flag occurs across the age groups. The orange line represents the frequency of risk occurrence, blue line represents no risk group and green line is the total population. As the plot shows, in the younger age group, from 20 to early 30s, the risk occurrence is at maximum. After it stays below the blue line across the plot. Therefore, it shows how there is a higher chance of rejection in younger age groups compared to older age groups.

Now, Income distribution was visualized.



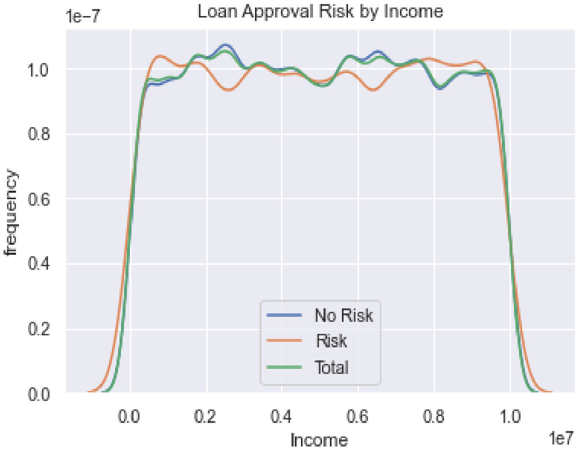


Figure 4: Income distribution

Figure 5: Income distribution by risk group

As figure 4 shows, income distribution of the dataset is also evenly distributed with mean income of 4997117. Also figure 5 shows how the risk group is distributed across the different income groups. As expected, the low income group had a higher chance of loan rejection. However, surprisingly, the extremely high income group also had higher chances of loan rejection. Therefore, only the start and end of each plot had a higher chance of risk and overall, it stayed below the no-risk group.

After, box plots of each numerical column are plotted to check for any outliers and visualize the overall distribution.

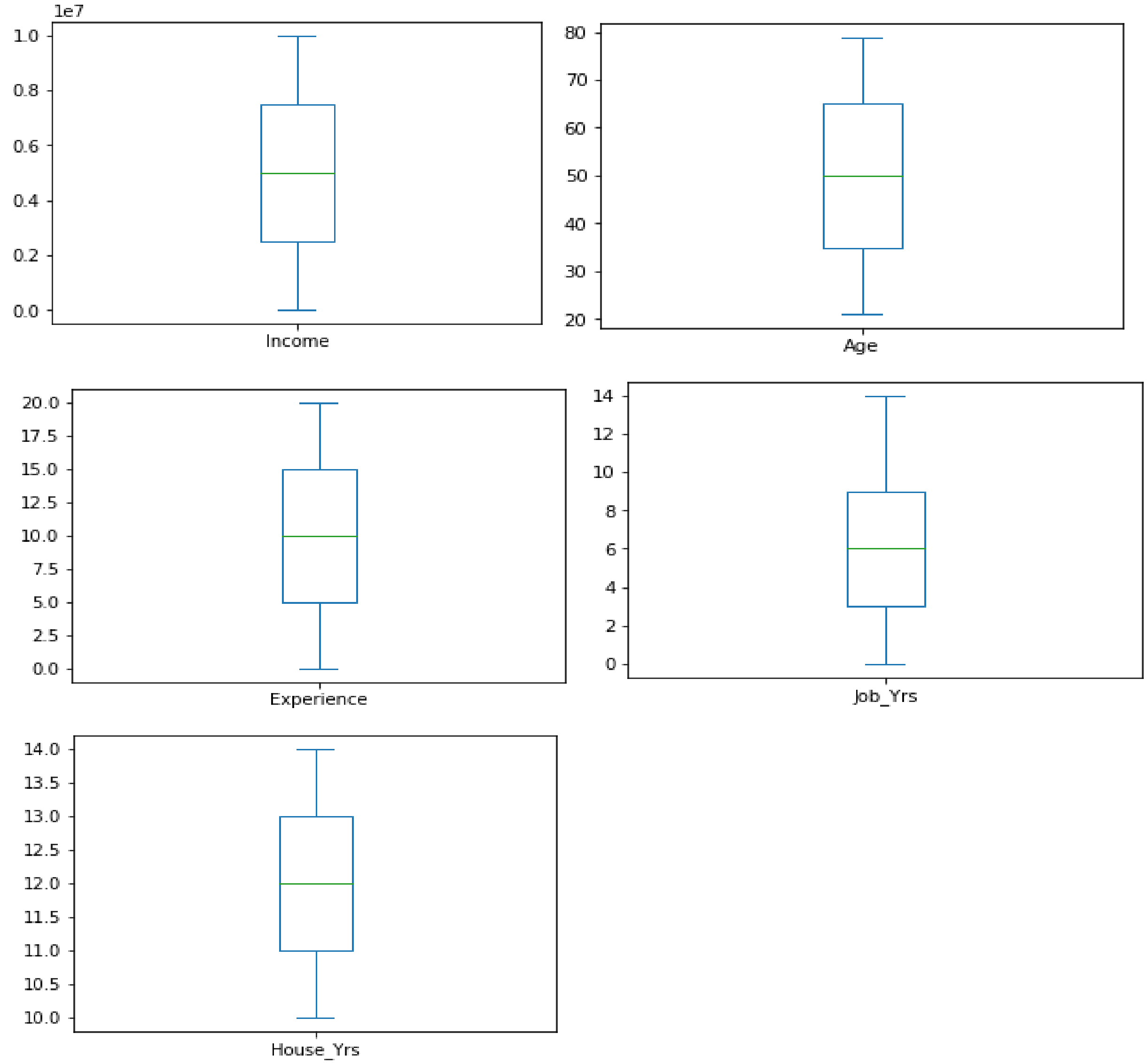


Figure 6: Box Plots of numerical features.

The box plots of numerical features mostly had similar shapes and no significant outliers were found. The shape of the boxes, those representing IQR (inter quartile range) also had similar shape except 'Job_years'. It had a slightly smaller shape with lower average compared to median.

Next, categorical columns are visualized by risk flag groups.

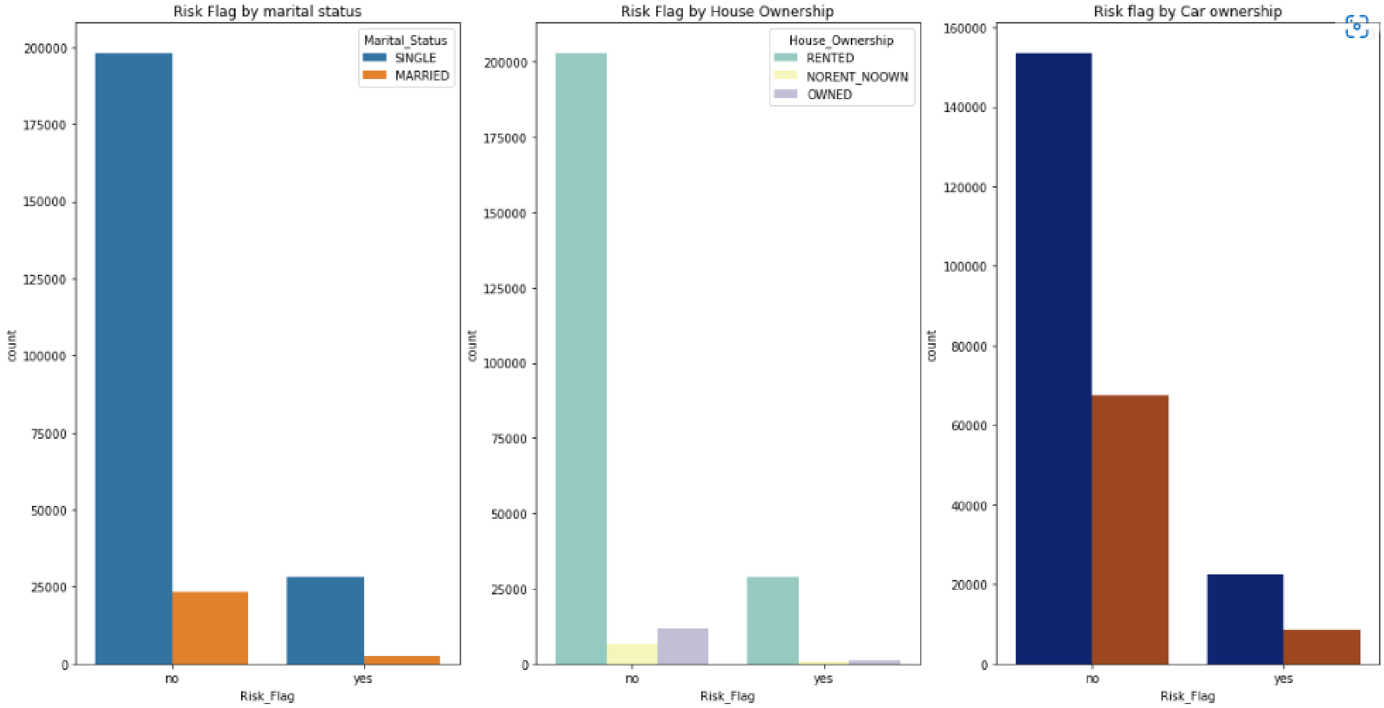


Figure 7: Bar plot, categorical columns by risk group

The three bar plots include risk flag by marital status, houser ownership, car ownership. The bar plot was created to compare how the proportion of each categorical feature varies within each class of the target variable. As expected, all three categorical features share similar proportions in two classes of our target variable. About 10:1 ratio is present in marital status in both class ‘no-risk’ and ‘risk’ groups. There are more clients who are single compared to married clients in both classes, ‘no-risk’ and ‘risk’ group. Also, most of the clients are living with their house rented and only a small proportion of clients have their house owned or not rented/not owned. Again the proportion in both classes ‘risk’ and ‘no-risk’ were the same. Lastly, the risk flag by car ownership was plotted. There are more clients without cars in both classes and the proportion was about the same for both classes as well.

After, to visualize the detailed picture of the proportion within the ‘Risk’ class, pie charts are created for categorical columns with ‘Risk-Flag’ class.

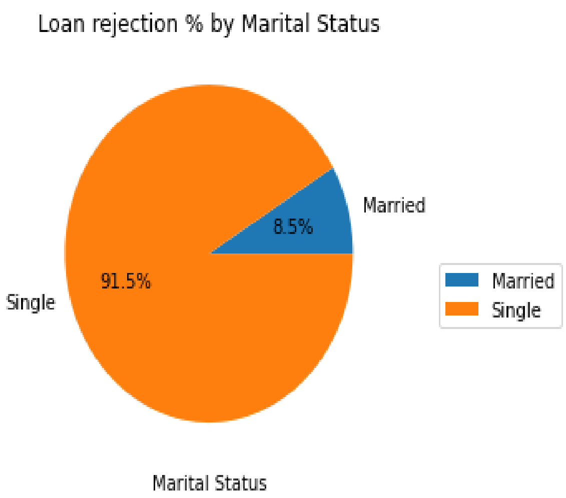


Figure 8: ‘Risk’ class by marital status

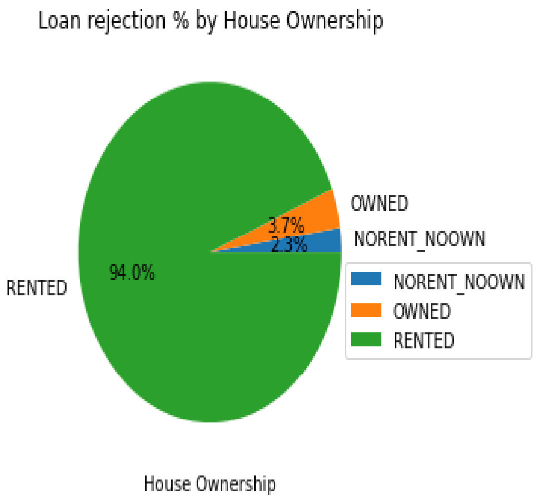


Figure 9: ‘Risk’ class by house ownership

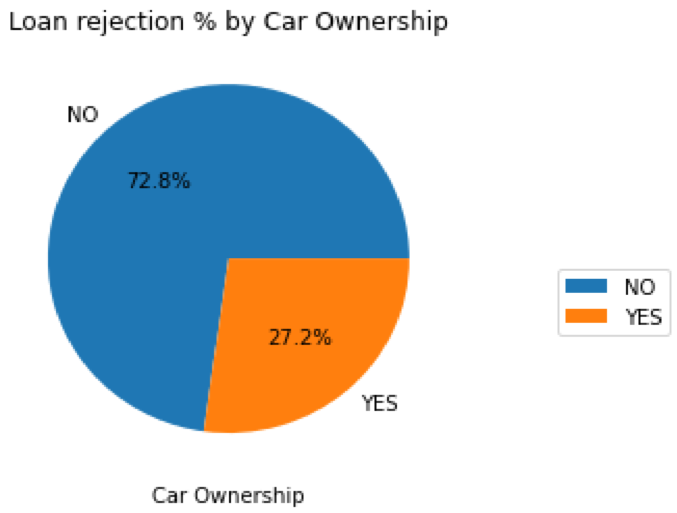


Figure 10: ‘Risk’ class by Car ownership

As the pie charts above represent, 91.5 percent of the ‘Risk’ class are single clients, in terms of house ownership 94 percent of rejected clients live their house rented, only 3.7 percent, and 2.3 percent live their house owned and not rented/not owned respectively. For car ownership, the proportion is not as extreme as other features. 72.8 percent of rejected clients do not own vehicles and 27.2 percent of clients own a vehicle. Analyzing categorical features is important since it helps understand the financial status of the clients and its relationship to our target variable.

Next, the dataset was visualized by profession. The profession is an important feature to measure the financial status of the client.

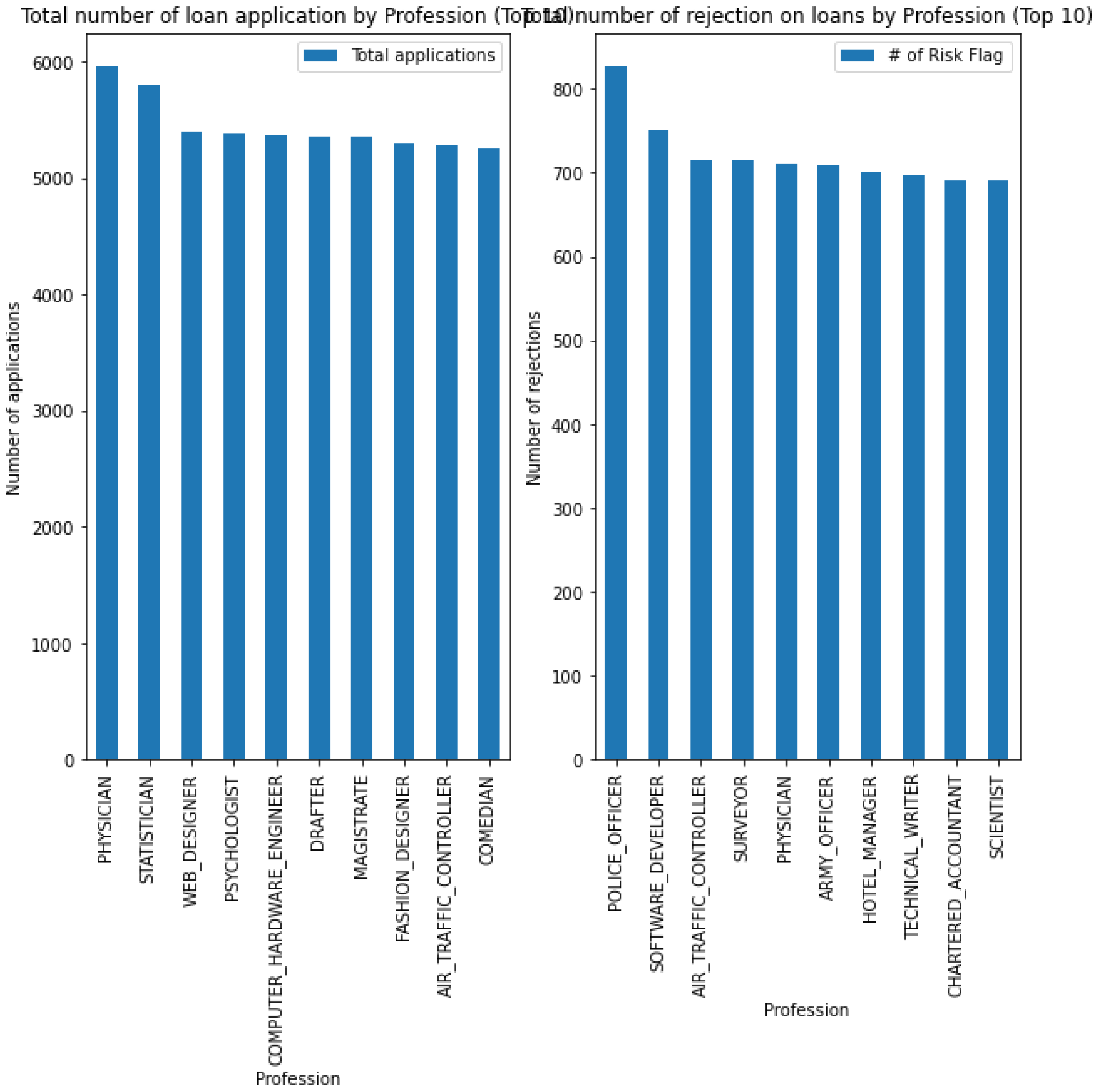


Figure 11: bar plots, total applications and rejections by profession (top 10)

Figure 11 shows the top 10 professions with the most number of applications made, and the top 10 professions that are rejected the most of the applications made. The initial expectation was, a profession with more applications will have more rejections, however, the result was different from what we expected. The profession with the most number of applications was “Physician”; however, it is ranked fifth on the second bar plot. Also, other than physicians from the left bar plot, only air traffic controllers are present on the top 10 list of rejections. Therefore, in terms of profession, an increase in the number of applications does not necessarily mean an increase in number of rejections. Therefore, more inspection on professions was done with income by professions.

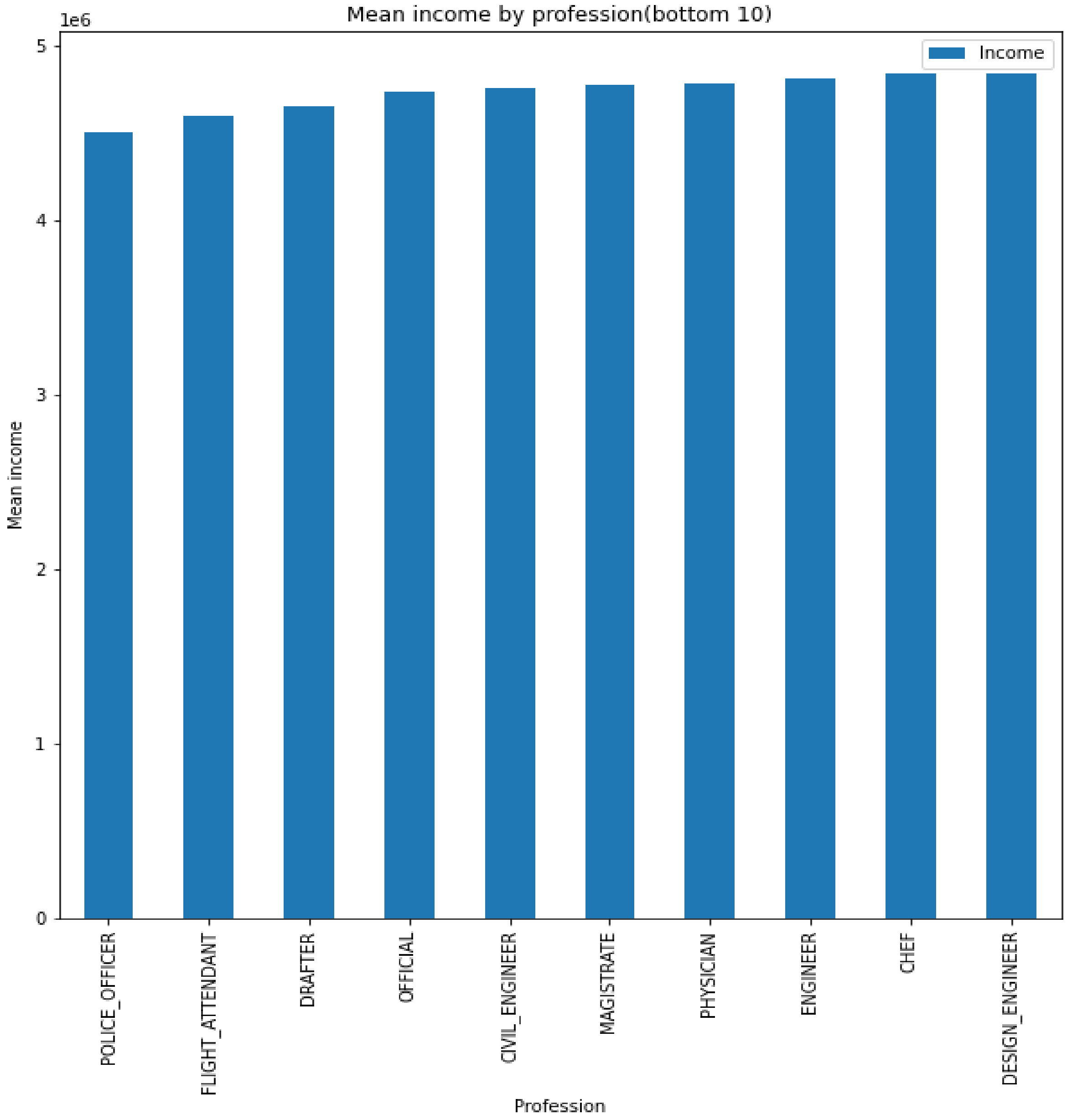


Figure 12: Bar plot, bottom 10 income list by profession

As the figure 12 shows, the 10 most rejected professions are more similar to the bottom 10 income list of professions. Therefore, clients' loan applications are rejected due to low income, not profession.

We also took a look at the heatmap of the correlation coefficient to determine if any features are highly correlated to each other.

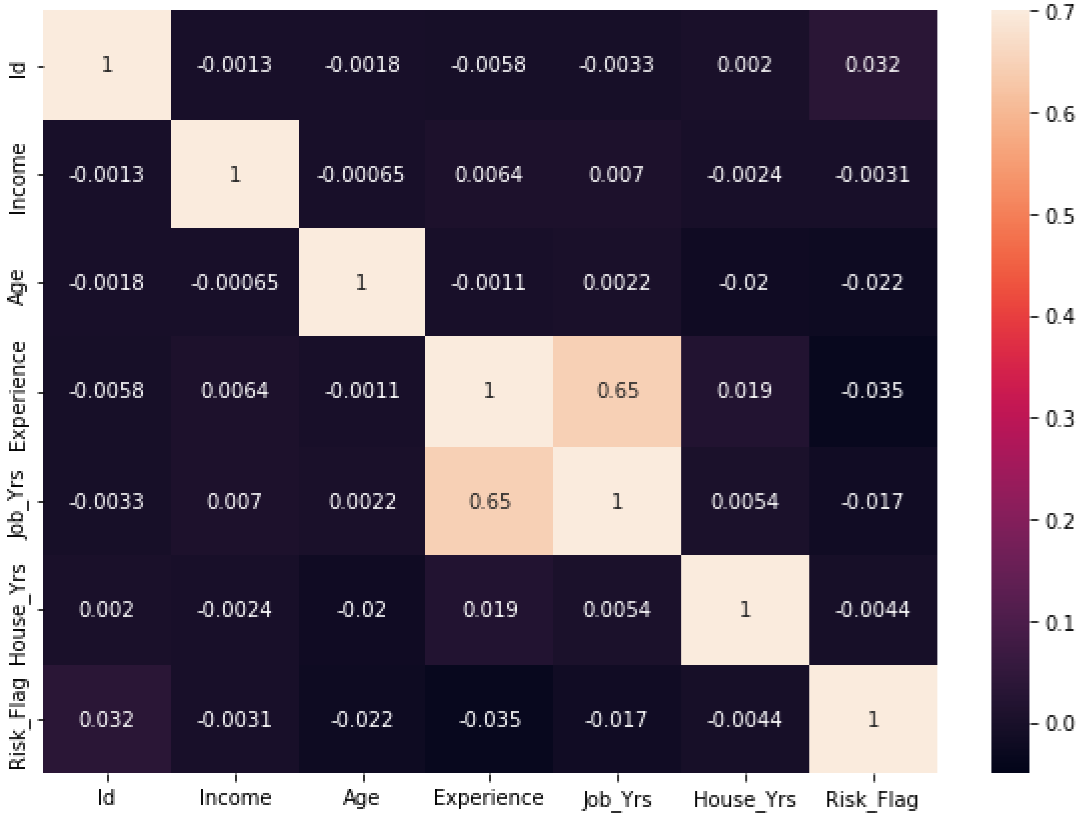


Figure 13: Heatmap, correlation coefficient to each variable.

The correlation heatmap was plotted and it can be observed that, in general, there is not a high correlation between pairs of features in the diagram in Figure 13 features. The highest absolute value was 0.65 between job years and experience. Since many of the clients working at the same work place for their whole career or clients with their first job will have the same number for job years and experience years. Other features do not have strong correlation since the next highest absolute value was 0.0064 between experience and income.

Instead, we decided to take a look at the relationship between income, age and risk flag. As figure 12 and 13 show, it is hard to observe any kind of relationship between risk flag and profession. However, we were able to observe that the list of professions with low income and the list of professions with most rejections share a similar list of professions. Therefore, a more detailed relationship between each other was analyzed. First we divided the datasets into two groups, one with above average income and the other group with below average income.

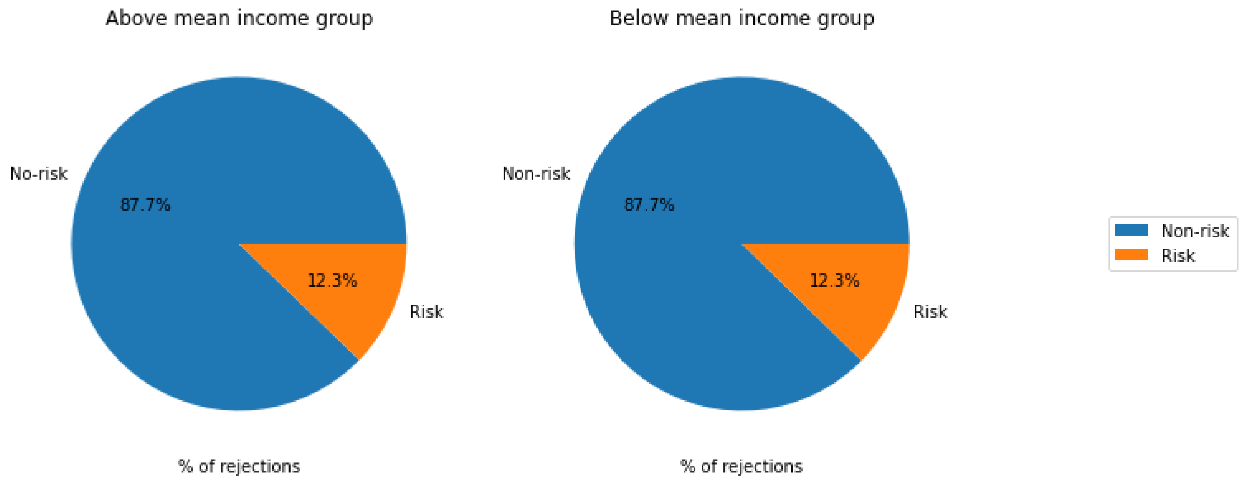


Figure 14: Pie charts, risk flag occurrence by above and below average income.

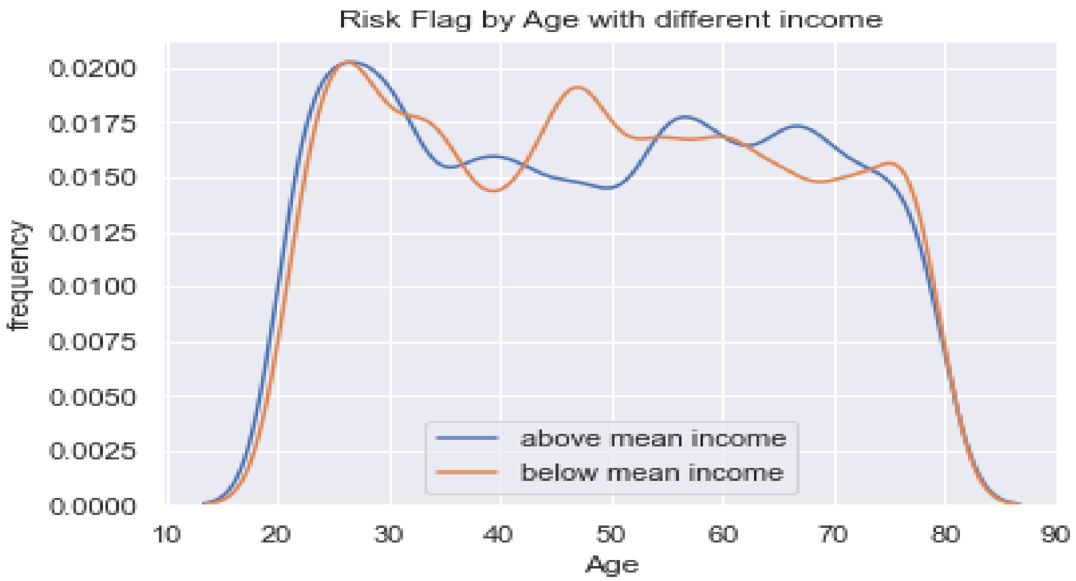


Figure 15: KDE plot, risk flag occurrence by above and below average income across different age groups

As pie charts show, the proportion of clients with risk and without risk was exactly the same in both, above and below average income groups with 87.7 percent of approval and 12.3 percent of denial. It is hard to conclude that income has anything to do with the risk flag in this case.

Also, figure 15 shows that both groups have about the same frequency and start to drop as they get older. Up to the early 50s, the below average income group has overall higher frequency compared to the above average income group however, as they get older, the higher income group's frequency stays above the lower income group until the oldest age group. The plot cannot tell us much but we can conclude that for young aged groups, and low income mid aged groups have a negative effect on loan approval rate.

Now, the most important features of the project were explored, the 'city' and 'state' features. One of the main goals of the exploratory data analysis was to answer the effects of 'city' and 'state' features to our models.

Let's see if a city differentiates itself from another city in the same state in terms of characteristics of population, including income, size of population and demographic. It might not be a good idea to develop one single machine learning model for the whole state, instead, it might be better to separate them by cities and develop models for each of them to boost the performance. Therefore it is important to analyze if our datasets' states and cities have significantly different characteristics and decide if developing models by city or state is necessary for our project.

The EDA of city and state are mostly conducted on Tableau, therefore the link provided above can be accessed for analysis.

First, the population of each city and state is plotted.

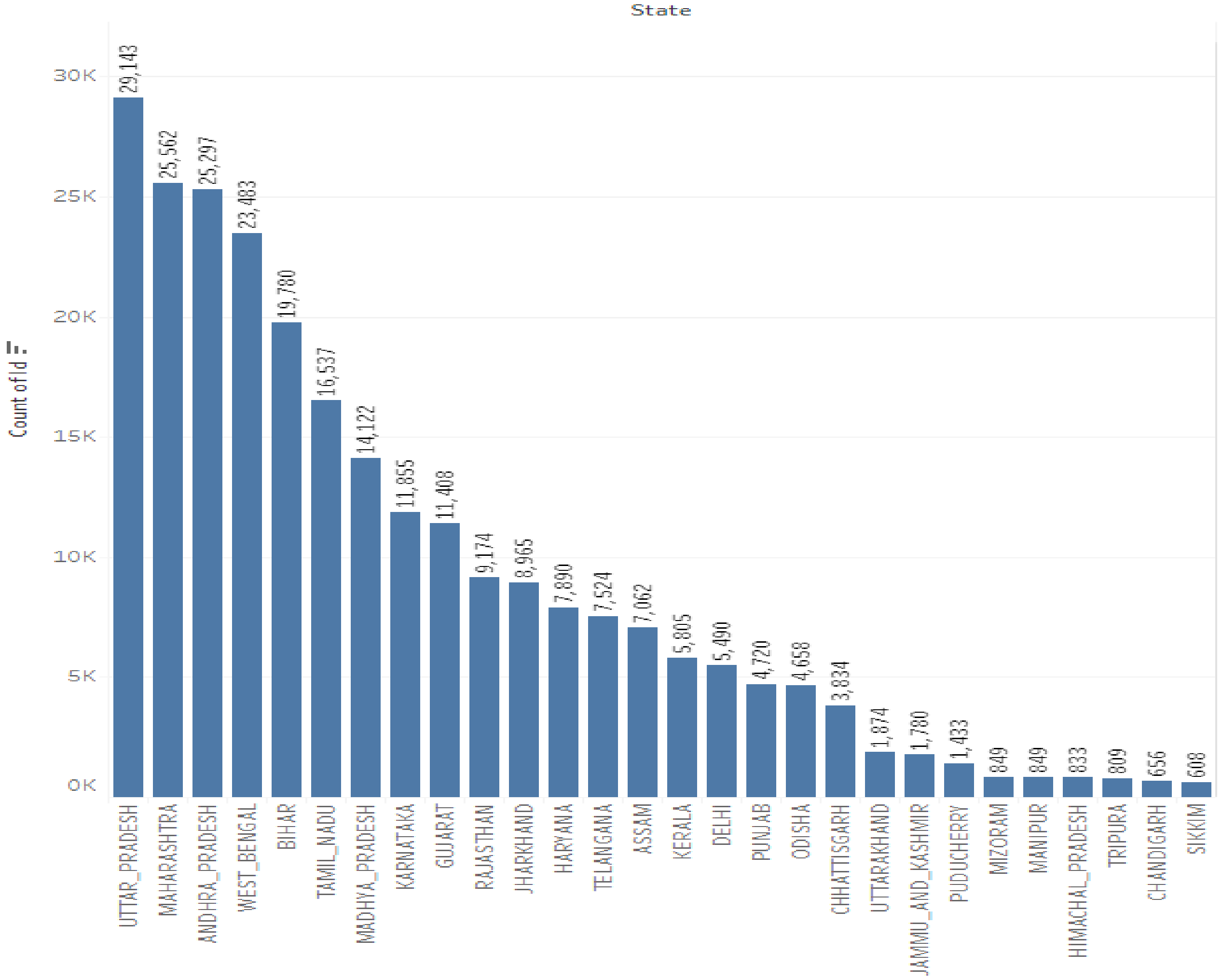


Figure 16: Bar plot, Population distribution by State

The bar plot above shows the distribution of population by state with the maximum of 29,143 and minimum of 608. If we only look at the figure, it looks like the population varies a lot by the states. However, the same plot for cities was plotted (can be found on Tableau link), and the number of cities in each state varies but the number of population within each city does not vary as much. Therefore we can conclude that the number of populations within cities are similar however, the number of cities within states vary.

Next, age and income distribution by city and state are plotted. Again, there are too many numbers of cities in our dataset, therefore it is attached on the link (Tableau link).

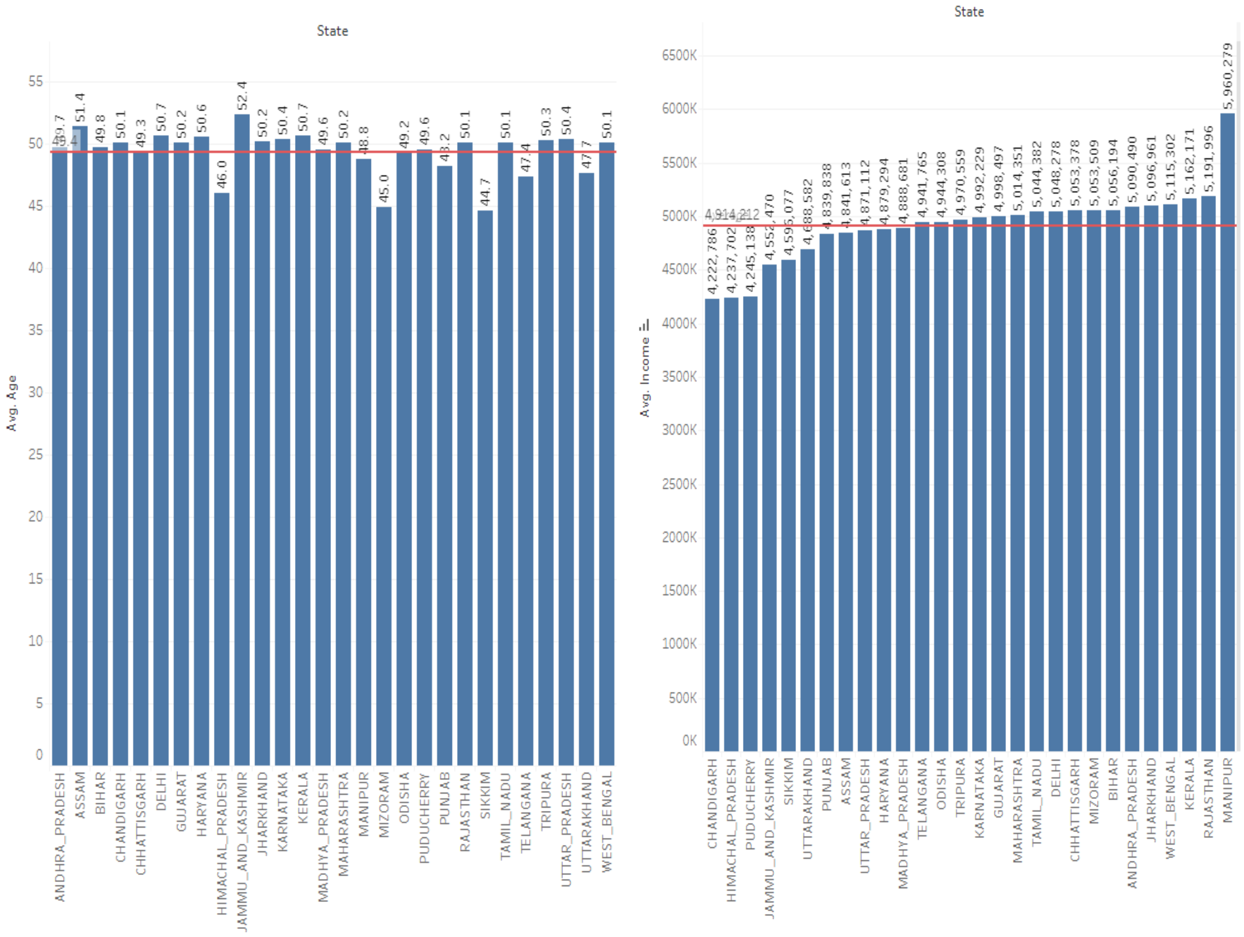


Figure 17: Bar plot, average income and average age distribution by states.

As the figure 17 shows, the average income and average age of each state are very similar. The red line represents the average number and most of the bars are around the line and it can be concluded that all the states' are evenly distributed in terms of age and income.

As it was discussed above, age and income are the important features that define the demographics and characteristics of each city and state. From what we see above and the distribution by city(from the link), the country shares the similar demographics of the population across the nation so far.

Next, our target variable, risk flag, by city and state was plotted. The point of looking at other distributions is to make sure that the proportion of our target variable within city and state are not affected. Therefore if the proportion of the target variable is all different by city and state, it probably indicates that different demographic characteristics are causing it and separate model development is necessary.

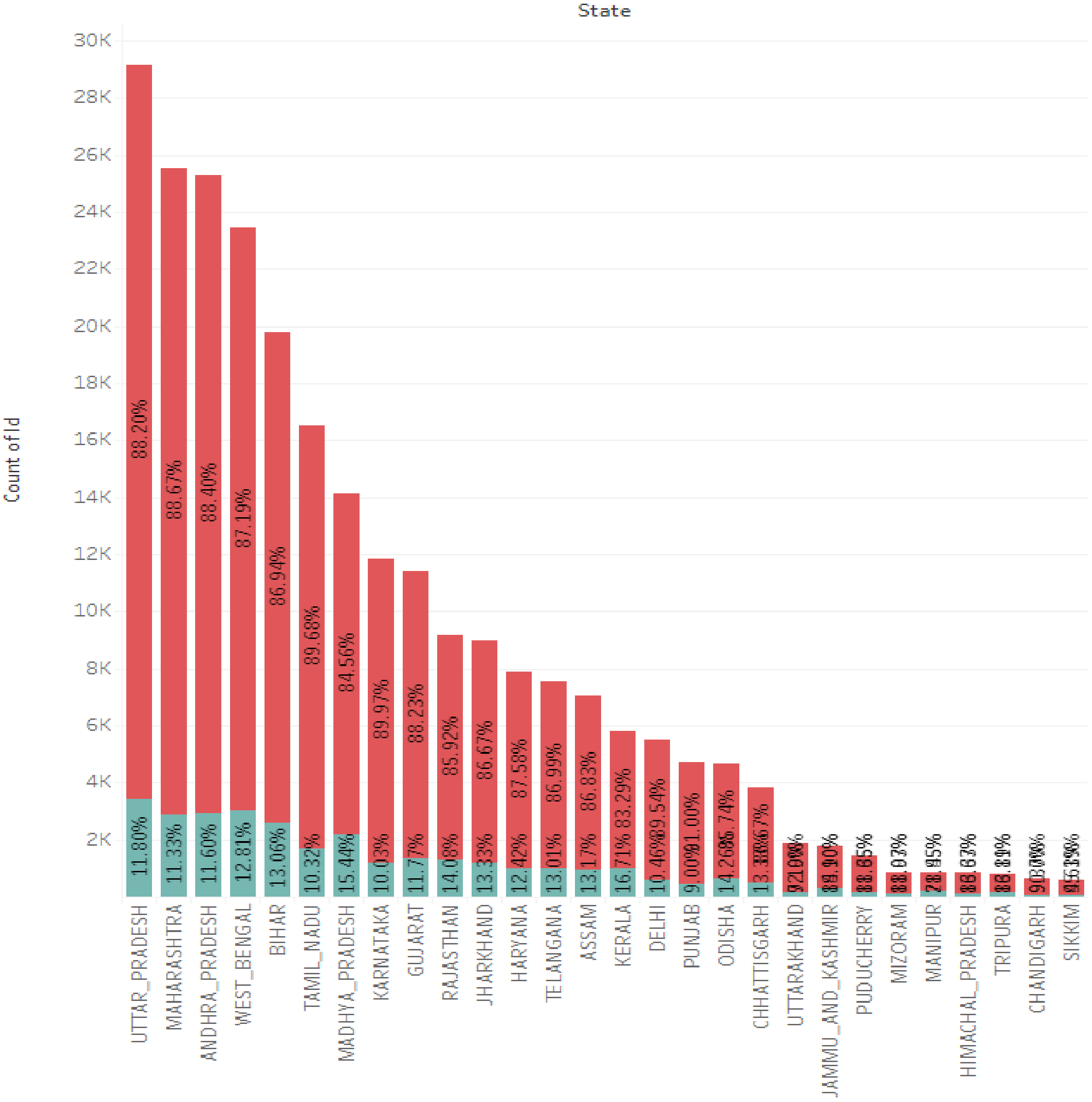


Figure 18: Risk Flag proportion by state;
Where red means Risk present, and green means Risk absent

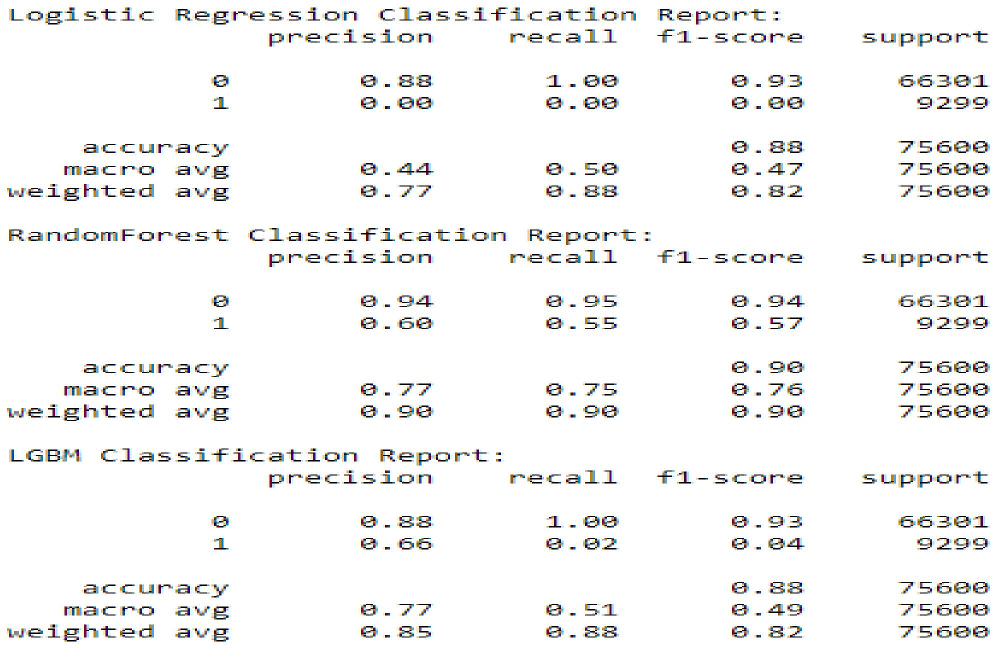
As the graph shows above, the proportion of risk flags within states are similar. The risk flag proportion mostly lies between 10-15 percent across the whole nation. It is also the same case for risk flag proportion by city. Therefore it can be concluded that there are no significant demographic characteristics that cause a huge variability by city or state.

After conducting exploratory data analysis of our dataset, we can conclude that there is no significant correlation between features by looking at the heatmap of correlation coefficients and other plots and charts. Also, as the main goal of EDA for our project was to answer if it is necessary to develop models for each city or state, distributional and proportional analysis are conducted. In conclusion, it is hard to say there is a strong variability in terms of demographic between city/state that causes high variation of our target variable proportion and separate modeling is not necessary for our project.

2.3) Baseline Modeling

For baseline modeling, logistic regression, random forest classifiers, and LGBM classifiers are used on two types of dataset. First dataset with ‘City’ and ‘State’ columns and second dataset without ‘City’ and ‘State’ columns. We have concluded that separate model development for each city and state is not necessary and now we are checking if the model performance shows improvement or consistency without ‘City’ and ’State’. Since including the two columns may lead to unexpected ‘noise’, therefore if it performs better without them, it is advantageous for us to omit them from the dataset.

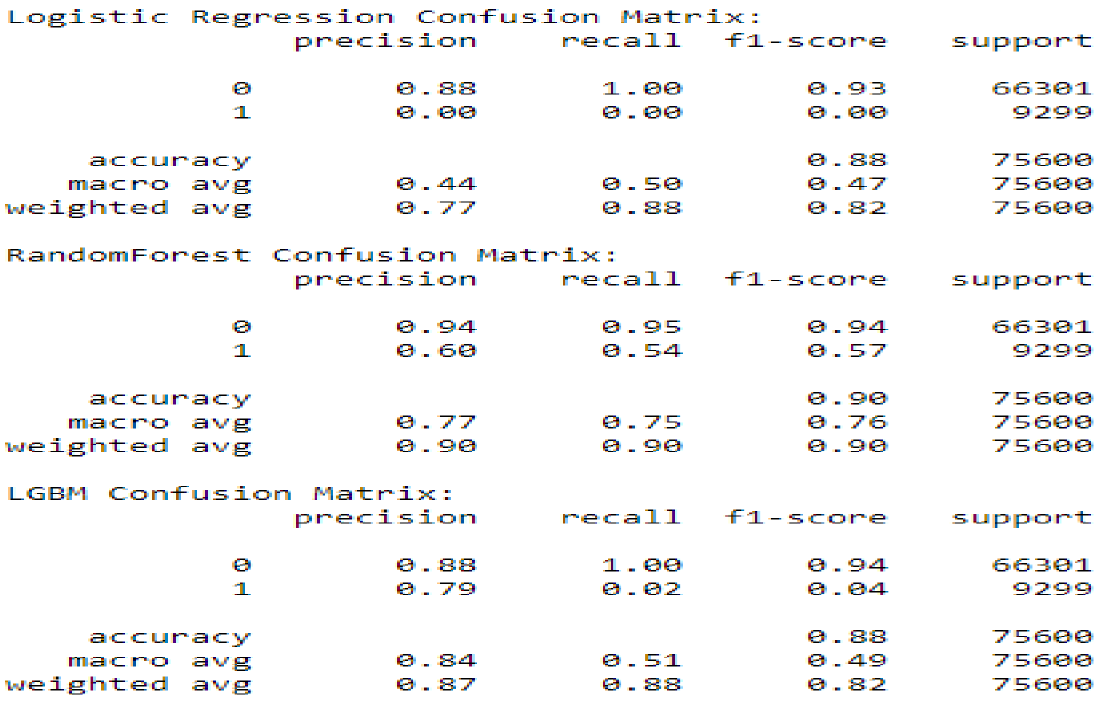
First, baseline modeling for the dataset with ‘City’ and ‘State’ was developed and results are shown below with the (stratified) test set.



```
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.88      1.00      0.93     66301
           1       0.00      0.00      0.00      9299

    accuracy                           0.88     75600
   macro avg       0.44      0.50      0.47     75600
weighted avg       0.77      0.88      0.82     75600

RandomForest Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.95      0.94     66301
           1       0.60      0.55      0.57      9299

    accuracy                           0.90     75600
   macro avg       0.77      0.75      0.76     75600
weighted avg       0.90      0.90      0.90     75600

LGBM Classification Report:
              precision    recall  f1-score   support

           0       0.88      1.00      0.93     66301
           1       0.66      0.02      0.04      9299

    accuracy                           0.88     75600
   macro avg       0.77      0.51      0.49     75600
weighted avg       0.85      0.88      0.82     75600
```

Figure 19: Classification Report of baseline models (with ‘City’, ‘State’)

After, the columns ‘City’ and ‘State’ were removed from the dataset and the new dataset was used to develop other baseline models.

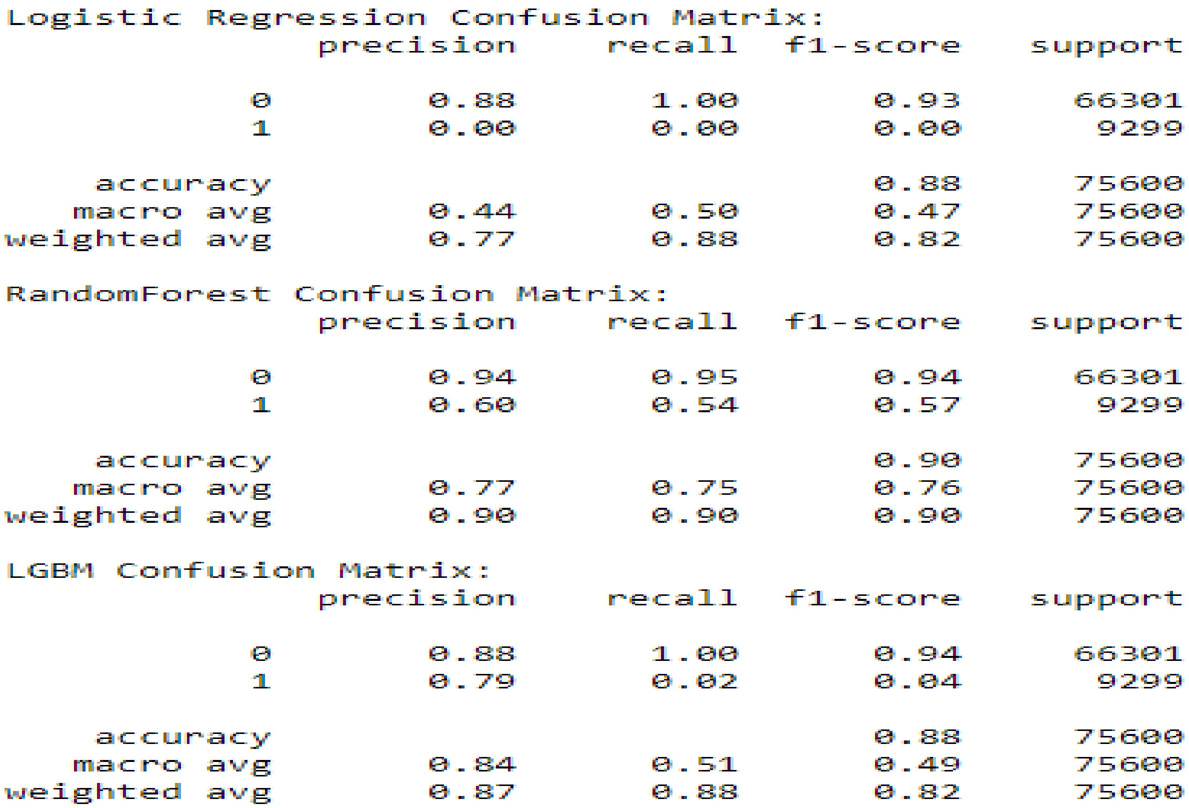


```
Logistic Regression Confusion Matrix:
              precision    recall  f1-score   support

           0       0.88      1.00      0.93     66301
           1       0.00      0.00      0.00      9299

    accuracy                           0.88     75600
   macro avg       0.44      0.50      0.47     75600
weighted avg       0.77      0.88      0.82     75600

RandomForest Confusion Matrix:
              precision    recall  f1-score   support

           0       0.94      0.95      0.94     66301
           1       0.60      0.54      0.57      9299

    accuracy                           0.90     75600
   macro avg       0.77      0.75      0.76     75600
weighted avg       0.90      0.90      0.90     75600

LGBM Confusion Matrix:
              precision    recall  f1-score   support

           0       0.88      1.00      0.94     66301
           1       0.79      0.02      0.04      9299

    accuracy                           0.88     75600
   macro avg       0.84      0.51      0.49     75600
weighted avg       0.87      0.88      0.82     75600
```

Figure 20: Classification Report of baseline models (without ‘City’, ‘State’)

As the confusion matrices of both baseline models show, the performance of the two baseline models are very similar and almost have exact same scores. Therefore we concluded that removing ‘City’ and ‘State’ from our model development is more beneficial for us in terms of model performance.

Next, a dataset without ‘City’ and ‘State’ is used to conduct a hyperparameter tuning of the baseline models.



```
Logistic Regression Confusion Matrix:
              precision    recall  f1-score   support

           0       0.88      1.00      0.93     66301
           1       0.00      0.00      0.00      9299

    accuracy                           0.88     75600
   macro avg       0.44      0.50      0.47     75600
weighted avg       0.77      0.88      0.82     75600

RandomForest Confusion Matrix:
              precision    recall  f1-score   support

           0       0.94      0.95      0.94     66301
           1       0.60      0.54      0.57      9299

    accuracy                           0.90     75600
   macro avg       0.77      0.75      0.76     75600
weighted avg       0.90      0.90      0.90     75600

LGBM Confusion Matrix:
              precision    recall  f1-score   support

           0       0.88      1.00      0.94     66301
           1       0.79      0.02      0.04      9299

    accuracy                           0.88     75600
   macro avg       0.84      0.51      0.49     75600
weighted avg       0.87      0.88      0.82     75600
```

Figure 21: Classification Report after hyperparameter tuning of baseline models

The classification report above in figure 21 shows how the baseline models perform after the hyperparameter tuning. However, the model performance has not improved significantly or it can be concluded that there was no to slight improvement. It is because of imbalance classification of the target variable in our dataset. As it was explored, the dataset’s proportion of ‘Risk-Flag’ is about 87.7 percent to 12.3 percent.

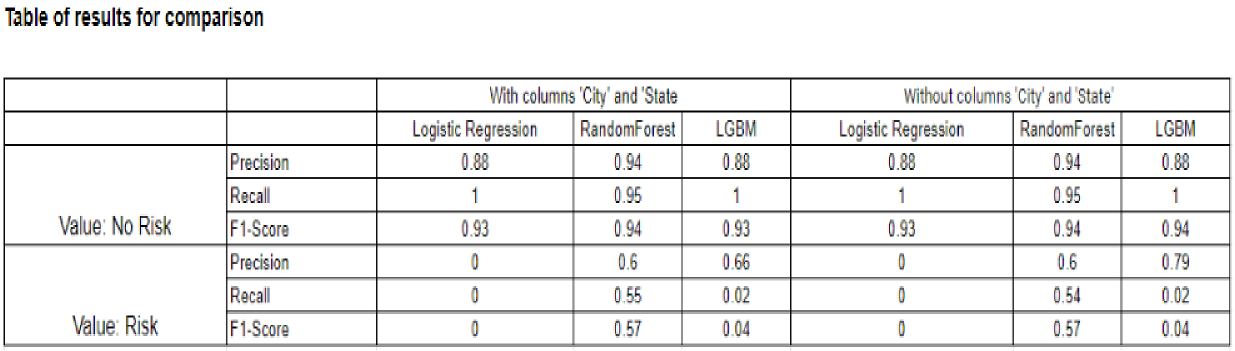


Table of results for comparison

		With columns 'City' and 'State			Without columns 'City' and 'State'		
		Logistic Regression	RandomForest	LGBM	Logistic Regression	RandomForest	LGBM
Value: No Risk	Precision	0.88	0.94	0.88	0.88	0.94	0.88
	Recall	1	0.95	1	1	0.95	1
	F1-Score	0.93	0.94	0.93	0.93	0.94	0.94
Value: Risk	Precision	0	0.6	0.66	0	0.6	0.79
	Recall	0	0.55	0.02	0	0.54	0.02
	F1-Score	0	0.57	0.04	0	0.57	0.04

Figure 22: Table of results of baseline models.

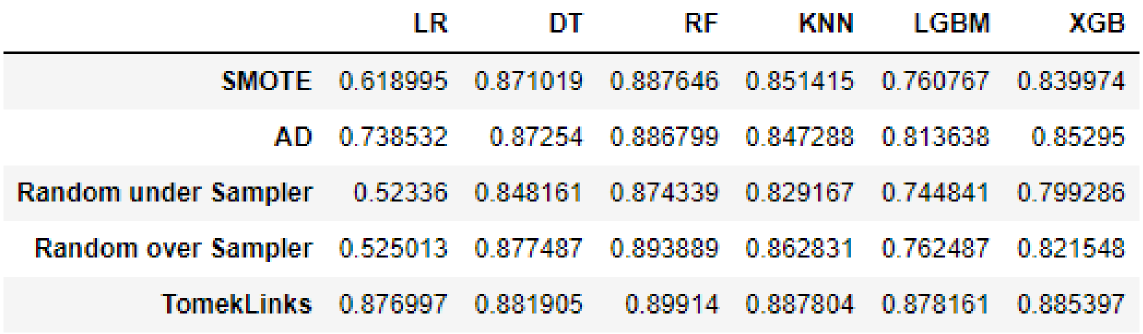
As the table shows, the model performance without ‘City’ and ‘State’ shows consistency but no improvement. Therefore, we decided to use the dataset without the two columns to minimize the ‘noise’ created from the model development.

2.3) Extended Modeling

Now with a new dataset without two columns, ‘City’ and ‘State’ are used to conduct an extended modeling. First, to deal with the imbalanced dataset, different resampling techniques are applied in combination with different machine learning classifiers.

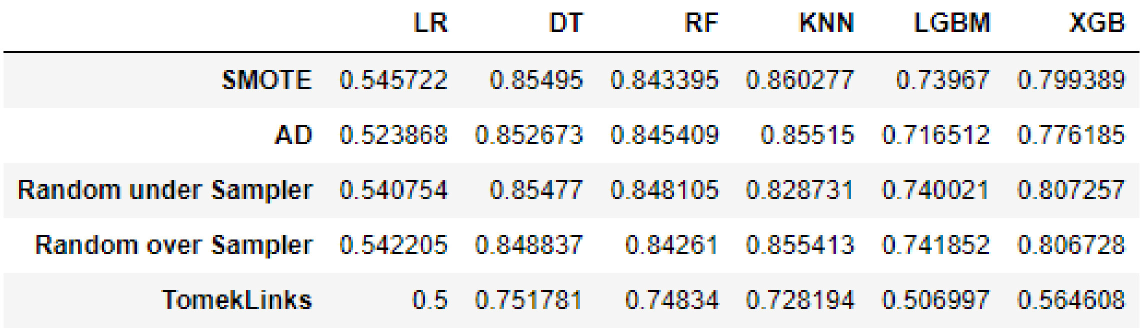
Types of resampling techniques we used include, SMOTE oversampling[3], Adasyn Oversampling[4], Random Oversampling[5], Random Undersampling[6] and TomekLinks[7] are applied on Logistic Regression, DecisionTrees, RandomForest, KNN, LGBM, and XGBoost classifiers. With all the combinations, accuracy scores and ROC-AUC scores are calculated for each of them.

[3] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
[4] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.ADASYN.html
[5] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html
[6] https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html
[7] https://imbalanced-learn.org/dev/references/generated/imblearn.under_sampling.TomekLinks.html



	LR	DT	RF	KNN	LGBM	XGB
SMOTE	0.618995	0.871019	0.887646	0.851415	0.760767	0.839974
AD	0.738532	0.87254	0.886799	0.847288	0.813638	0.85295
Random under Sampler	0.52336	0.848161	0.874339	0.829167	0.744841	0.799286
Random over Sampler	0.525013	0.877487	0.893889	0.862831	0.762487	0.821548
TomekLinks	0.876997	0.881905	0.89914	0.887804	0.878161	0.885397

Figure 23: Accuracy Score of resampling techniques on classifiers



	LR	DT	RF	KNN	LGBM	XGB
SMOTE	0.545722	0.85495	0.843395	0.860277	0.73967	0.799389
AD	0.523868	0.852673	0.845409	0.85515	0.716512	0.776185
Random under Sampler	0.540754	0.85477	0.848105	0.828731	0.740021	0.807257
Random over Sampler	0.542205	0.848837	0.84261	0.855413	0.741852	0.806728
TomekLinks	0.5	0.751781	0.74834	0.728194	0.506997	0.564608

Figure 24: ROC-AUC Score of resampling techniques on classifiers

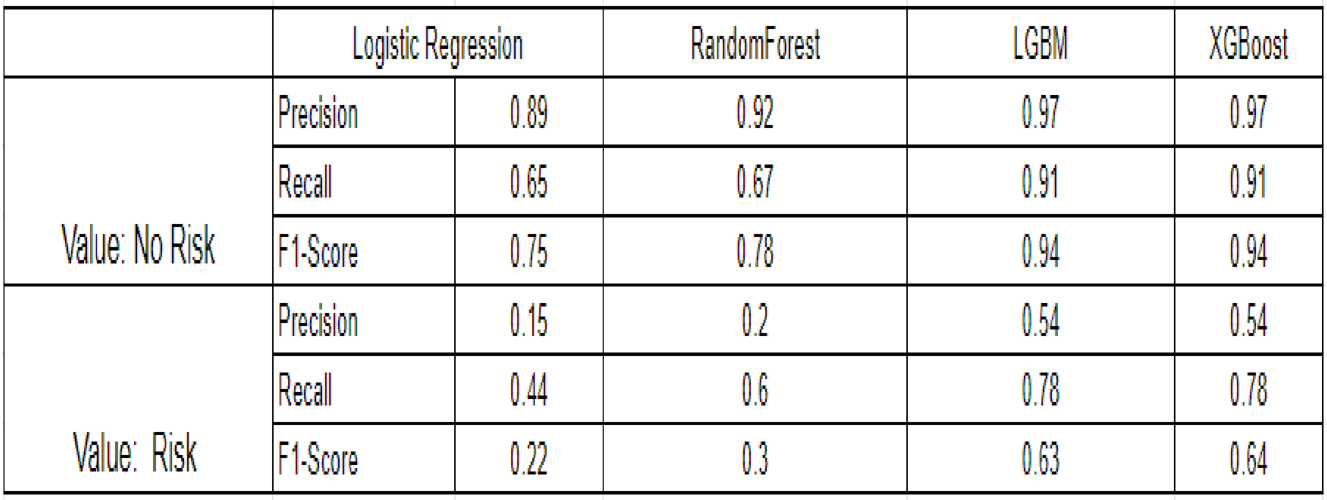
A pipeline that stores scores of every combination in the tables is created and the outputs are printed out as shown on the figures above. The tables and classification report of every classifier and resampling technique combinations, the best resampling technique was chosen for model development.

As the tables show, SMOTE and Adasyn oversampler techniques performed the best overall. However, classification report (link to modeling notebook) showed that SMOTE performs slightly better compared to Adasyn therefore, SMOTE oversampler was chosen to resample the dataset for further model development.

After a resampling method is chosen, a new dataset that does not encounter imbalanced classification problems is created. The new dataset is used to conduct hyperparameter tuning.

For hyperparameter tuning, the four best classifiers that performed the best, including, Logistic Regression, RandomForest, LGBM and XGBoost classifiers, are tuned with SMOTE resampled dataset. Again, a pipeline that conducts hyperparameter tuning with different parameters for different classifiers and calculates scores is created.

Also, for this project, recall score was optimized for when hyperparameter tuning is conducted since the goal is to minimize the false negatives therefore the clients do not waste extra time and money.



	Logistic Regression		RandomForest	LGBM	XGBoost
Value: No Risk	Precision	0.89	0.92	0.97	0.97
	Recall	0.65	0.67	0.91	0.91
	F1-Score	0.75	0.78	0.94	0.94
Value: Risk	Precision	0.15	0.2	0.54	0.54
	Recall	0.44	0.6	0.78	0.78
	F1-Score	0.22	0.3	0.63	0.64

Figure 25: Table of results after Hyperparameter Tuning

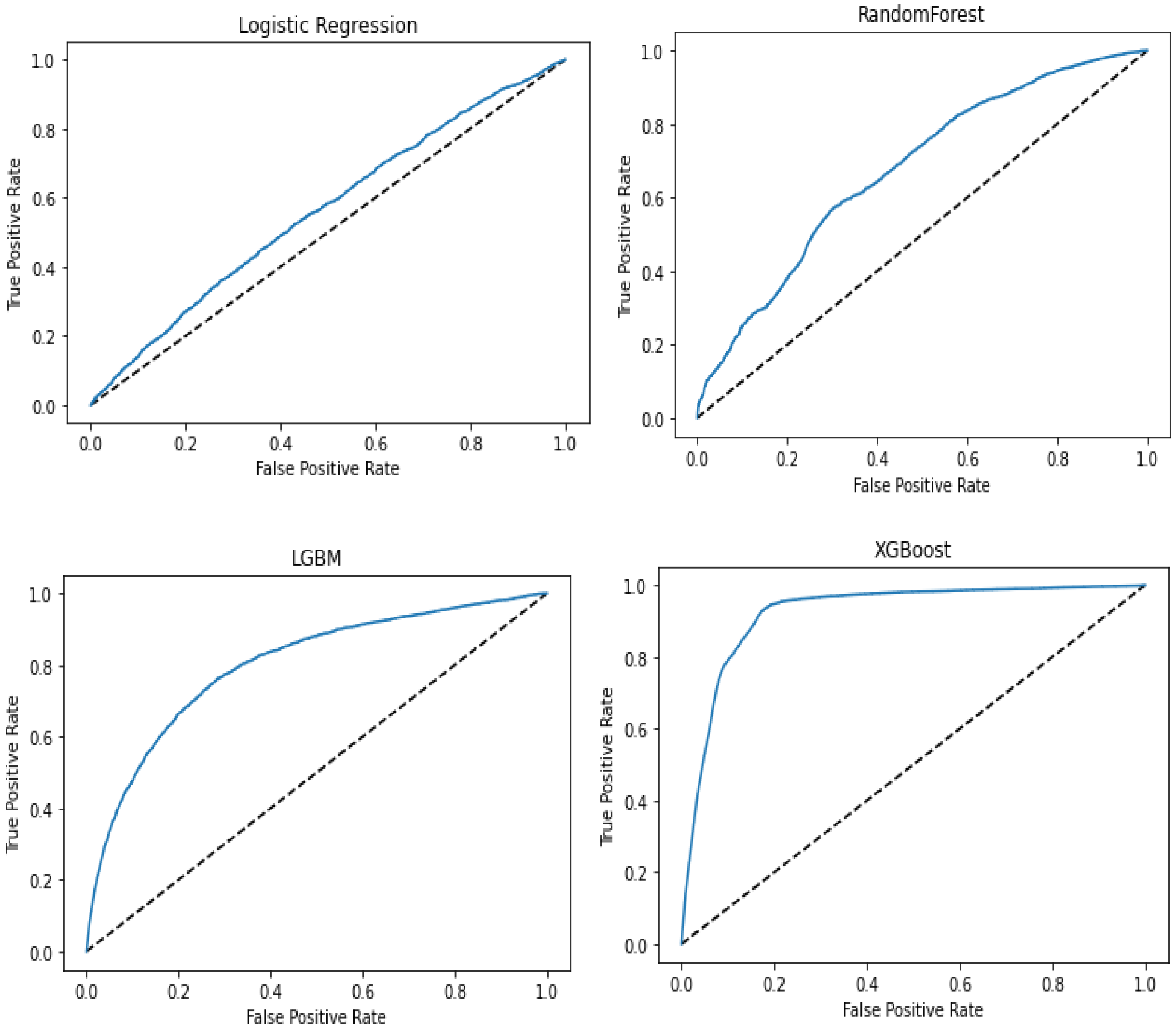


Figure 26: ROC-AUC Curve visualization of models

As the figure 25 shows, LGBM and XGBoost Classifiers both performed well on optimizing recall score for both classes. They almost have the exact same score, therefore ROC-AUC curves (figure 26) are plotted for more evaluation. As the curves show, the XGBoost classifier had a slightly higher ROC-AUC score with 84.24 percent whereas LGBM classifier had ROC-AUC score of 84.20 percent. However, the difference between the two classifiers was very small therefore confusion matrices of the two models are also visualized.

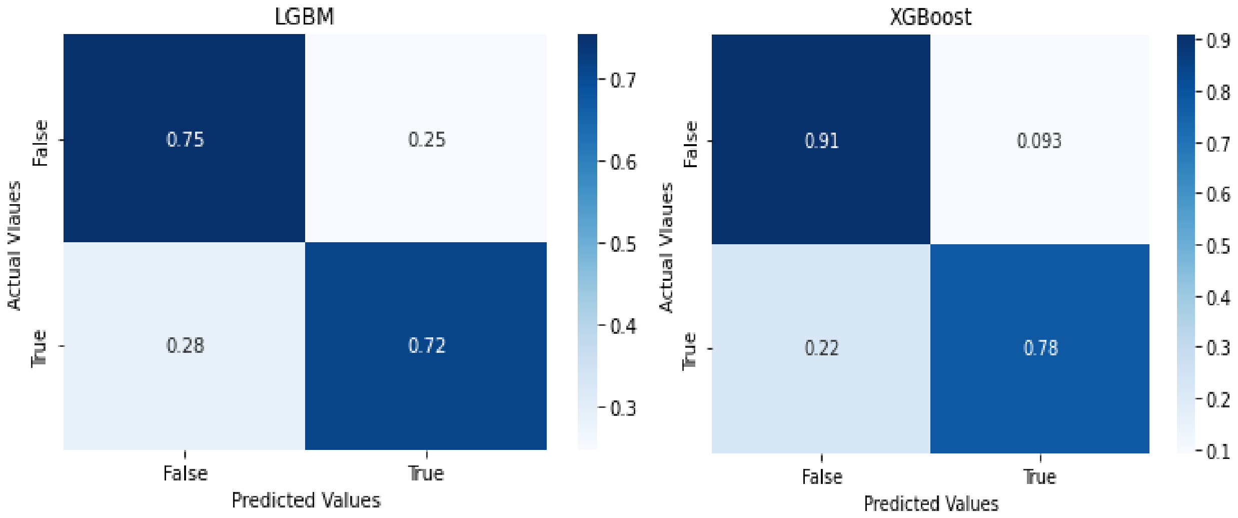


Figure 27: Confusion Matrices of LGBM and XGBoost Classifiers

As figure 27 shows, XGBoost classifier performed better compared to LGBM. Also as it was mentioned above, the recall score was optimized for the project to minimize the false negatives. For the XGBoost classifier, even if the recall score was optimized, the false positive rate was also lower than the LGBM classifier model. In conclusion, the XGBoost classifier model was chosen for our model with 84.24 percent of ROC-AUC score.

3. Findings

3.1) Feature Importance

For our findings, Logistic Regression coefficients, RandomForest feature importance tool and SHAP library was used to understand the feature importance of our models.

First, Logistic Regression coefficients are calculated to find out the feature impact and the numbers are visualized for better understanding.

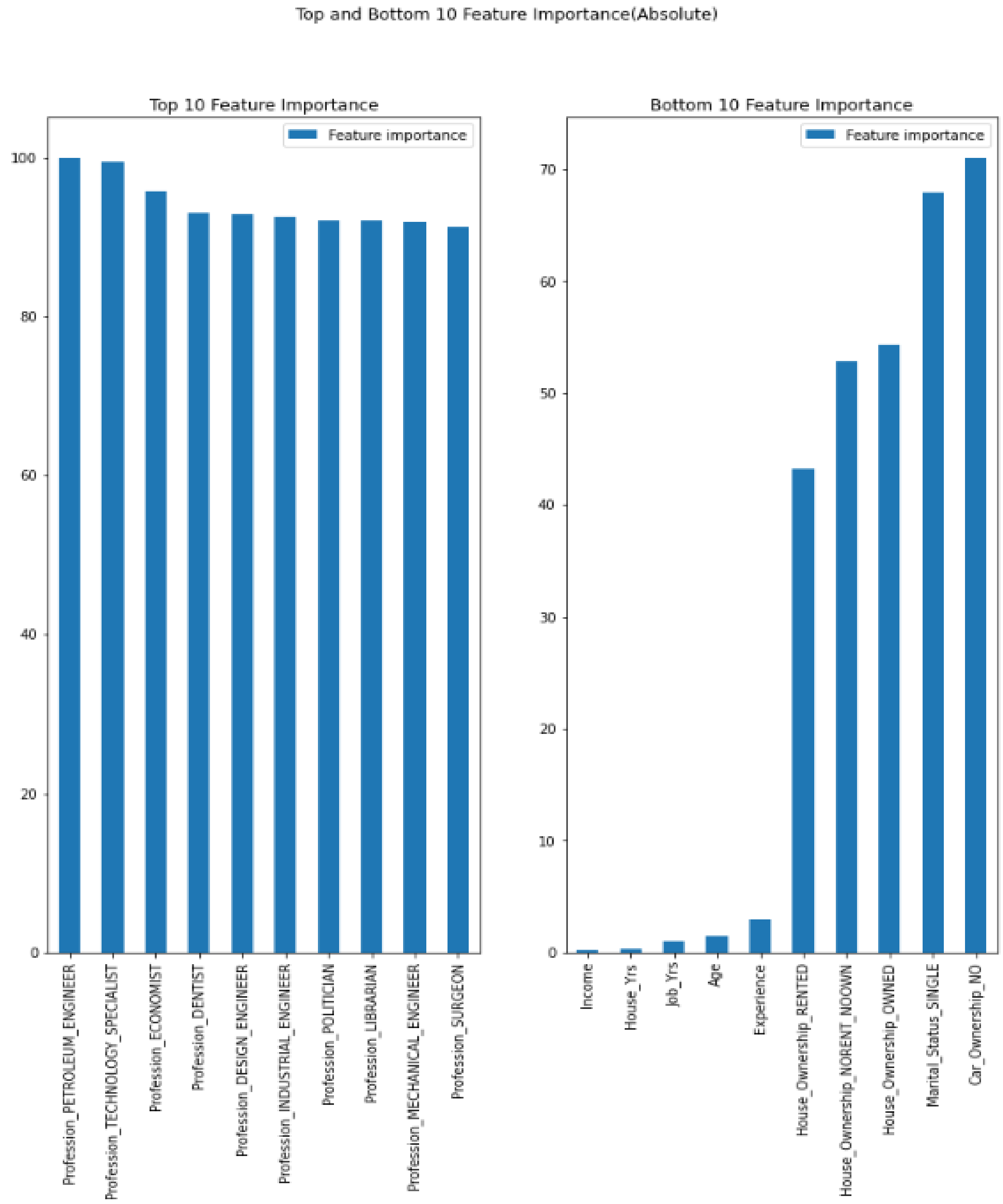


Figure 28: Bar graph of top and bottom 10 important features (Logistic Regression)

Figure 28 shows the top ten positive and negative features that lead to more and less risk in Logistic Regression. As the graphs show, the ten features that cause the higher risk are all professions and the ten features that reduce the risk include, Income, house years, job years, age, and experience. Therefore, for logistic regression, professions were the most important feature in predicting our target value. However, the Logistic Regression model did not perform well compared to other models. Therefore, other models' feature importance are analyzed for important features on well performing models.

Secondly, the Random Forest model's feature importance was analyzed.

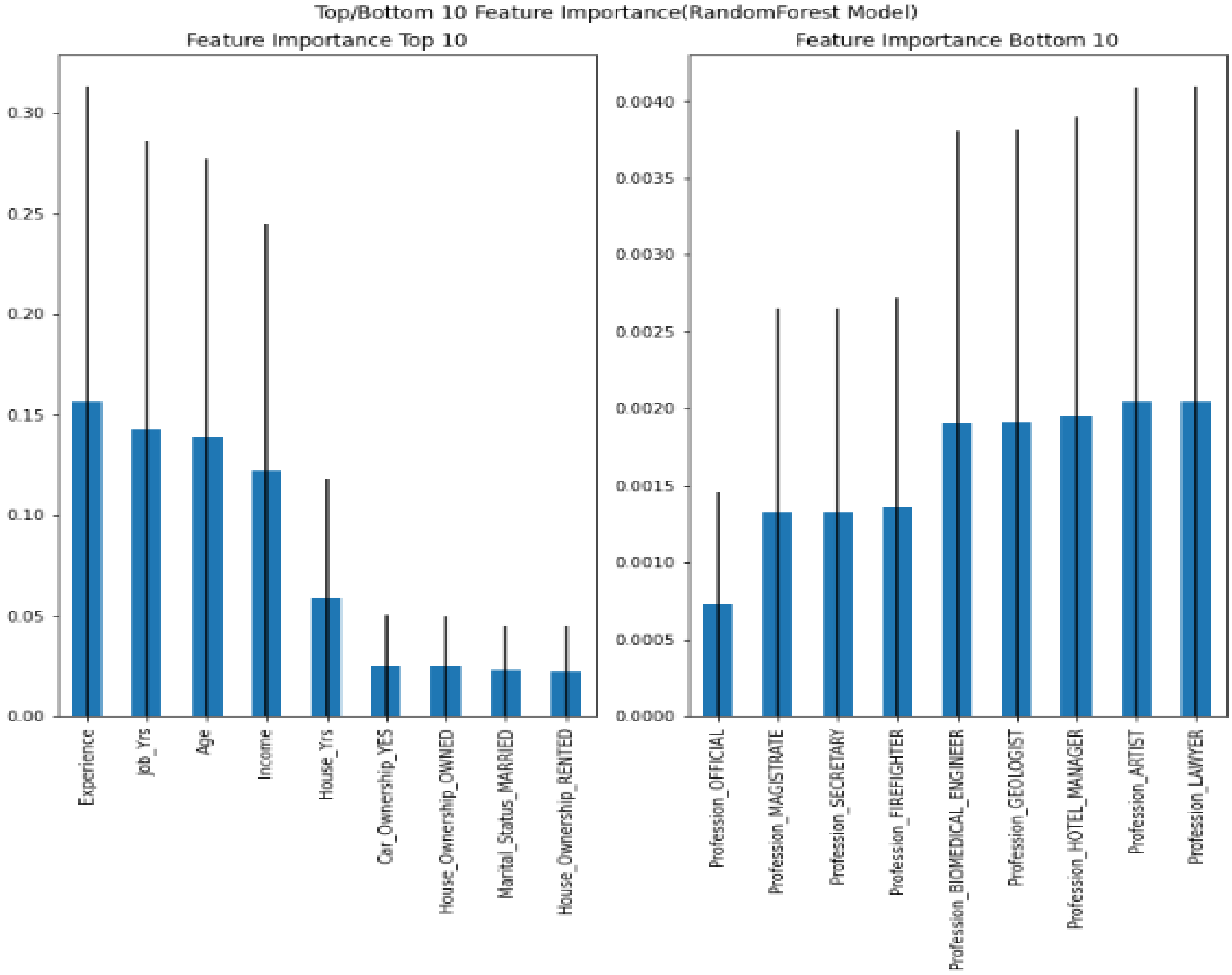


Figure 29: Bar graph, Top and bottom 10 important features(RandomForest Classifier)

The bar graphs are prepared for the RandomForest Classifier model and surprisingly, the result was the exact opposite from the Logistic Regression model. As the figure 29 shows, the ten most important features include, experience, job years, age, income and house years in order. Also, the ten least important features are all professions. Therefore, it means predicting our target value profession was not an important feature to consider for the Random Forest Classifier model.

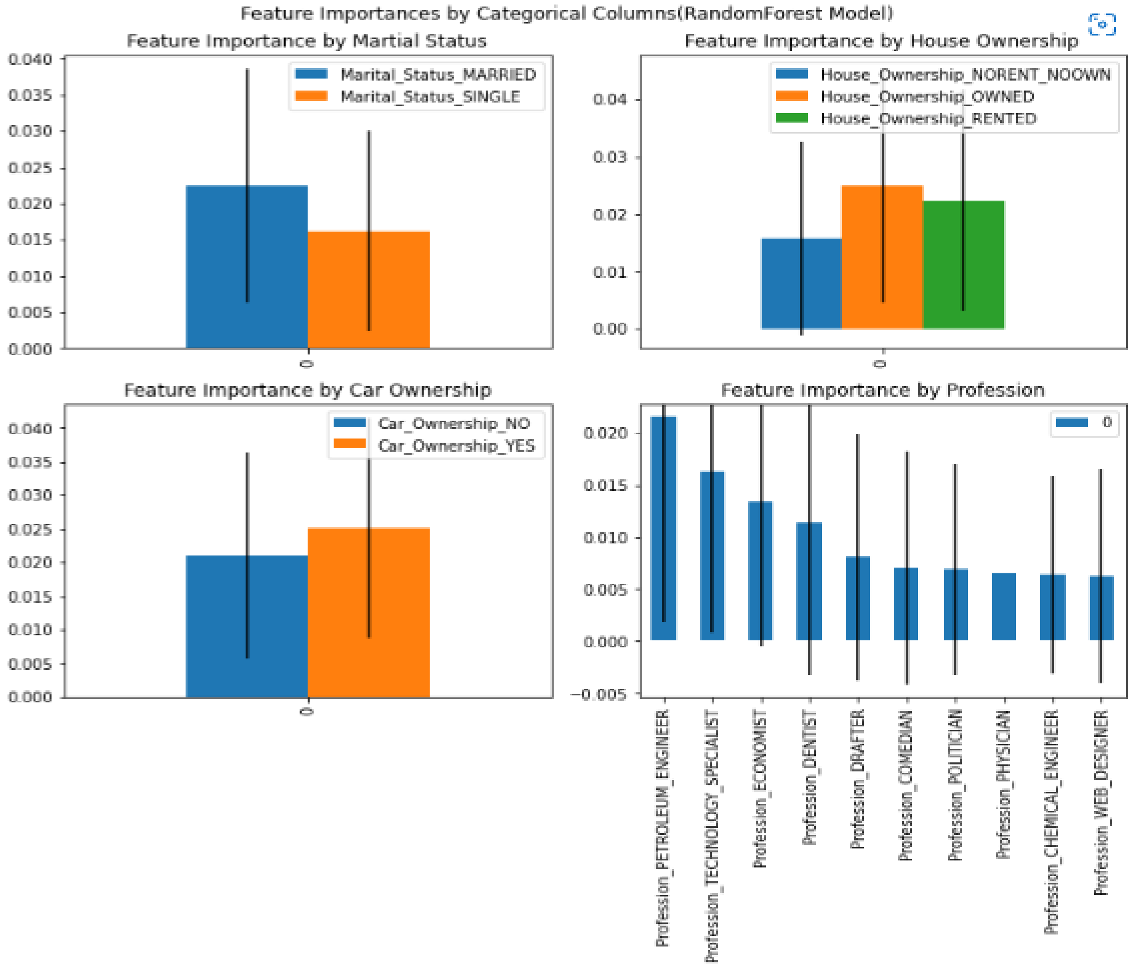


Figure 30: Feature importance by categorical columns (RandomForest Classifier)

Secondly, feature importance by categorical columns are plotted. It shows the importance of the values of categorical features. For marital status, clients who are married have a more important value compared to clients who are single. House ownership and car ownership, clients who own a house and car are considered more important values than the others. Lastly, professions including petroleum engineer, tech specialist, economist, dentist and drafter are the important values within a profession while predicting our target variable.

Lastly, SHAP library was used on our best performing model, XGBoost Classifier, to understand the feature importance.

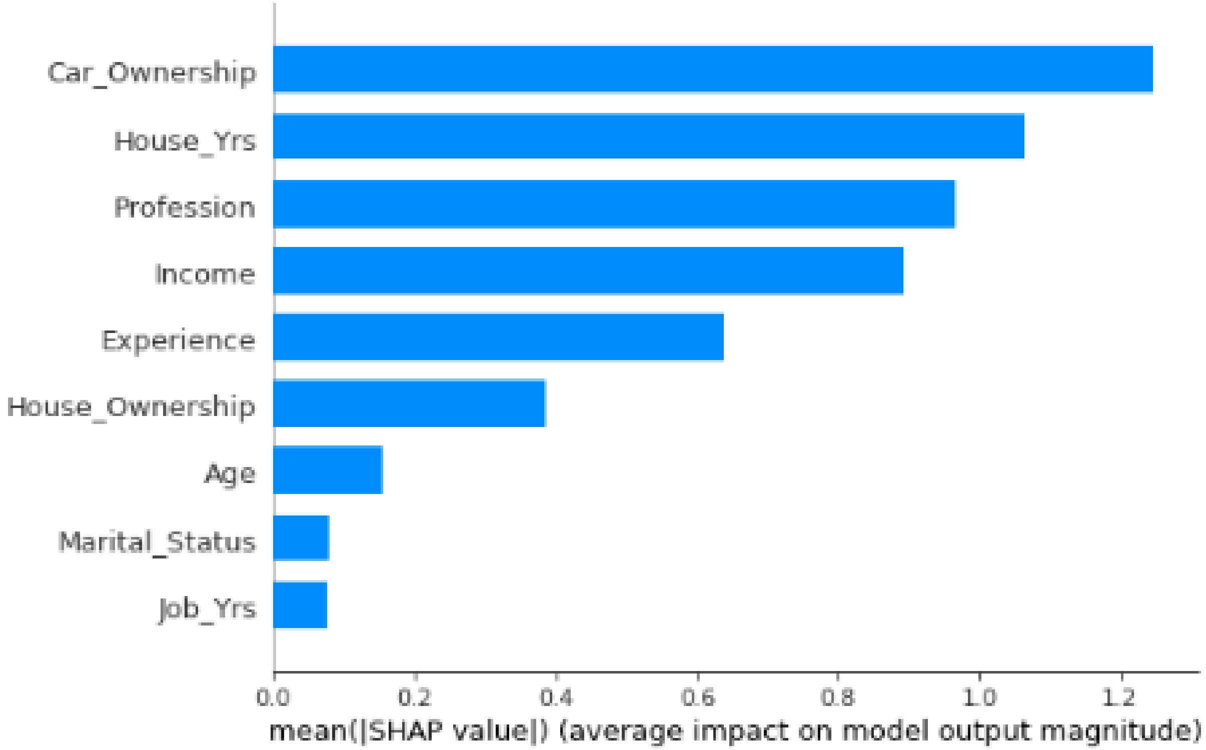


Figure 31: Bar plot, SHAP summary

Using a SHAP library, SHAP values of the features are calculated and the encoded columns' values are added to calculate the SHAP summary. As a result, we came up with a bar plot as FIgure 31 shows. As the graph shows, car ownership, house years, profession, income, experience, house ownership, age , marital status and job years was the order of feature importance in predicting the target variable in our final XGBoost model. Interestingly, car ownership and house years were not one of the most important features in logistic regression or random forest classifier.

Secondly, a beeswarm plot was created, with encoded columns included, to analyze the feature impact by the values as well. The professions are omitted to shrink the plot and the columns 'house ownership' was attached at the bottom. (The original beeswarm plot can be found on the modeling notebook).

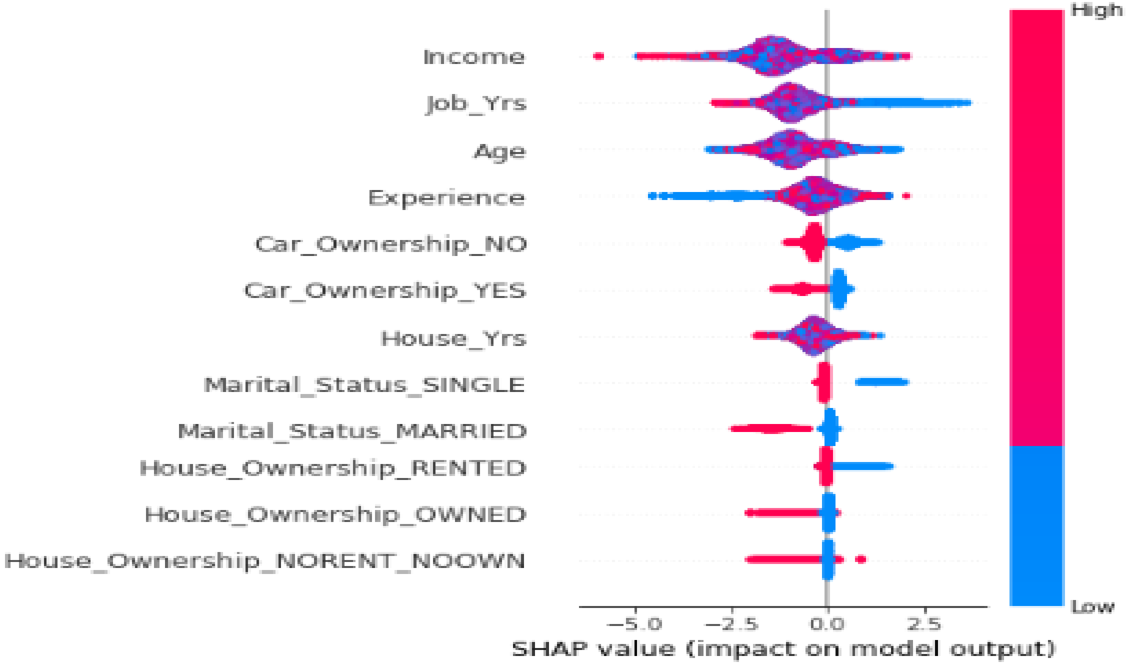


Figure 32: Beeswarm summary plot o fSHAP values

Based on the figure 32 feature impact of our model can be interpreted as follows.

More Risk:	**Less Risk:**
High income	Low Income
High number of Job years	Low number of Job years
Older age	Younger age
Lower years experience	Marital status single
Marital Status married	House rented
House owned or not owned/rented	

In conclusion, using the SHAP library we were able to understand the feature impact and how the values of the features are affecting the prediction of the target variable in our final model.

4. Future Work

For future work, some of the options may be implemented for potential improvement of the model performance and the options are as follows.

- Another dataset where it contains more financial status features of the clients can be used to create a new dataset where it has been joined to the dataset we have used in our project currently. The model still performed well however, two columns ‘city’ and ‘state’ were removed and adding more features may lead to improvement of the model performance.

- As we discussed in both Exploratory Data Analysis and Modeling notebooks, removing ‘City’ and ‘State’ columns do not affect the model performance and reduce the ‘noise’. However, another way to deal with the issue could have been developing separate models by state or city. Developing models by state would be more appropriate since it has an appropriate number of samples and a reasonable number of states. Therefore, developing models by state may be more useful for citizens in each state.

5. Recommendations for the Clients

As we consider recommendations for how to use these findings, it is important to understand that the final model has 84.4 percent of ROC-AUC score. Therefore the model can be a good reference however, it should not be a only resource to make a final decision for clients.

- First, the model can be used to provide high level and detailed counseling to loan applicants. As it was mentioned above, the detailed reasons for rejections are not fully disclosed to applicants. Therefore applicants who are seeking advice to improve their applications can be counseled on the potential reasons for denial and interventions that might lead to increasing or maximizing the probability of approval.

- Secondly, third parties and banks can use the model as a resource to check before the final application is applied for loan. Therefore, it gives an idea if the application will be approved or not. Therefore it can be used as a final check, and if the model can detect applicants who might not be eligible for loan, it will save money and time for applicants.

- Lastly, the model can be used for marketing purposes. Third party loan lenders invest huge amounts of money to advertise the company to potential clients who are seeking loans every year. However, most people prioritize banks over third party lenders and only a small percentage of people consider third party lenders as an option due to the low credit score of an individual. Therefore the model can be used by third party lenders to find their potential clients.