# LOAN APPROVAL PREDICTION

# TEAM INTRODUCTION

- Jyothi Sevakula
- Sanjna Kumari
- Muskaan Gupta
- Sri Venkateswara Swami Tumu

# Problem statement

- In this project we will predict weather a customer to be granted loan based on his behaviour. We chose this project because  It is very applicable in real life  and can be widely used by banks.
- Some of The behavioural features we included in our project are marital status, car ownership, experience etc.
- The data set we chose contains defaulters who did not pay loan and also people who paid and their behaviors.

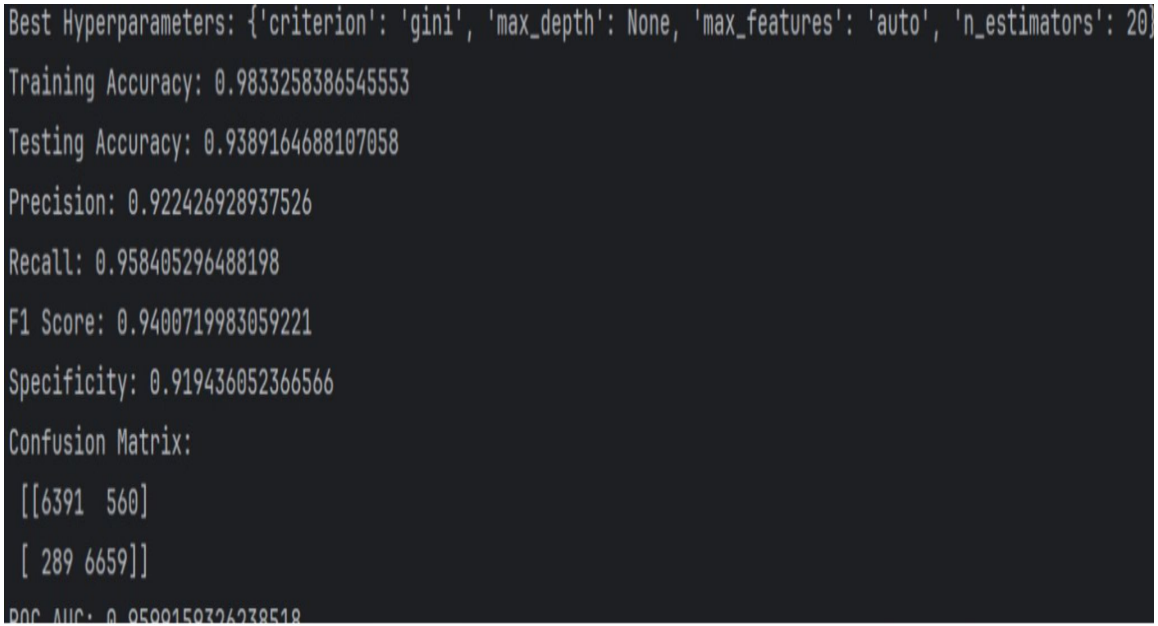
# Approaches

In this project we will predict loan approval using:

- Logistic Regression
- KNN
- Naive Bayes
- Support Vector Machine
- Random Forest

# Results

We obtained best result using RANDOM FOREST Classifier:



# Info of the Data set

```
RangeIndex: 252000 entries, 0 to 251999
Data columns (total 13 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   Id                  252000 non-null  int64
 1   Income              252000 non-null  int64
 2   Age                 252000 non-null  int64
 3   Experience          252000 non-null  int64
 4   Married/Single      252000 non-null  object
 5   House_Ownership     252000 non-null  object
 6   Car_Ownership       252000 non-null  object
 7   Profession          252000 non-null  object
 8   CITY                252000 non-null  object
 9   STATE               252000 non-null  object
 10  CURRENT_JOB_YRS     252000 non-null  int64
 11  CURRENT_HOUSE_YRS   252000 non-null  int64
 12  Risk_Flag           252000 non-null  int64
dtypes: int64(7), object(6)
memory usage: 25.0+ MB
```

# More info on data:

It has 11 features and a target variable ‘risk_flag’. If risk_flag is 1 then the person will not be eligible for the loan approval and if it is  0 he is eligible for the loan approval.

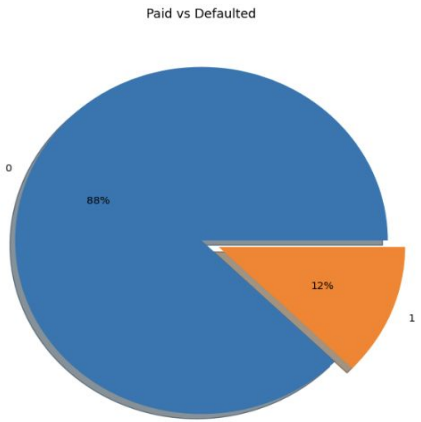
The initial 11 features in the dataset are [Income,Age,Experience,Married/Single,House_Ownership,Profession, CITY,STATE,CURRENT_JOB_YRS,CURRENT_HOUSE_YRS]

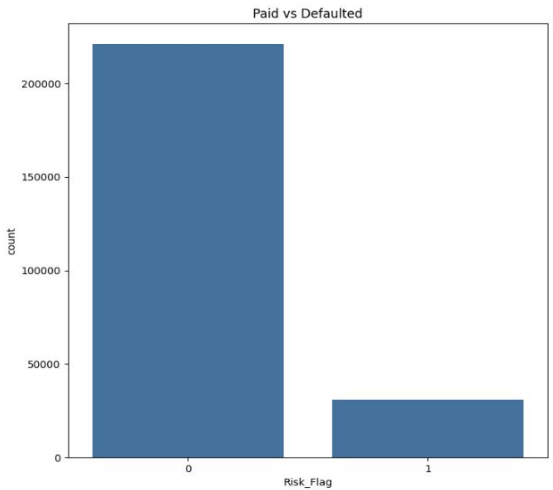
There are totally 5 numerical and 6 categorical features.

[Income,Age,Experience,CURRENT_JOB_YRS,CURRENT_HOUSE_YRS] are numerical and [Age,Experience,Married/Single,House_Ownership,Profession,CITY,STATE] are categorical features.

# How many have defaulted on loan(risk_flag=1)?

```
The rate of 'default-on-loan' is 12.3% and total number of defaulter are 30996
```



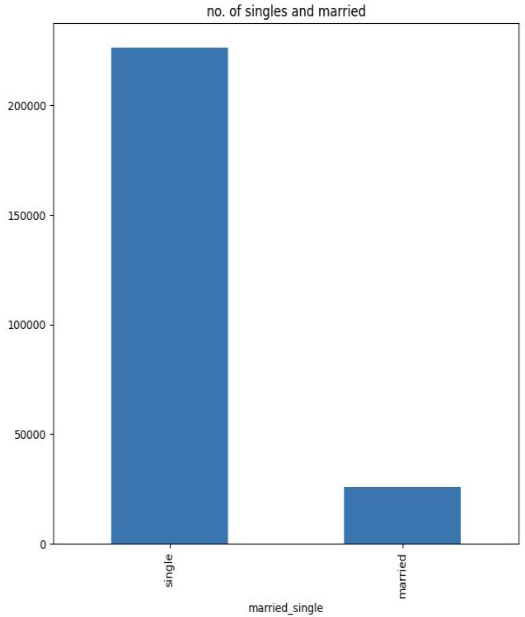

Here we can see that about 12% have defaulted-on-loan.

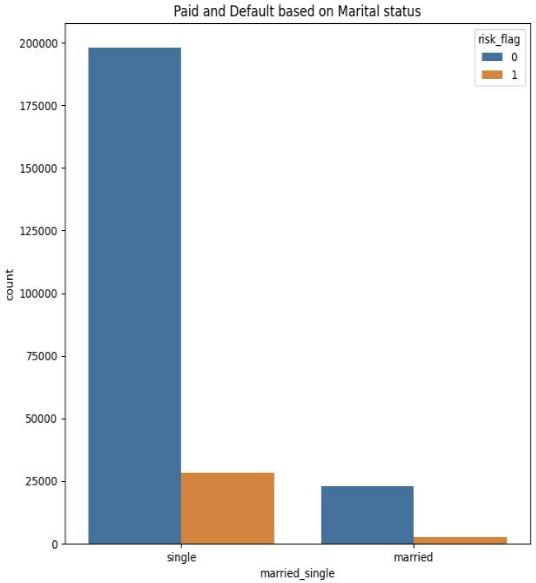
# Married _single VS Risk Flag:

defaulters for married is 10 %

defaulters for single is 13%

based on calculation and figures we can say that singles have more Risk Flags than married when compared and there are more singles that wants loan than married.

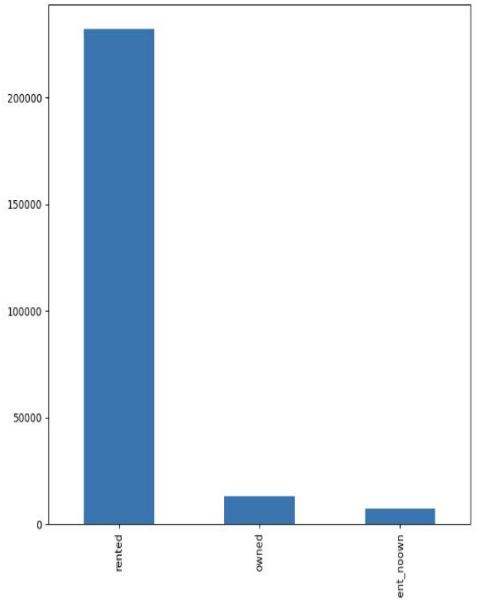


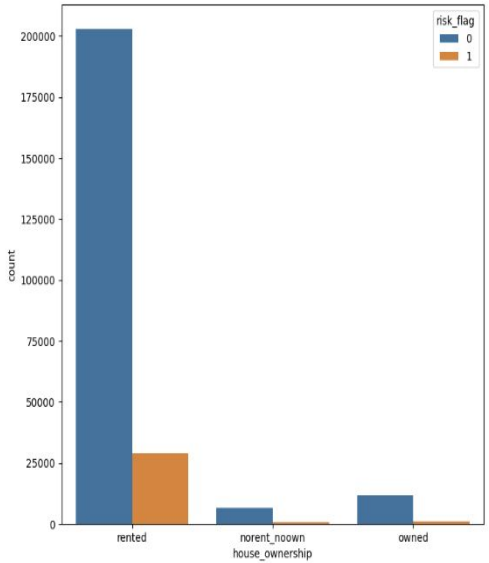



# Explore House Ownership vs Risk Flag

| house_ownership | |
|---|---|
| rented | 231898 |
| owned | 12918 |
| norent_noown | 7184 |

People who Rented are the highest loan takers and highest defaulters



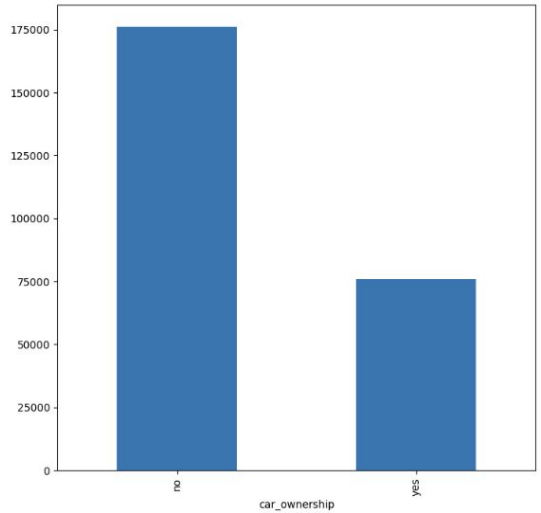
# Car Ownership vs Risk Flag

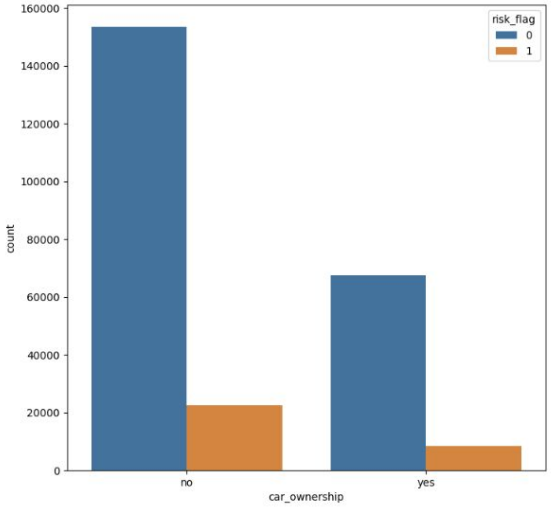
Name: count, dtype: int64

car_ownership

no 176000

yes 76000



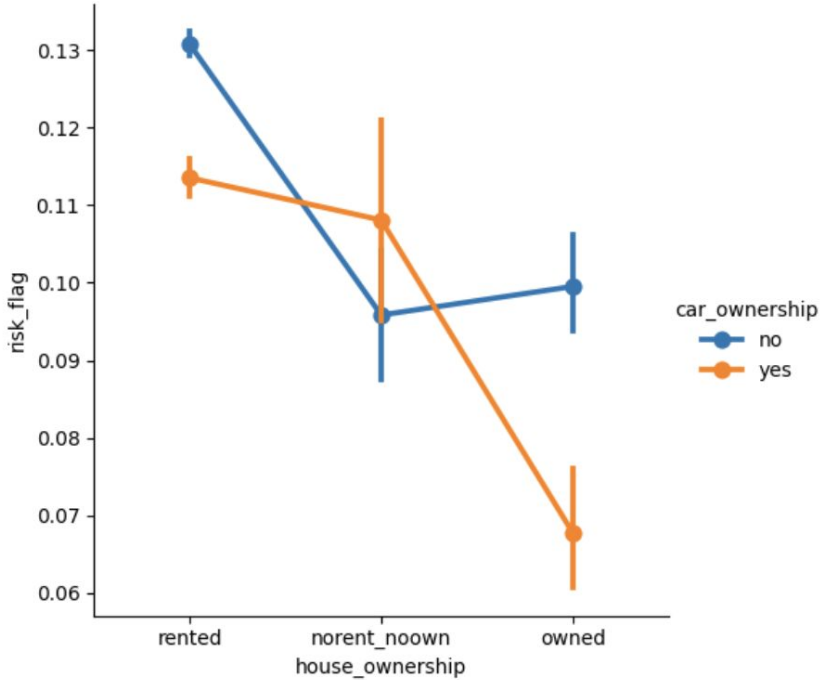
| risk_flag | 0 | 1 | All |
|---|---|---|---|
| car_ownership | | | |
| All | 221004 | 30996 | 252000 |
| no | 153439 | 22561 | 176000 |
| yes | 67565 | 8435 | 76000 |

defaulters with car: 11 %

defaulters with no car: 13 %

If we observe percentage of defaulters with no car are more than people with car .

# Catplot of house _owenership, car _ownership and Risk _flag




People with no car(13%) tend to default-on-loan more than people with car(11%). From catplot we can see that people with house ownership as well as car ownership default-on-loan less than other category.

# Exploring State and city features

There are 29 states and 317 cities

```
(29,)
(317,)
```

so, basically cities are subset of states, and there are almost 317 cities we only explore State for visual representation

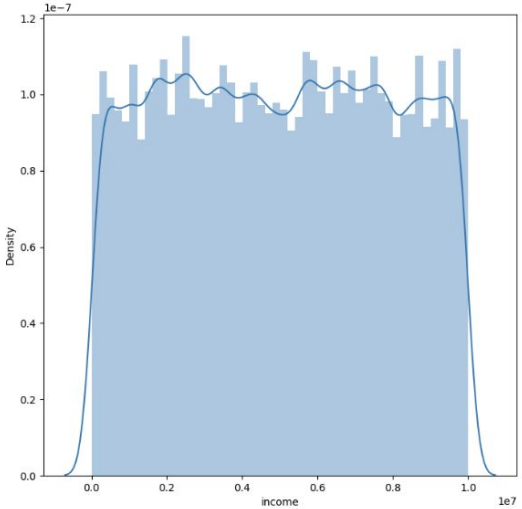
| state | |
|---|---|
| Manipur | 0.216 |
| Tripura | 0.168 |
| Kerala | 0.167 |
| Jammu_and_Kashmir | 0.159 |
| Madhya_Pradesh | 0.154 |

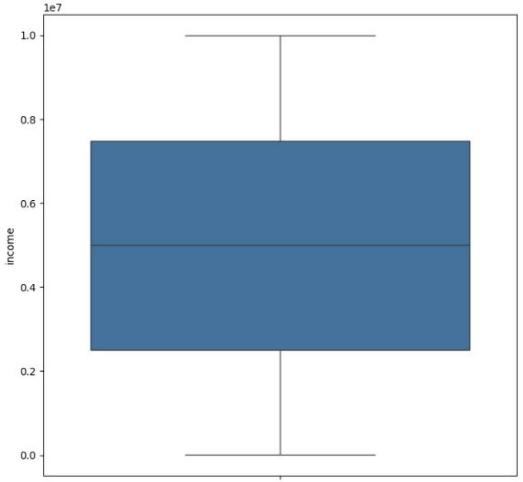
| city | |
|---|---|
| Bhubaneswar | 0.326 |
| Gwalior | 0.273 |
| Bettiah[33] | 0.267 |
| Kochi | 0.253 |
| Raiganj | 0.240 |

By observing in states manipur has the highest defaulters percentage of 21.6 and in cities bhubaneswar has highest defaulters percentage of 32.6.

# Exploring Income

```
Highest Income is: 9999938
Lowest Income is: 10310
Average Income is: 4997116.66
```




Income is Normal

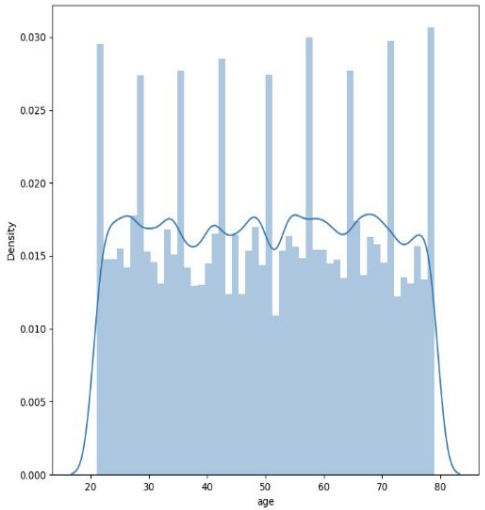
As there is no

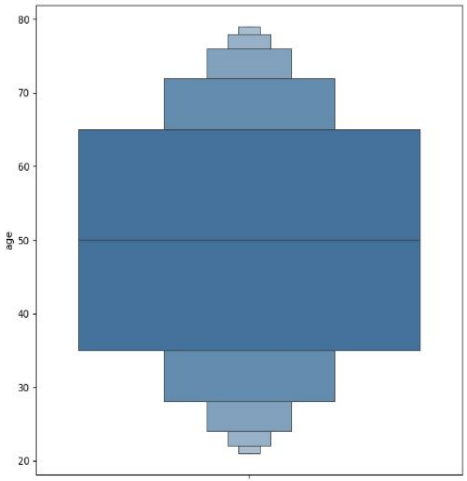
Outlier in boxplot

# Exploring Age

```
Highest age is: 79
Lowest age is: 21
Average age is: 49.95
```

Highest age is 79 years old while youngest is 21 and average is 49 years old.And from box plot we can see that from 35 to 65 the density is more i.e. 35-65 takes most loan.

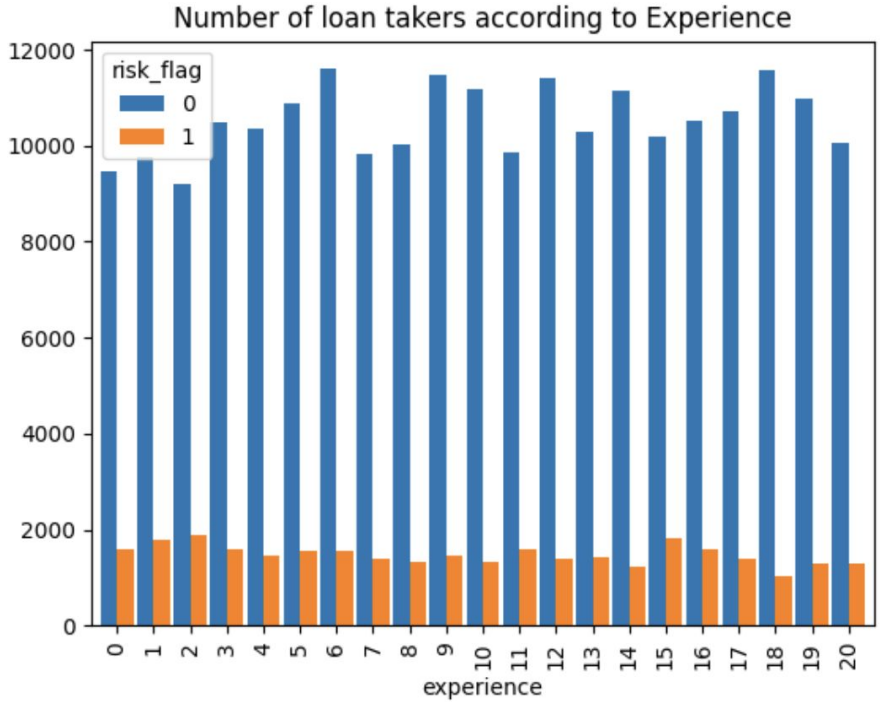


# Exploring Experience

| experience | 0 | 1 | 2 | 3 | ... | 18 | 19 | 20 | All |
|---|---|---|---|---|---|---|---|---|---|
| risk_flag | | | | | ... | | | | |
| 0 | 9461 | 9773 | 9197 | 10483 | ... | 11572 | 10982 | 10066 | 221004 |
| 1 | 1582 | 1802 | 1890 | 1586 | ... | 1029 | 1305 | 1284 | 30996 |
| All | 11043 | 11575 | 11087 | 12069 | ... | 12601 | 12287 | 11350 | 252000 |

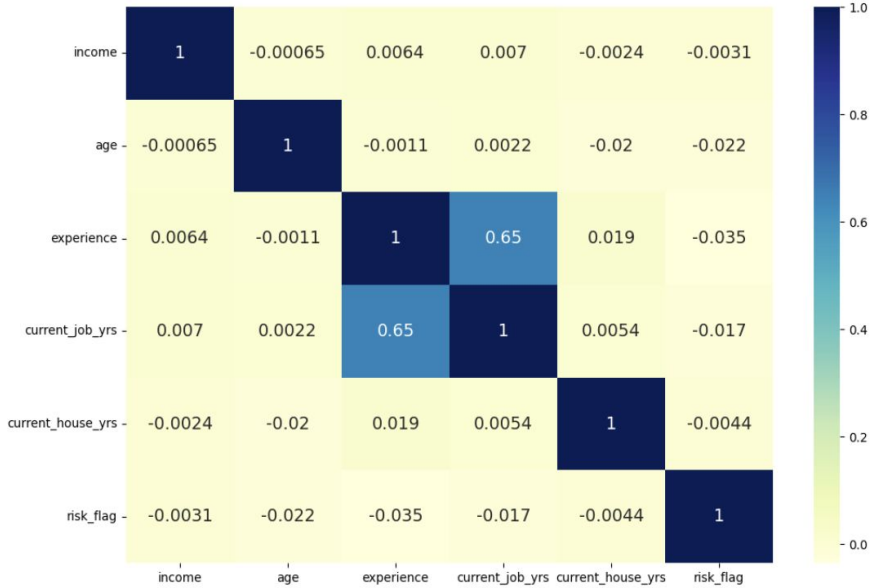
[3 rows x 22 columns]



# Correlation Matrix :

- The correlation matrix shows the strength and direction of linear relationships between pairs of variables. A correlation coefficient close to +1 indicates a strong positive linear relationship, while a coefficient close to -1 indicates a strong negative linear relationship.

- We can observe that Correlation coefficient of current_job_yrs and Experience is high so We can drop any one,I chose experience to drop.

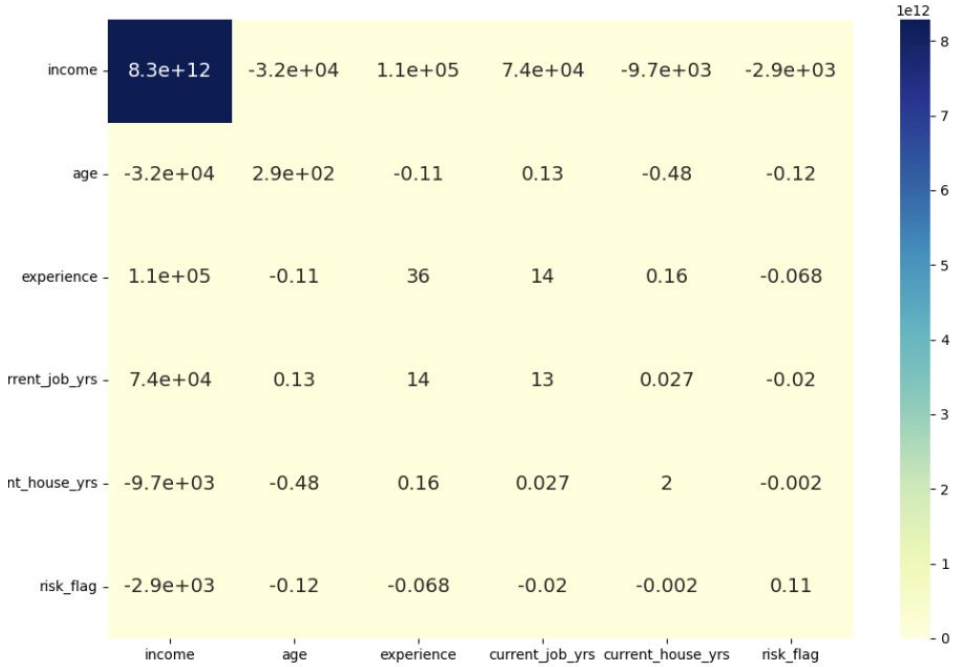


# Covariance matrix:

Covariance indicates whether two variables tend to increase or decrease together. A positive covariance suggests a positive relationship, while a negative covariance suggests a negative relationship

The diagonal elements of the covariance matrix represent the variances of individual variables. The variance is a measure of how much a variable deviates from its mean. Larger variances indicate greater variability in the data for a specific variable.

If we observe ‘experience’ and ‘current job years ‘have larger covariance of 14 so they increase or decrease together so we can use any one feature, because Variability in the data can be drawn from one feature.



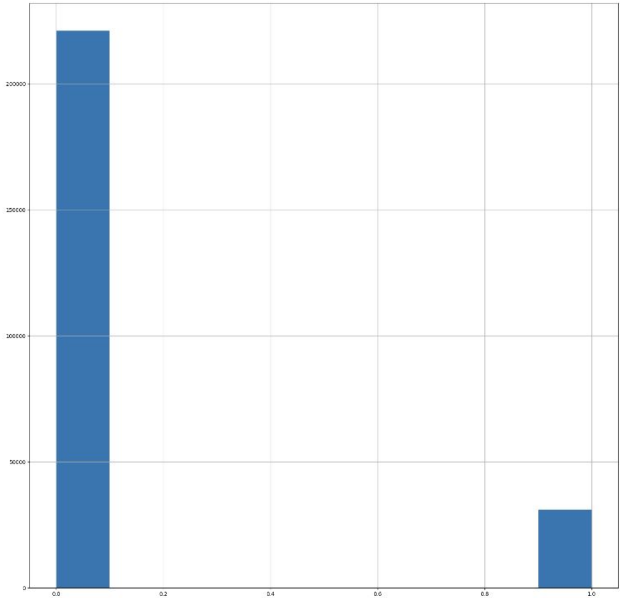
# Balancing the Data

- If one class significantly outnumbers the others, the algorithm may be biased toward the majority class, leading to poor performance on the minority class.

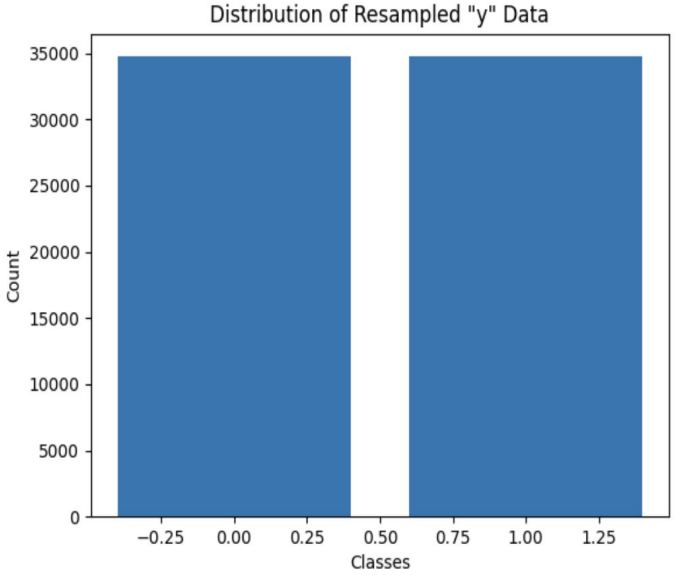
- If the dataset is imbalanced, a model may achieve high accuracy by simply predicting the majority class most of the time. However, such a model may not generalize well to new data. Balancing the data helps prevent the model from overfitting to the majority class

- Balancing the data is best practice for any machine learning analysis , mainly during classification.

# Balancing the Data



Data is not balanced



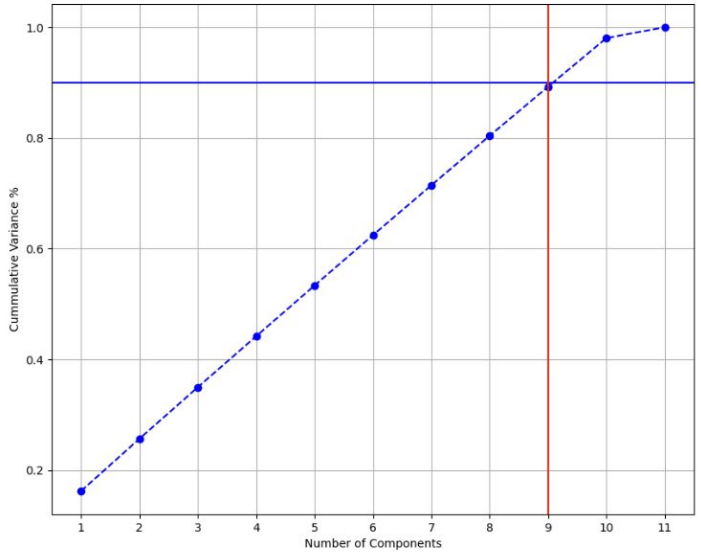

Removed duplicates and done Oversampling

# Performed PCA:

```
dtype='object')
Number of components needed to explain 85% of the variance: 9
```

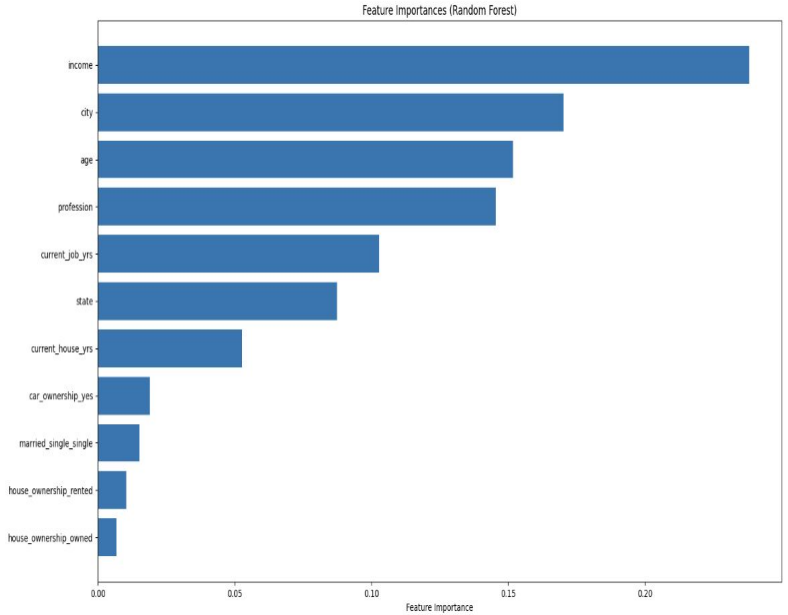
This is dimensionality reduction method.

If we observe the pca analysis, the minimum number of features needed to explain the 85% of variance are 9 out of 11 features.



# Randomforest for Feature Importance

- If we observe the feature importance graph there are four features which does not have a minimum importance of 0.05. I have taken the threshold as 0.05 and have dropped the features car_owned_yes,married_single_single,house_ownership_rented,house_ownership_owned



# Regression analysis:

## T-test:

t-test is often used to assess the significance of individual coefficients in a linear regression model. The null hypothesis (H0) associated with a t-test for a specific coefficient typically states that there is no significant effect of that particular predictor variable on the dependent variable.

After performing the t-test and obtaining a p-value, you would compare the p-value to a chosen significance level (e.g., 0.05) to make a decision about whether to reject the null hypothesis. If the p-value is less than or equal to the significance level, you may reject the null hypothesis, suggesting that there is evidence of a significant relationship between the predictor variable and the dependent variable. If the p-value is greater than the significance level, you do not reject the null hypothesis, indicating that there is not enough evidence to conclude a significant effect.

In my final regression model there are features which have a p-value of 0.05 so we reject the null hypothesis.

# F-Test

In the context of regression analysis, the F-test is used to assess the overall significance of a linear regression model. The F-test compares the fit of the full model (i.e., the model with predictors) to a reduced model (i.e., a model without predictors) to determine whether the inclusion of predictors significantly improves the model's ability to explain the variability in the dependent variable

$$F = \frac{\frac{\text{Explained Variance in Full Model}}{\text{Number of Parameters in Full Model}}}{\frac{\text{Unexplained Variance in Full Model}}{\text{Degrees of Freedom (Error)}}}$$

In summary, the F-test helps determine whether the inclusion of predictor variables in a regression model is justified and whether the overall model explains a significant amount of variance in the dependent variable.

# Backward Regression Analysis:

We perform  Backward Regression analysis. If we eliminate features based the p values. We eliminate the features with the highway value every time we run a regression analysis. The threshold of p-test is 0.05 , any p-value greater than 0.05 will be deleted .

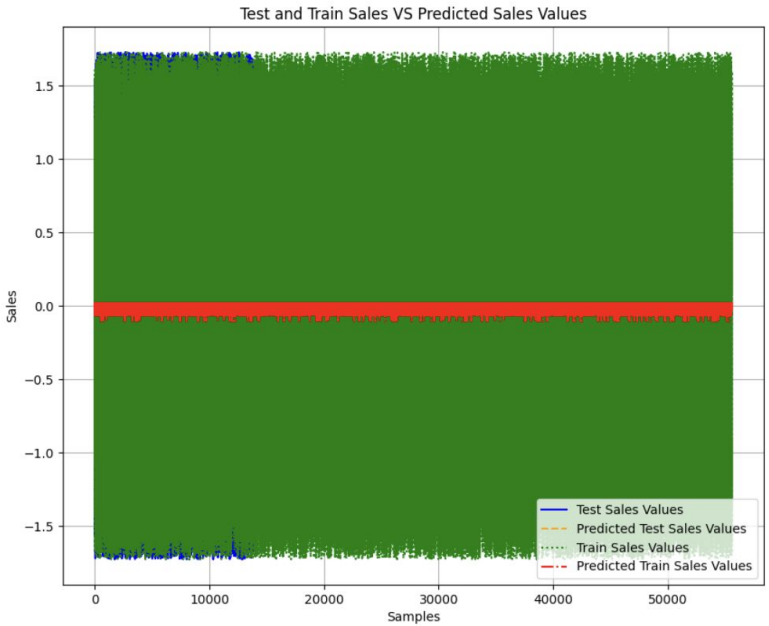
** While performing backward regression we should monitor Adj R2 value too the best model will have high Adj R2 value. We should perform any analysis unless there is sudden decrease of adj R2.

Mean Squared Error: 1.0033678442927012

| process Update | AIC | BIC | Adj R^2 | p-value |
|---|---|---|---|---|
| const | 157706.973 | 157814.083 | 0.001 | 0.806 |
| house_ownership_owned | 157705.033 | 157803.217 | 0.001 | 0.209 |
| city | 157704.613 | 157793.871 | 0.001 | 0.479 |
| risk_flag | 157703.114 | 157783.446 | 0.001 | 0.465 |
| current_job_yrs | 157701.648 | 157773.055 | 0.001 | 0.117 |
| current_house_yrs | 157702.105 | 157764.586 | 0.001 | 0.899 |
| profession | 157700.121 | 157753.676 | 0.001 | 0.275 |
| age | 157699.313 | 157743.942 | 0.001 | 0.17 |
| state | 157699.2 | 157734.903 | 0.001 | 0.192 |

In the graph We have depicted the prediction values and original value of both test and train data.As guided We have plotted both test and train predicted values in the same plot.

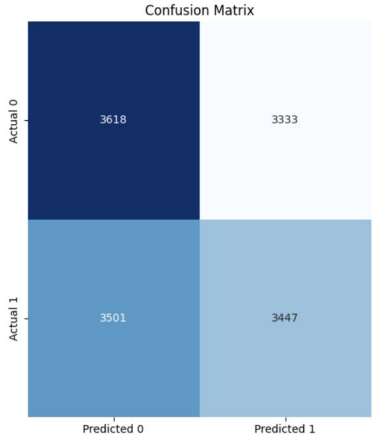
Our original dataset is classification problem. We have chosen to perform the regression on income as We felt that is the most relevant out of all features.

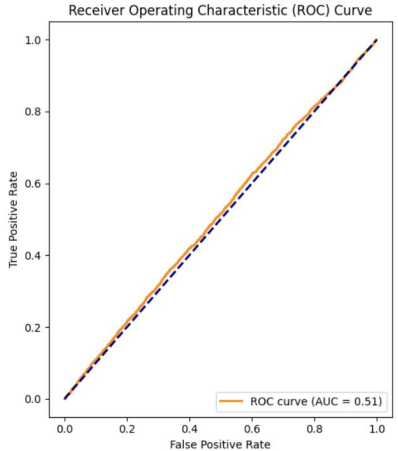


# Logistic Regression

Logistic Regression serves as a statistical technique employed for binary classification tasks, where it predicts the probability of an event occurring for a categorical dependent variable, typically represented as 0 or 1. A grid search was conducted on parameters 'C', 'Solver', and 'Penalty', alongside stratified k-fold cross-validation, resulting in the identification of the best hyperparameters.

```
Best Hyperparameters: {'C': 10, 'penalty': 'l1', 'solver': 'saga'}
Training Accuracy: 0.5111610756362982
Testing Accuracy: 0.5083099503561407
Precision: 0.5084070796460177
Recall: 0.4961139896373057
F1 Score: 0.5021853146853146
Specificity: 0.5205006473888649
Confusion Matrix:
 [[3618 3333]
 [3501 3447]]
ROC AUC: 0.5121999029807054
```

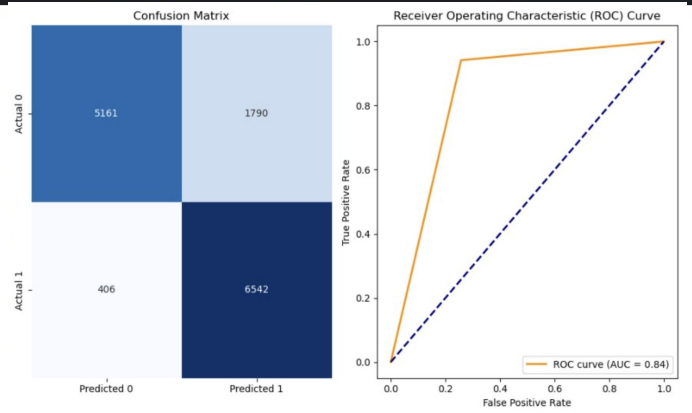


# KNN

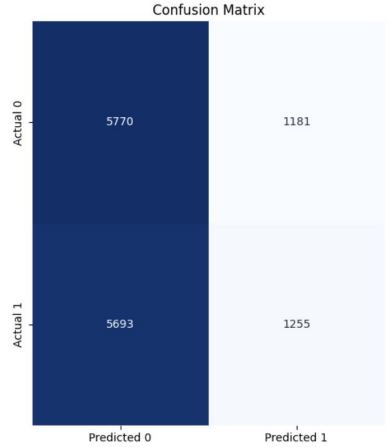
K-Nearest Neighbors (KNN) is a simple yet effective supervised learning algorithm used for both classification and regression tasks. The 'K' in KNN refers to the number of nearest neighbors used for prediction. When a new data point needs to be classified or predicted, the algorithm identifies its K nearest neighbors based on a chosen distance metric

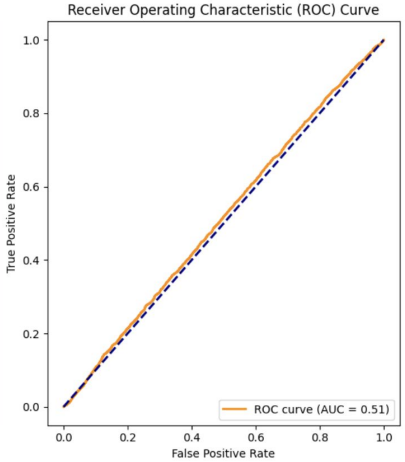
```
Optimal k: 1
Training Accuracy: 0.9755733429265222
Testing Accuracy: 0.8420030218001295
Precision: 0.78516562650024
Recall: 0.941565918249856
F1 Score: 0.8562827225130889
Specificity: 0.7424830959574162
Confusion Matrix:
 [[5161 1790]
 [ 406 6542]]
ROC AUC: 0.8420245071036361
```



# Naive bayes

Naive Bayes is a popular and simple probabilistic machine learning algorithm used primarily for classification tasks. It's based on Bayes' theorem with an assumption of independence among predictors (features), known as the "naive" assumption. Grid search is not possible as there are no parameters. Stratified K fold approach followed in calculating the test accuracy.

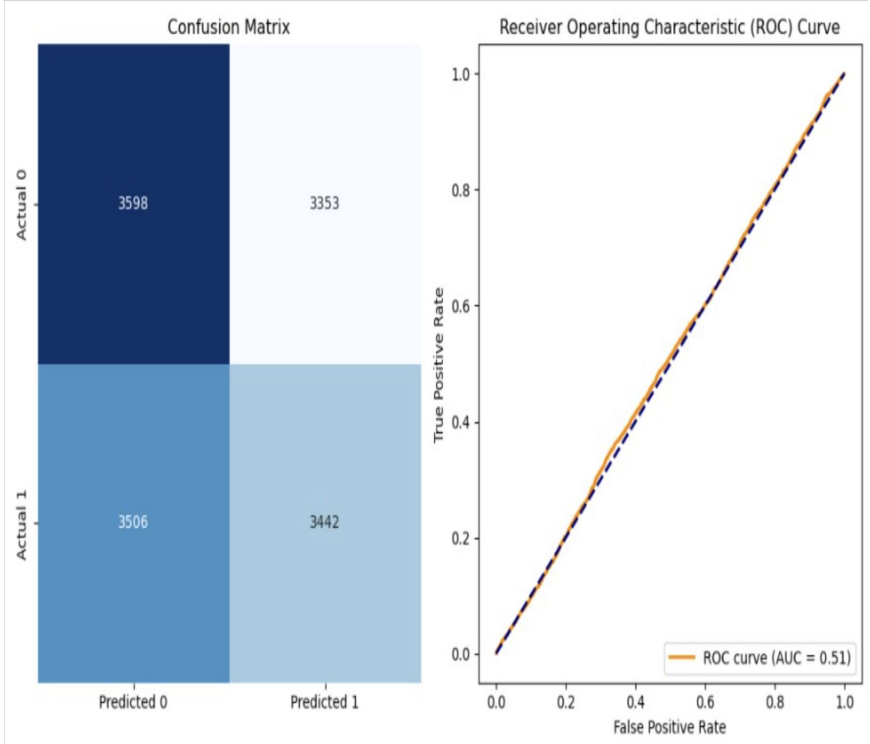



```
Training Accuracy: 0.5043619030488353
Testing Accuracy: 0.5065112598028635
Precision: 0.5065489330389993
Recall: 0.4953943580886586
F1 Score: 0.5009095539547406
Specificity: 0.5176233635448136
Confusion Matrix:
 [[3598 3353]
 [3506 3442]]
ROC AUC: 0.5064803074602238
```

# SVM

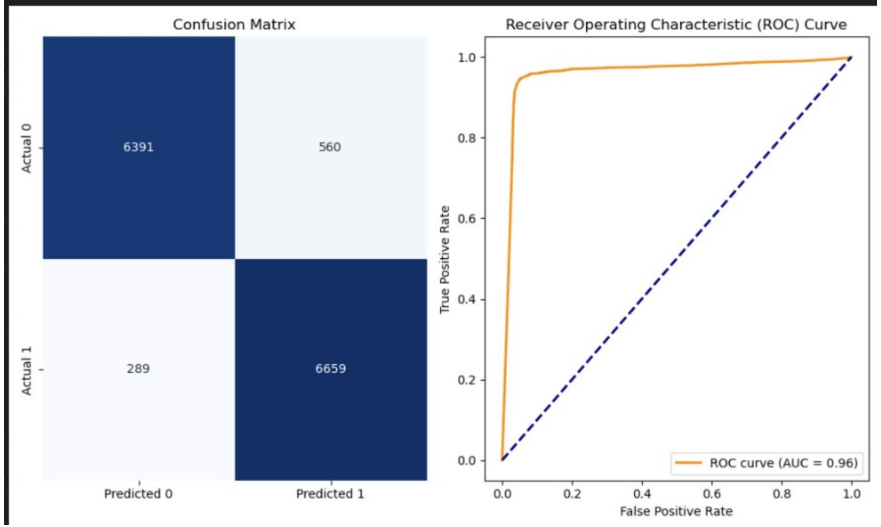
SVM finds the hyperplane that maximizes the margin between the closest data points (support vectors) of different classes. Performed grid search with linear, poly and rbf kernel and stratified k fold cross validation on searching for the best kernel. Best kernel is found to be the radial basis function which is widely used for nonlinear problems




```
Best Hyperparameters: {'kernel': 'rbf'}
```

# Random Forest

Random Forest, a robust ensemble learning technique applicable to both classification and regression tasks, builds multiple decision trees during training. The model produces its final prediction by considering the mode of classes for classification or the mean prediction for regression from the individual trees. Best hyperparameters are determined through grid search and stratified k-fold cross-validation. The model exhibits commendable accuracy, with decent precision and good specificity, albeit showing lower recall and f-score.




```
Best Hyperparameters: {'criterion': 'gini', 'max_depth': None, 'max_features': 'auto', 'n_estimators': 20
Training Accuracy: 0.983325838545553
Testing Accuracy: 0.9389164688107058
Precision: 0.922426928937526
Recall: 0.95840529648819 8
F1 Score: 0.9400719983059221
Specificity: 0.919436052366566
Confusion Matrix:
 [[6391  560]
 [ 289 6659]]
ROC AUC: 0.9509150326238518
```

# Best Classfiers

If we observe all the parameters which decide the the efficiency of a model, Randomforest is has all the  best outcomes of around 96% accuracy. so Randomforest classifier is the best classifier.