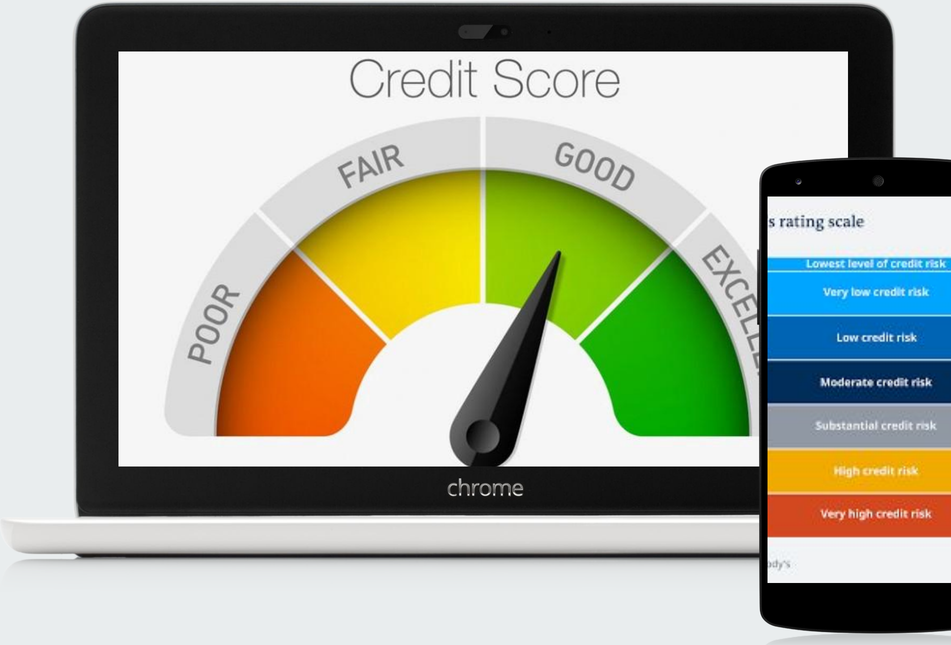
FE-520 Project Presentation

Predicting credit rating using machine learning

Team Members:
Parth Parab (10444835)
Varsha Raghavendra (10444838)



Outline

The Problem

Our Data

Classification Algorithms

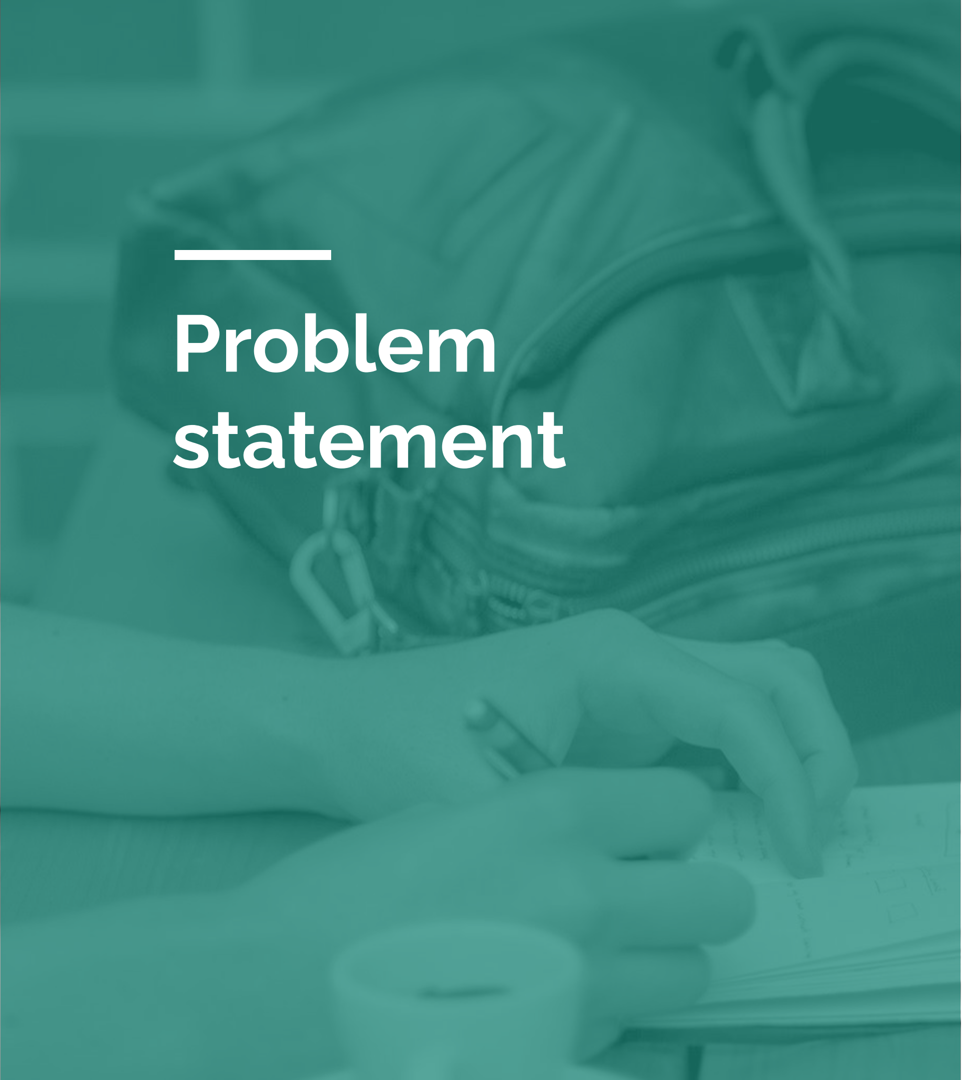
Next Steps

Conclusion

The Problem

Topic: Predicting credit rating using machine learning
The topic we chose specifically is to predict whether a credit card will be approved or not for that customer based on the entire credit rating and customer information given to us by an anonymous bank.

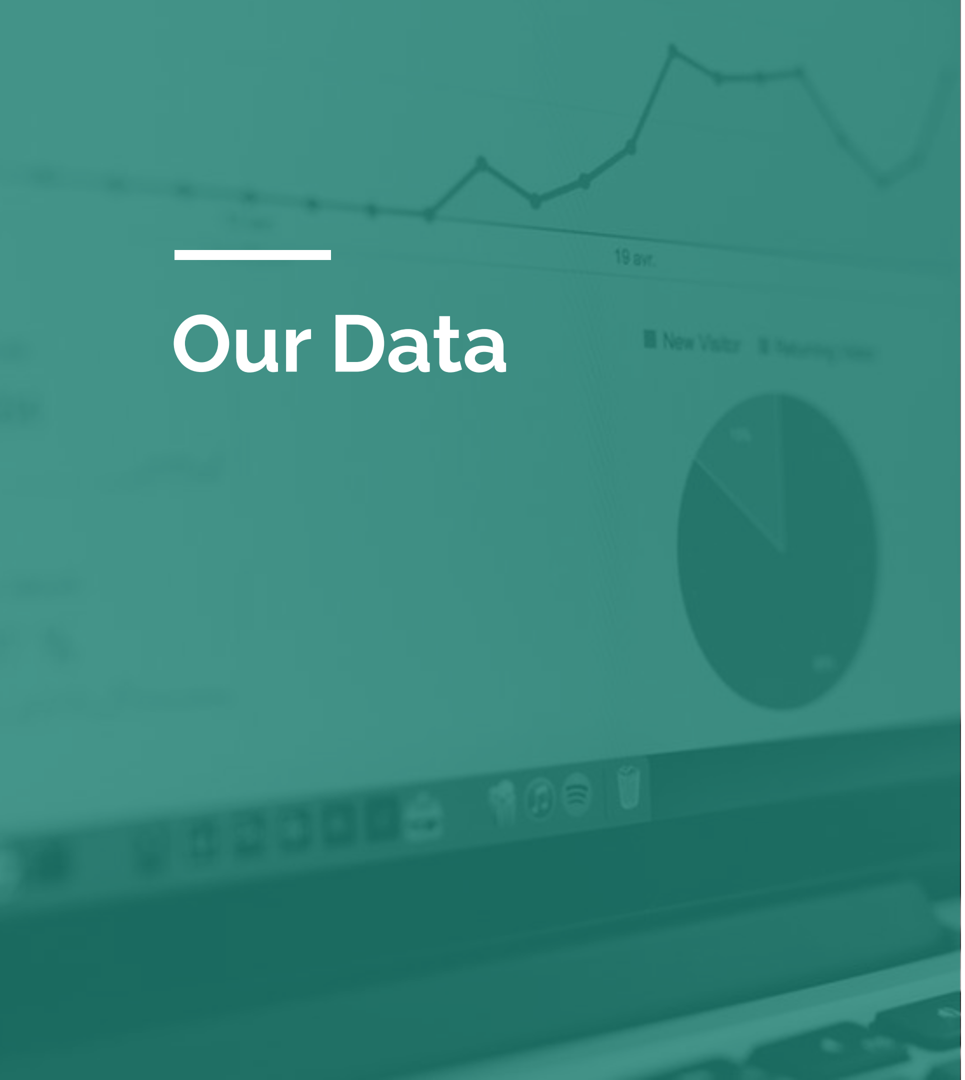
Analysis of customer history data often gives us a lot of insight into whether approving credit cards would prove to be disastrous to the bank or not. Repayment of credit and loans are how banks function and are necessary to be ensured before giving away credit cards to customers. Other than checking for income or bank balance a lot of other factors play a role. Analyzing the data we have, making them useful and building predictive models will make credit card approvals a more automatic and faster process also ensuring integrity of the customer



Problem statement

Given applicant data history, predict whether a credit card should be approved or not

This task is achieved using Supervised Machine Learning Classification techniques where a model is trained on data that has certain features in order to accurately predict the label for the given target variable which in our case would be a binary class of 0s and 1s (whether credit card was approved or not)



Our Data

Our data is extracted from Kaggle

https://www.kaggle.com/rikdifos/credit-card-approval-prediction

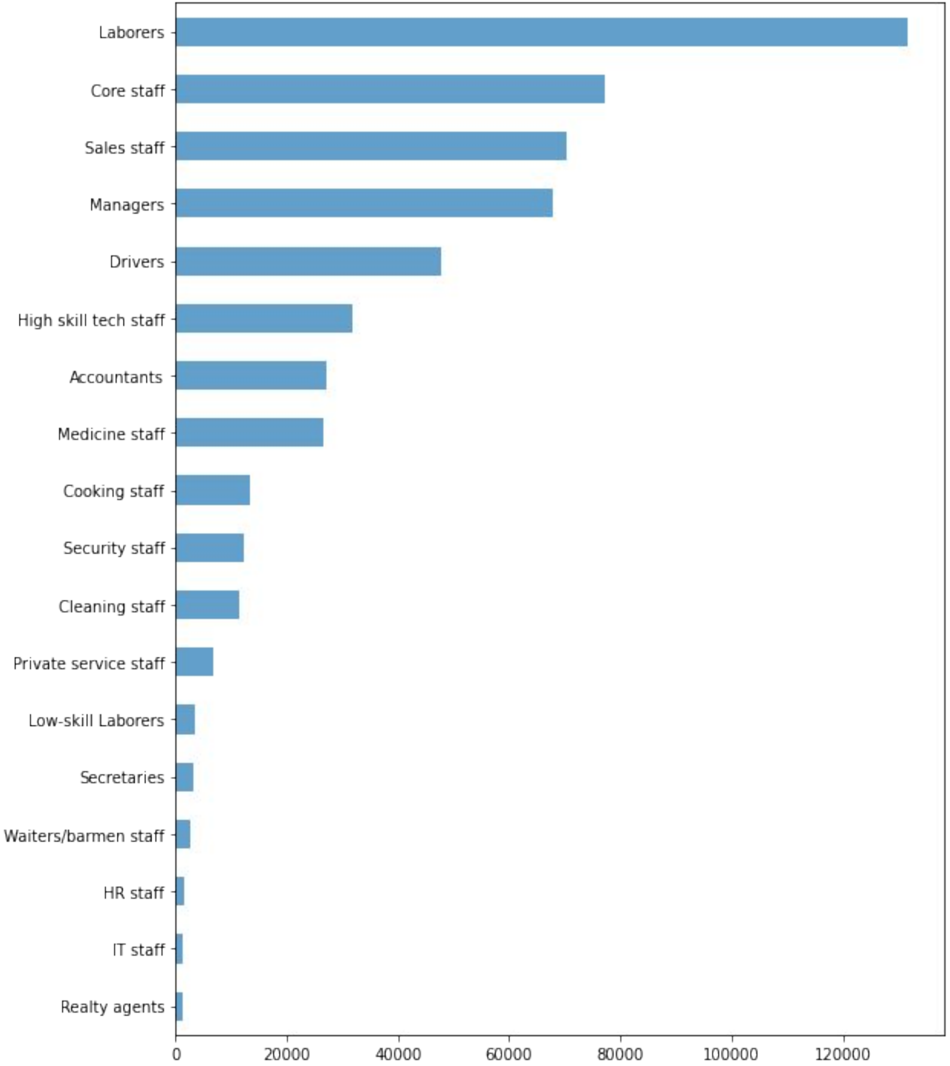
We have 2 datasets here, application_record.csv which has features like ID, Gender, car or realty owned, number of children, annual income, income type, education type, family status, housing type, age, days employed, whether they own a mobile, work and a home phone, email and occupation type, number of family members. credit_record.csv has ID, months of balance left and the status of payment

Data Preprocessing

1. We first merged both our datasets using left join on ‘ID’
2. We removed all rows with NULLs
3. We used column ‘MONTHS_BALANCE’ and binned values into a new column ‘Risk’ more than -3 to be “no” risk and values less than -3 to be “yes” risk.
4. For our target variable we created a column ‘Label’ where the column with STATUS is 0, 1, 2, 3, 4, 5 means 1 (don’t approve) or else 0 (approve)
5. We then performed data visualization to understand the data and infer trends
6. We used LabelEncoder to convert all the variables in consideration to labels or classes so they can be input into the classification models as features.
7. We used train_test_split to split our dataset with our features and target into X_train, y_train, X_test and y_test in the ratio 80:20

Data Visualization

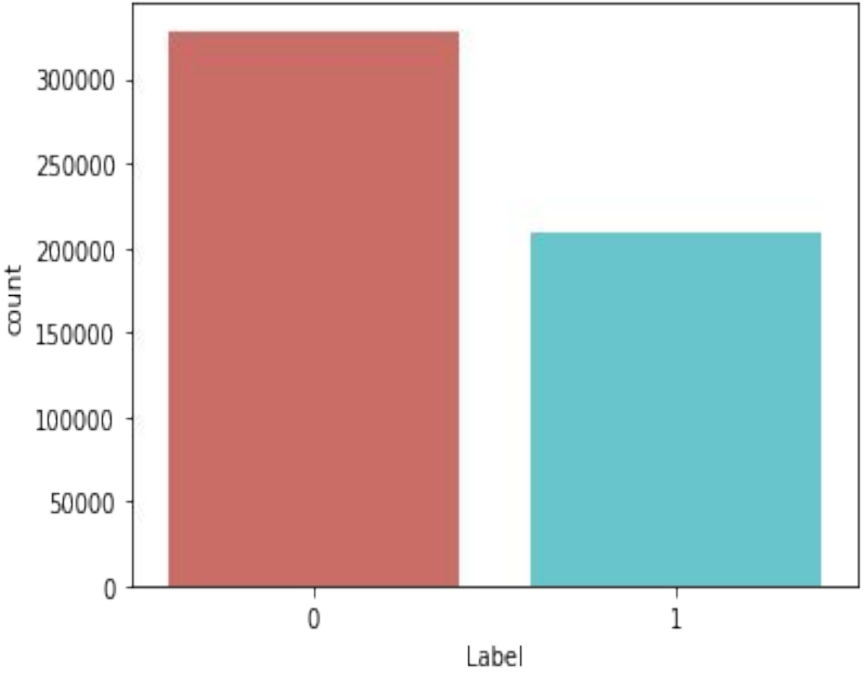
Let’s see the various occupation types and the count of customers in each type. We see that laborers and core staff are the highest number of applicants for credit cards



Data Visualization

Let’s see the count of each class we have (0s and 1s) and observe the number of customers who got their credit cards approved

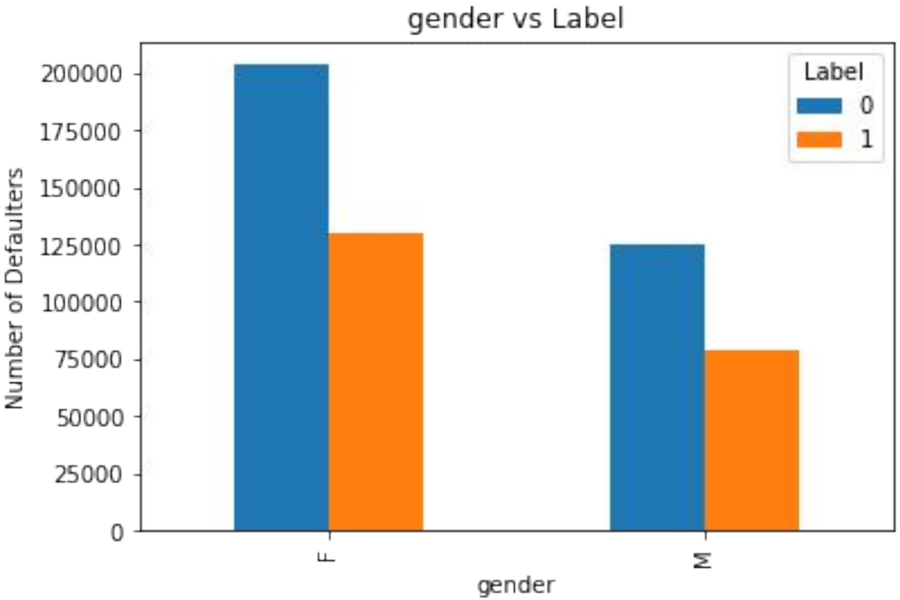
We can see that more credit cards were approved then denied



Data Visualization

Let’s see the count of the number of defaulters as opposed to the ones that didn’t with respect to Gender.

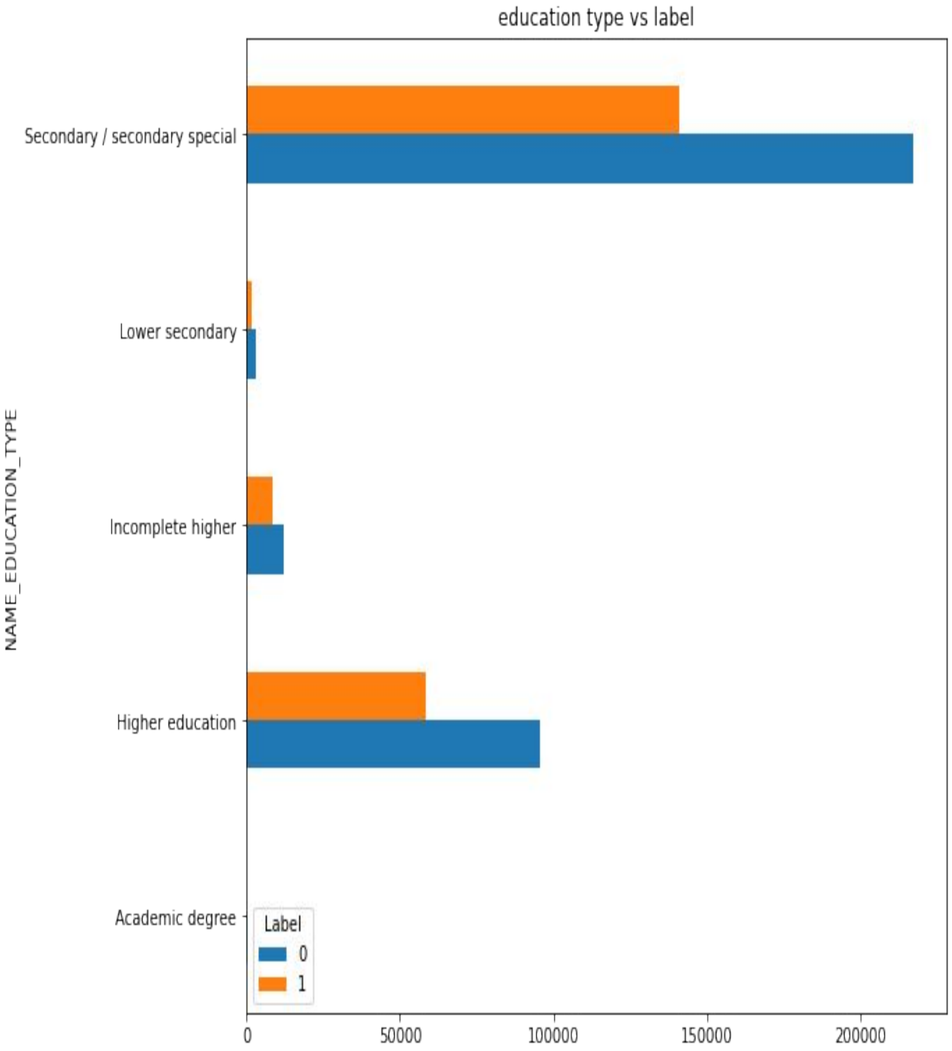
We can see that there were more female applicants and in general more customers in both male and female were approved credit cards as opposed to those who weren’t approved credit cards



Data Visualization

Let’s see the count of the number of defaulters as opposed to the ones that didn’t with respect to the education type.

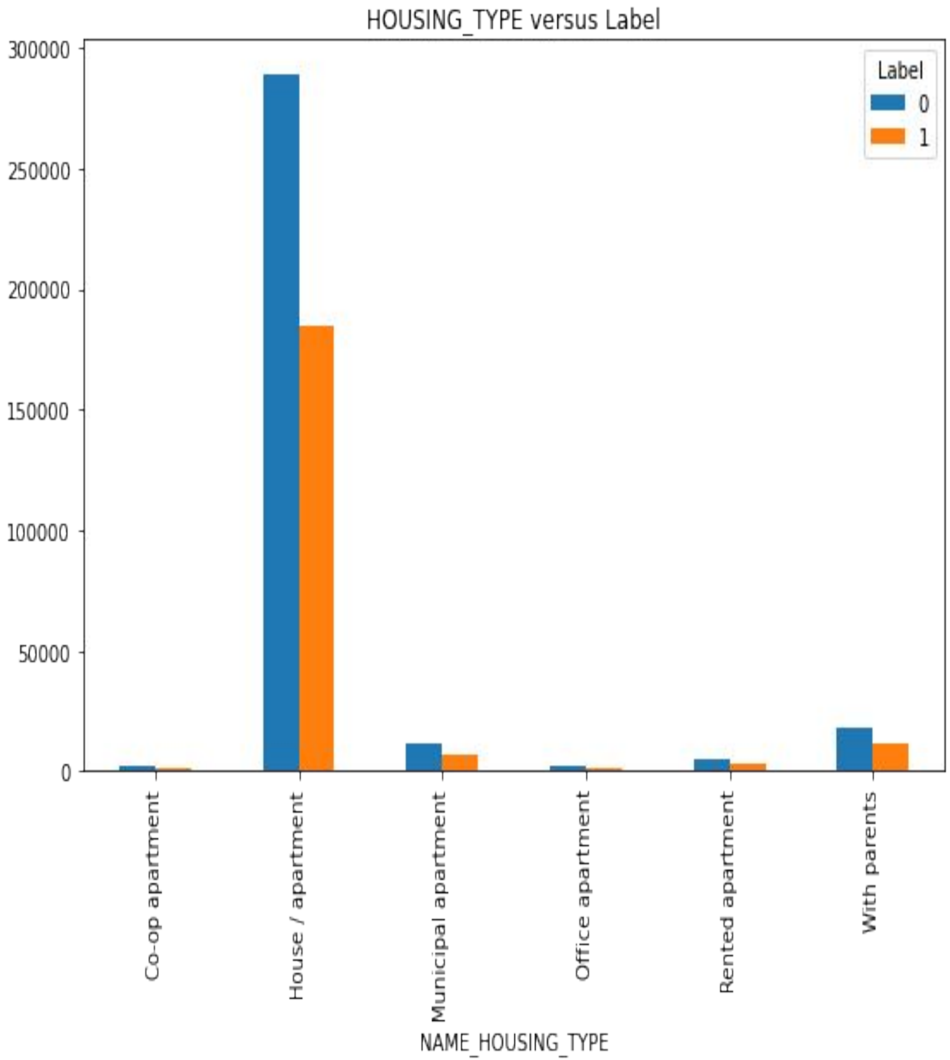
We can see that there were more customers with secondary and higher education that applied for credit cards as opposed to just those with lower secondary or incomplete education



Data Visualization

Let’s see the count of the number of defaulters as opposed to the ones that didn’t with respect to the housing type.

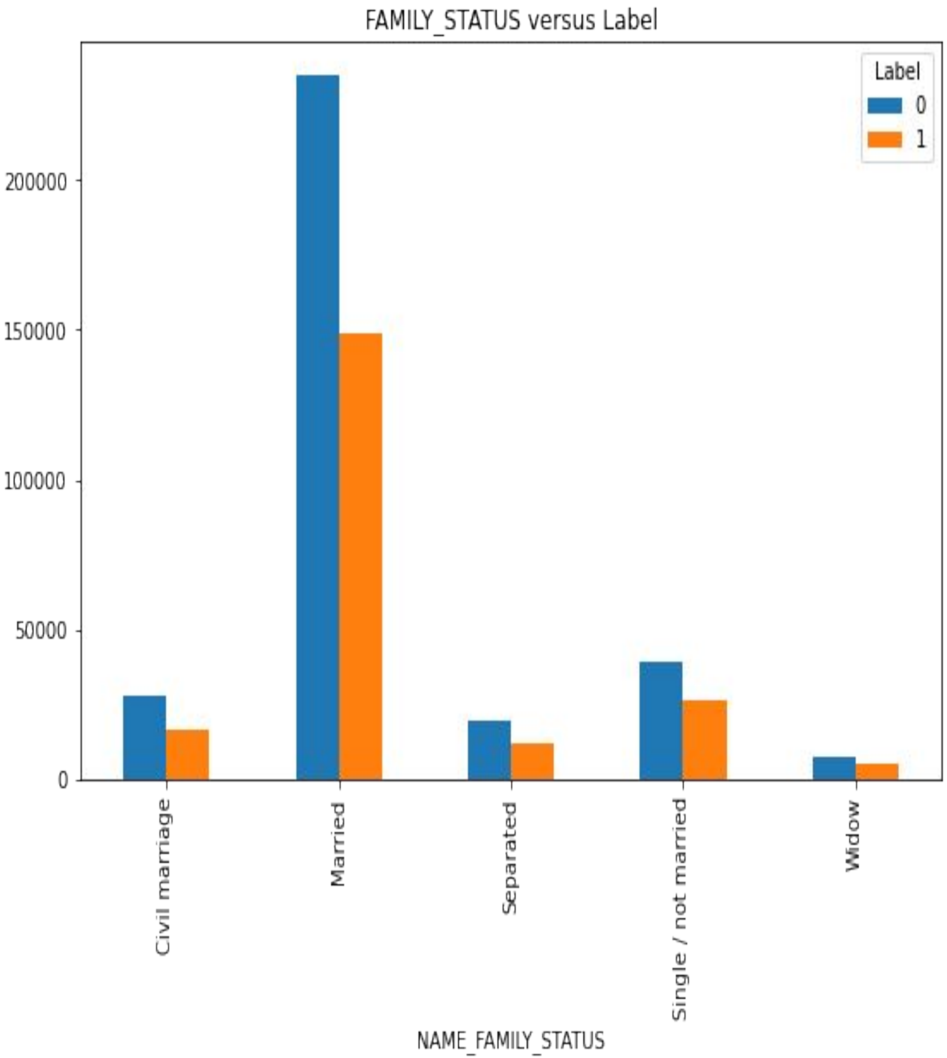
We can see that there were majority of customers that lived in a single house or apartment that applied for credit cards



Data Visualization

Let’s see the count of the number of defaulters as opposed to the ones that didn’t with respect to the family status.

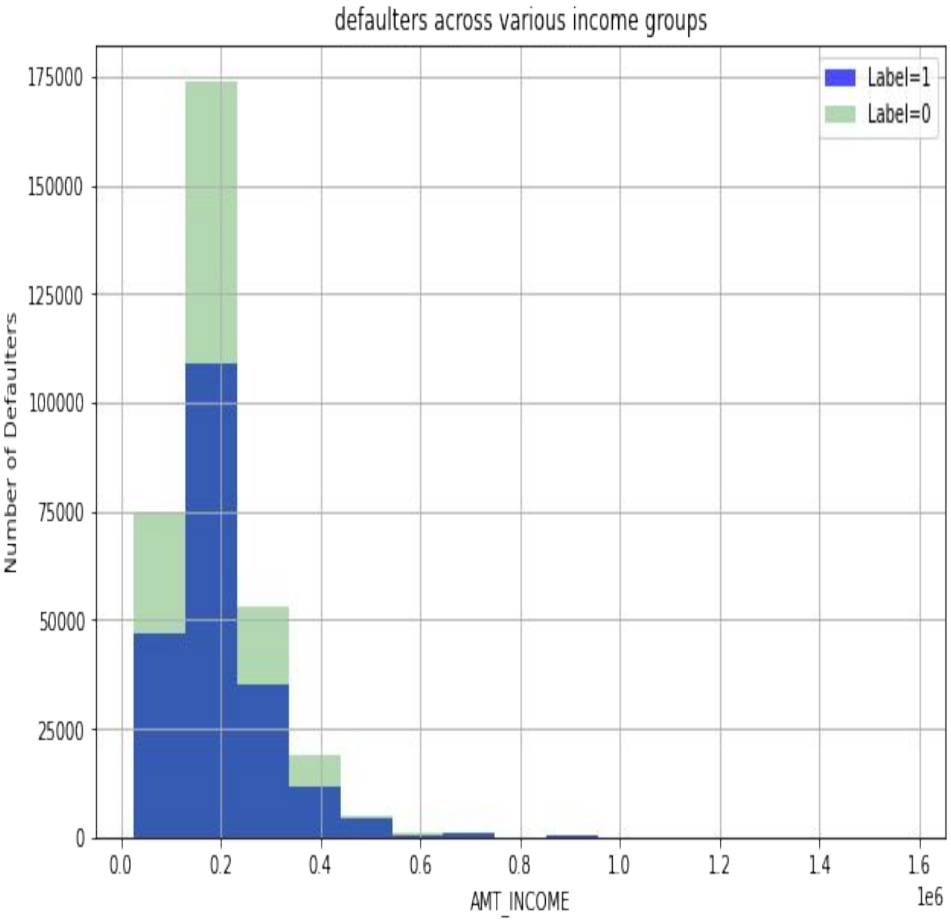
We can see that there were majority of customers that applied for credit cards were married.

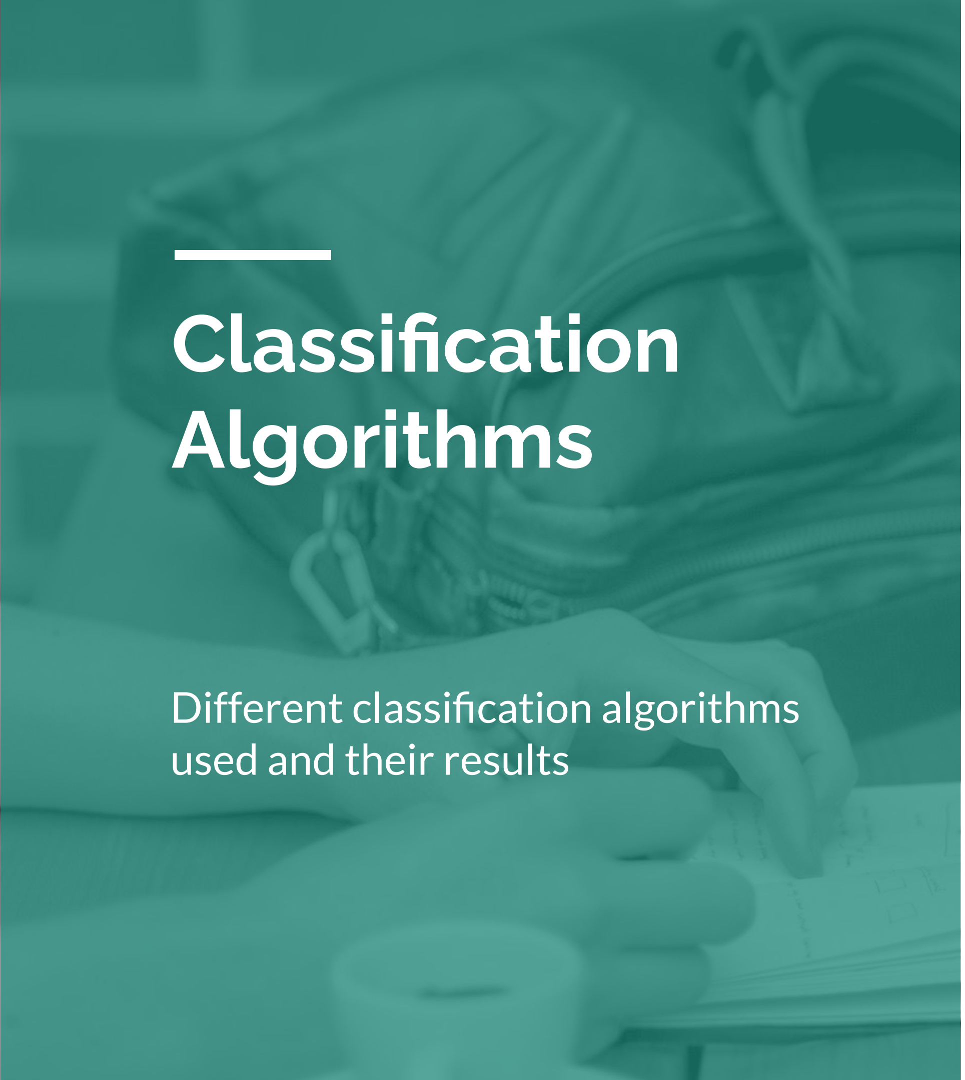


Data Visualization

Let's see the count of the number of defaulters as opposed to the ones that didn't with respect to annual income.

We can see that there were majority of customers that applied for credit cards had approximate annual incomes of $200000





Classification Algorithms

Different classification algorithms used and their results

Gaussian Naive Bayes

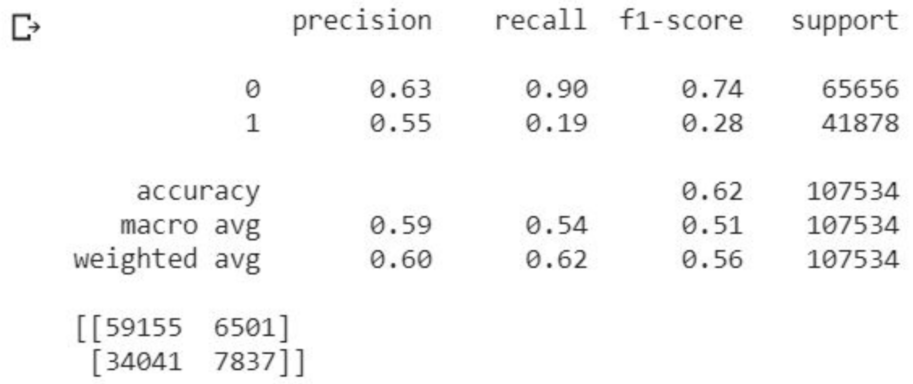
Random Forest

XGBoost Classifier

Randomized Search on XGBoost

Gaussian Naive Bayes Classifier

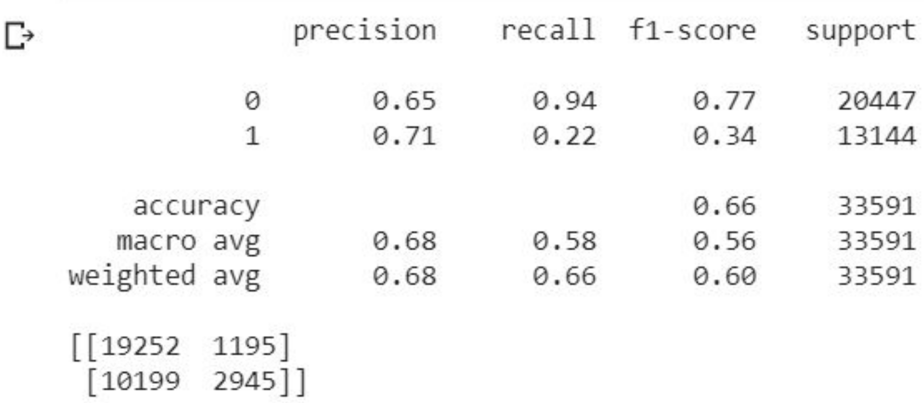
Gaussian Naive Bayes is an algorithm having a Probabilistic Approach. It involves prior and posterior probability calculation of the classes in the dataset and the test data given a class respectively. Prior probabilities of all the classes are calculated using the same formula



```
              precision    recall  f1-score   support

           0       0.63      0.90      0.74     65656
           1       0.55      0.19      0.28     41878

    accuracy                           0.62    107534
   macro avg       0.59      0.54      0.51    107534
weighted avg       0.60      0.62      0.56    107534

[[59155  6501]
 [34041  7837]]
```

Random Forest

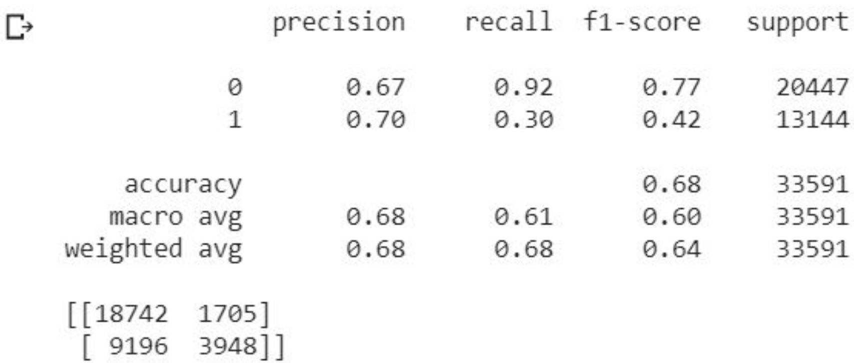
Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean/average prediction of the individual trees.



```
              precision    recall  f1-score   support

           0       0.65      0.94      0.77     20447
           1       0.71      0.22      0.34     13144

    accuracy                           0.66     33591
   macro avg       0.68      0.58      0.56     33591
weighted avg       0.68      0.66      0.60     33591

[[19252  1195]
 [10199  2945]]
```

XGBOOST Classifier

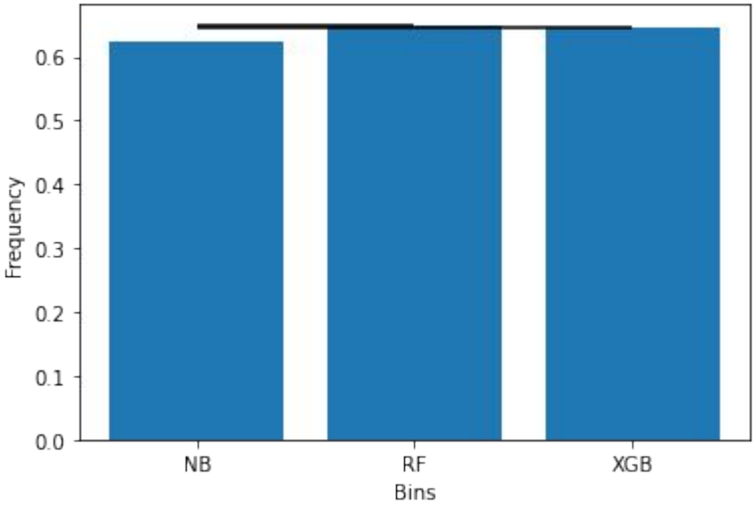
XGBoost is termed as Extreme Gradient Boosting Algorithm which is again an ensemble method that works by boosting trees. XGboost makes use of a gradient descent algorithm which is the reason that it is called Gradient Boosting.



```
              precision    recall  f1-score   support

           0       0.67      0.92      0.77     20447
           1       0.70      0.30      0.42     13144

    accuracy                           0.68     33591
   macro avg       0.68      0.61      0.60     33591
weighted avg       0.68      0.68      0.64     33591

[[18742  1705]
 [ 9196  3948]]
```

Comparison of all models

```
NB: 0.62 (0.000405)
RF: 0.65 (0.001779)
XGB: 0.65 (0.000438)
```

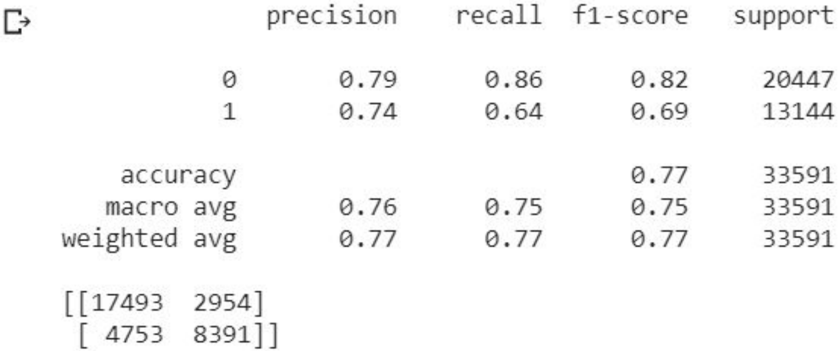


Hyperparameter Tuning & Best Params

RandomizedSearchCV implements a “fit” and a “score” method. It also implements “predict”, “predict_proba”, “decision_function”, “transform” and “inverse_transform” if they are implemented in the estimator used. The parameters of the estimator used to apply these methods are optimized by cross-validated search over parameter settings. In contrast to GridSearchCV, not all parameter values are tried out, but rather a fixed number of parameter settings is sampled from the specified distributions. The number of parameter settings that are tried is given by n_iter.

```
{'subsample': 0.9, 'silent': False,
'reg_lambda': 10.0, 'n_estimators': 100,
'min_child_weight': 0.5, 'max_depth': 15,
'learning_rate': 0.2, 'gamma': 0.5,
'colsample_bytree': 0.6, 'colsample_bylevel':
0.4}
```

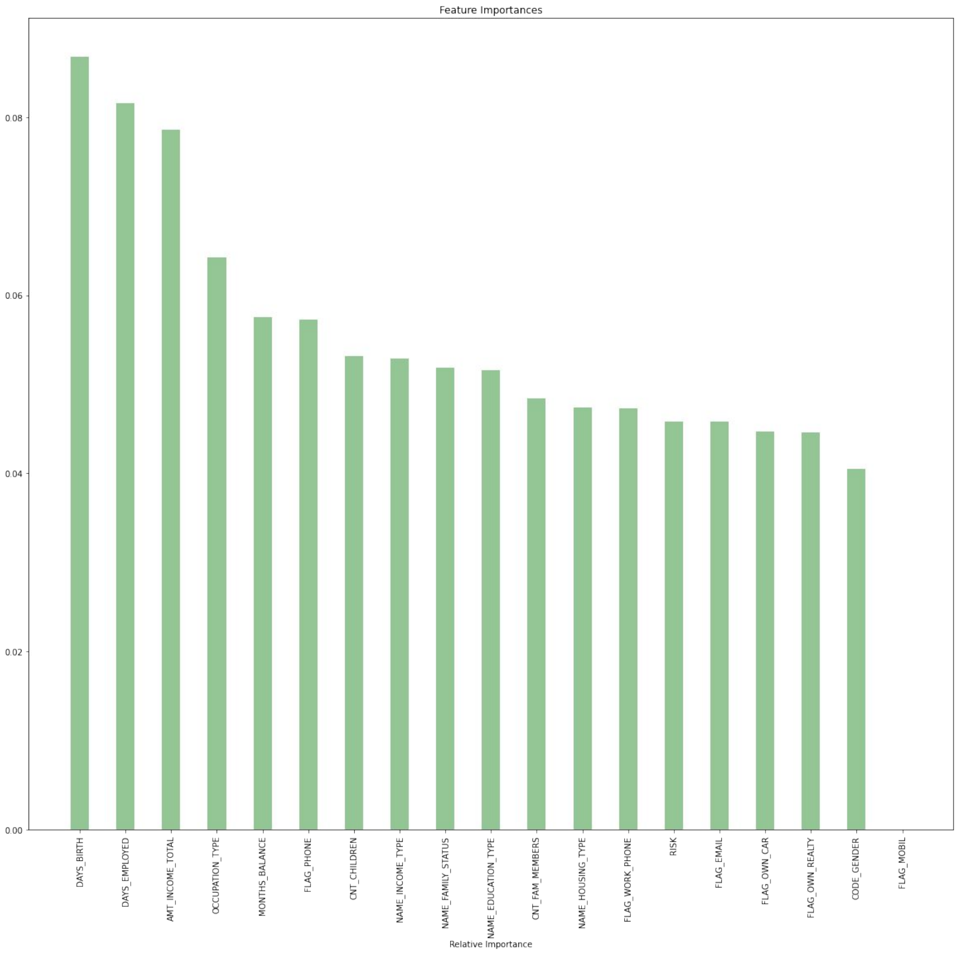
Using Randomized Search on XGBoost



```
              precision    recall  f1-score   support

           0       0.79      0.86      0.82     20447
           1       0.74      0.64      0.69     13144

    accuracy                           0.77     33591
   macro avg       0.76      0.75      0.75     33591
weighted avg       0.77      0.77      0.77     33591

[[17493  2954]
 [ 4753  8391]]
```

Feature Importances

We use XGBoost’s feature importances function to estimate the most important features weighing in determining our target variable for every customer. The age, number of days employed and the annual income are crucial features as seen from the graph



Conclusion

- With less liability and higher education, applicants tend to get their credit cards approved
- More married couples with no children tend to apply for credit cards
- Using XGBoost with RandomizedSearchCV's best parameters gives us an outstanding accuracy of 77% as compared to other algorithms
- When we assessed the most important features that help determine our outcome, whether a credit card should be approved or not, we saw that age, days since employed and annual income played a crucial role
- We believe that the model we built will be able to perform well in determining an applicant's credit rating, risk factor and a decision on whether to approve credit card or not given the applicant's history

Questions?

References

https://www.kaggle.com/rikdifos/credit-card-approval-prediction

https://scikit-learn.org/stable/modules/naive_bayes.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://xgboost.readthedocs.io/en/latest/python/python_api.html

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html