Determining Client Credibility

By: Dama D. Daliman

What's the problem?

In a credit bureau we often need to assess the credibility of potential clients from collected data of that client. Doing this manually can often be tiring and possibly lead to human errors.

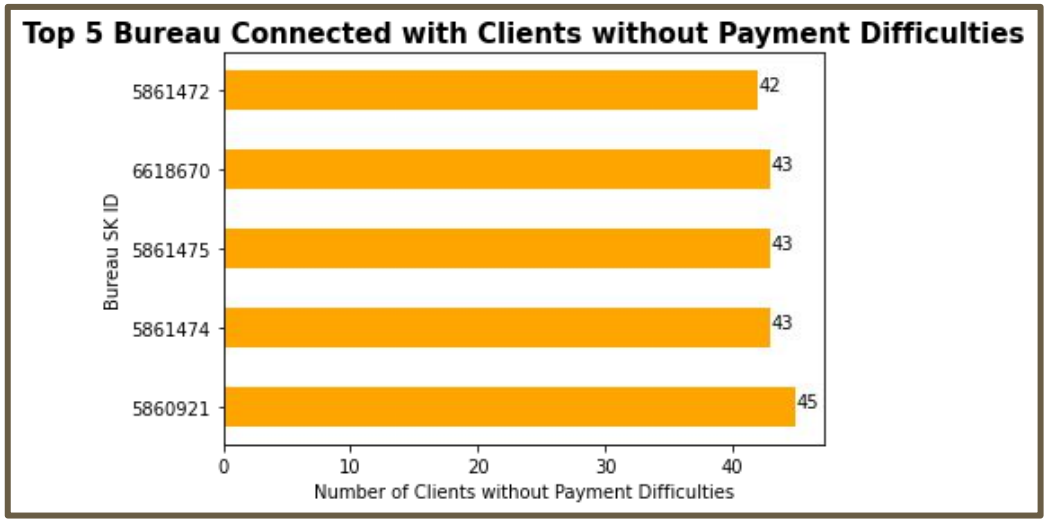
So the goal of this analysis and experiment is to provide insights and a machine learning model that can help determine clients' credibility.

The Dataset

The dataset is composed of application, bureau, credit card, installments payments, POS Cash balance, and previous application data.

The main analysis and experiment will be done on the application data. The other data will be explored as well, only for insight purposes.

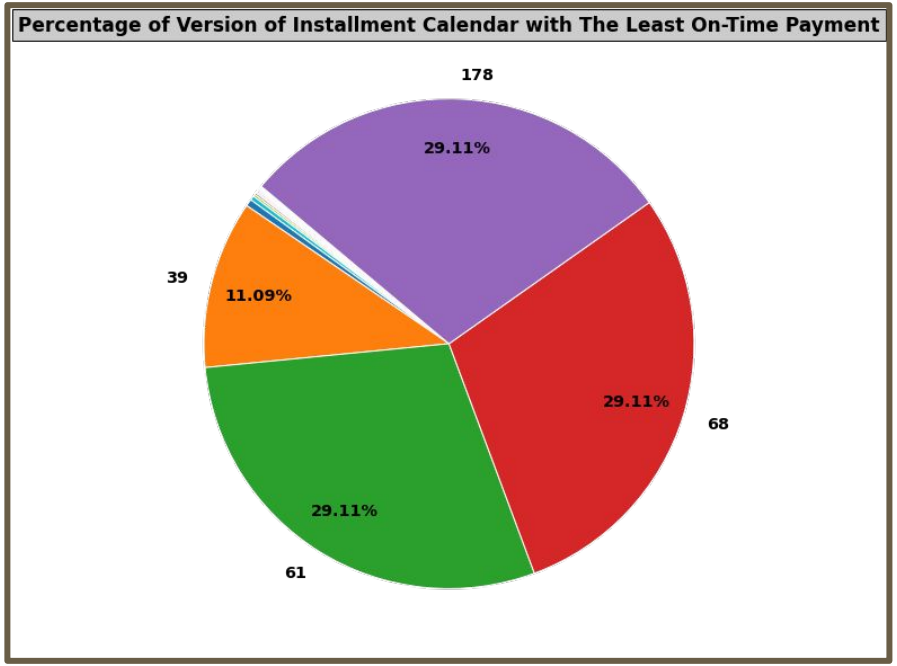
Interesting Insights



From the bureau data, we found out that clients who have no difficulty in payments were mostly affiliated with credit bureaus of SK ID 5860921, 5861474, 5861475, 6618670, and 5861472. As such, in the future we might consider collaborating with the bureau's mentioned above or look for potential clients who are affiliated with said bureau's.

Interesting Insights

The next insight we found was that version installment calendar number 178, 68, and 61 were the least installment calendar version with on-time payment. In other words, clients whose installment calendar version are number 178, 68, and 61 should be followed up more often by the staff, to ensure on-time payment of the credit.



Model Building - Data Preprocessing

Before building a model, we preprocessed or cleaned the data first. The data cleaning was comprised of 5 steps, namely:

1. Dropping single value columns with drop() and nunique()
2. Checking for duplicate rows with duplicated()
3. Checking and clearing outliers with IQR scores
4. Converting categorical data using pandas get_dummies()
5. Imputing missing data using sklearn's Simple Imputer

Model Building - Feature Engineering

After the data is preprocessed, now it is time to do some feature engineering. We select the most relevant features from the large dataset, so that we can effectively and efficiently train our model.

The feature engineering is done using three methods

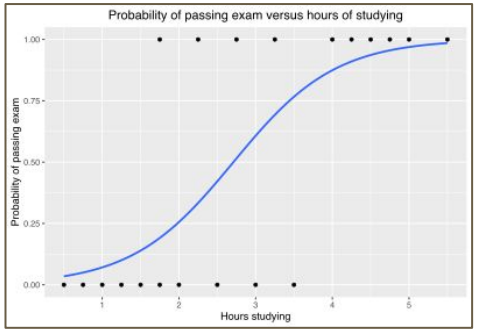
1. Correlation Matrix Heatmap
2. Univariate Selection with ANOVA f-score
3. Feature Importances

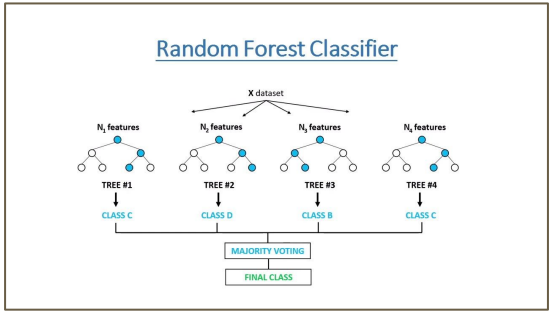
Model Building - Algorithm Selection

When building a model, we need to decide how many machine learning algorithms do we need? Which machine learning algorithms? And why?

In this project, we selected 3 supervised binary classification algorithms, which are:

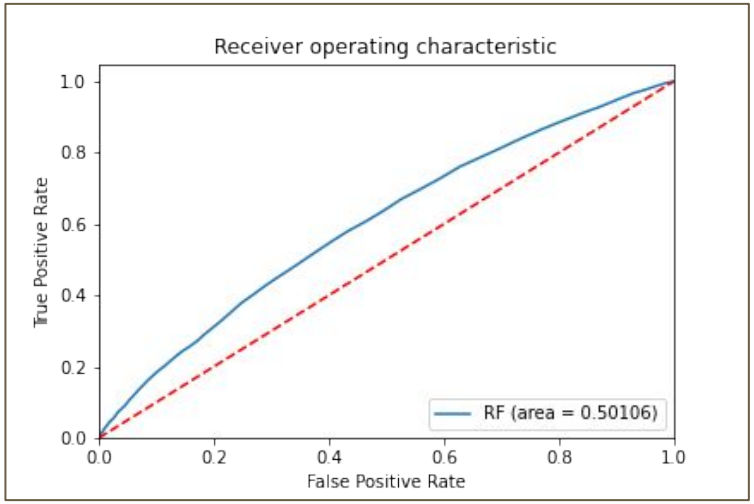
1. Logistic Regression
2. K-Nearest Neighbors
3. Random Forest





Model Evaluation and Prediction

After evaluating the models made from the 3 algorithms by their accuracy score and area under the ROC curve, followed by some hyperparameter customization. We finally decided on using the Random Forest algorithm with 225 nodes and random state of 0*. This model performs best on the validation set with an accuracy score of 0.90922 and area under the ROC curve of 0.50106 (0.00106 lead over the other algorithms). We then used this model to do a prediction on the test set**.



*Hyperparameter details can be checked in the project's jupyter notebook

**The prediction results can be viewed in the project's jupyter notebook

Reference and GitHub Repository

Links to knowledge and methods can be found below:

https://www.projectpro.io/recipes/select-features-using-best-anova-f-values-in-python

https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/

https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/

https://www.statology.org/what-is-a-good-auc-score/#:~:text=0.5%2D0.7%20%3D%20Poor%20discrimination,0.8%2D0.9%3D%20Excellent%20discrimination

https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

https://scikit-learn.org/stable/modules/model_evaluation.html#multimetric-scoring

You can access the GitHub Repository containing the Jupyter Notebook used in this analysis with this link:

Github Repo Link