

# Analyzing Home Credit Default Risk

Muhammad Kashif  
Audrey Nkrumah  
Lovepreet Singh  
Vinay Vattipally



# Project Overview

- Lots of folks find it hard to borrow money because they don't have enough or any credit history. Sadly, some dishonest lenders take advantage of these people.
- This project is aimed at building a highly accurate predictive model that can predict risk probability of default for loan applicants based on the dataset provided by the home credit and we went further to calculate amount of loan customer can get if one is at low risk of default.
- After analyzing the data, which met all the requirement to building this model was met, we divided the task into 4 main parts and distributed amongst group members.
- Data Source: <https://kaggle.com/competitions/home-credit-default-risk>

# Process Task Flow

## ETL & EDA (Extract, Transform, Load & Exploratory Data Analysis)

- Preparing and cleaning the data for analysis and modeling
- Transforming it into a suitable format
- Loading it into the required data structures

## Data Analysis and Visualizations

- Analyzing the data, exploring its characteristics, identifying patterns, trends, and relationships within the data
- Creating visualizations

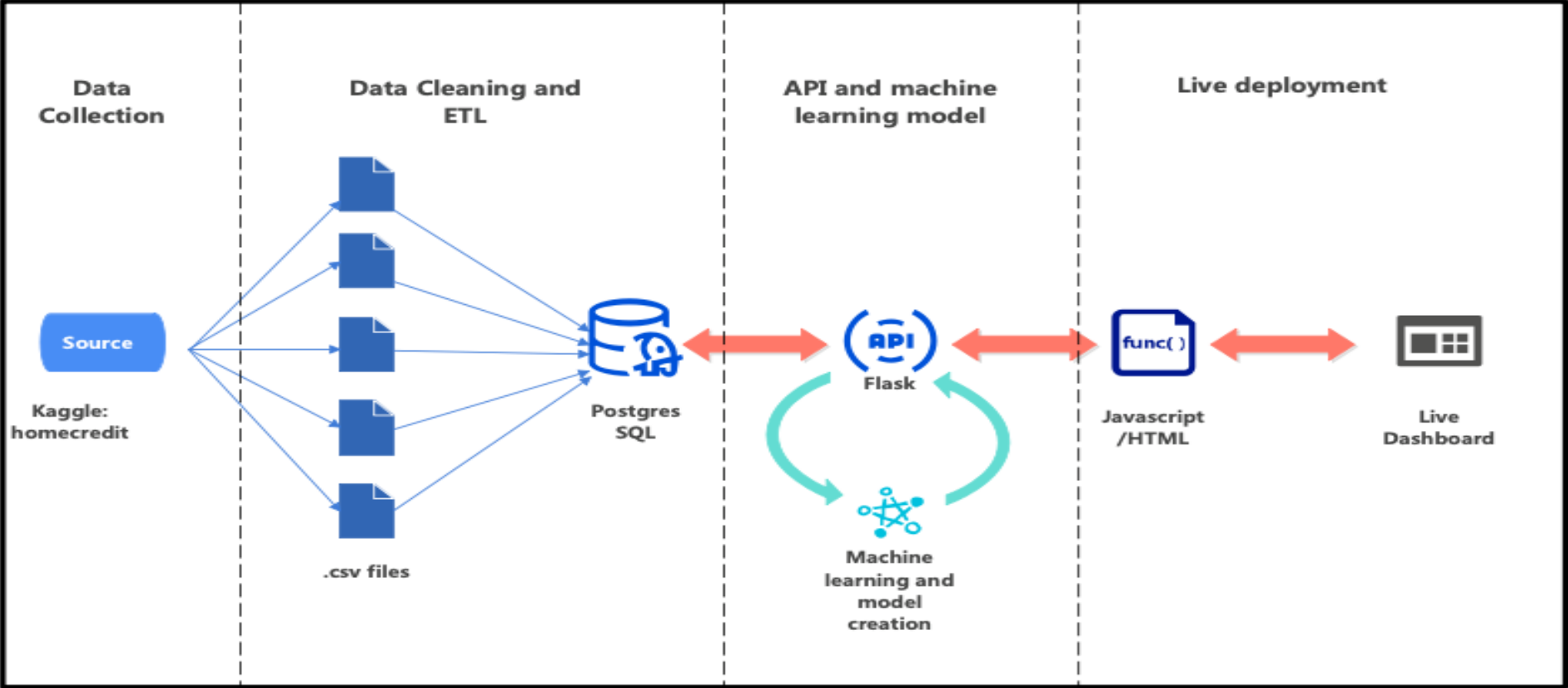
## Machine Learning and Model Integration

- Developing machine learning models based on the analyzed data
- Includes selecting appropriate algorithms, training and evaluating models, fine-tuning parameters, and integrating the models with the front end of the application or system

## Front End

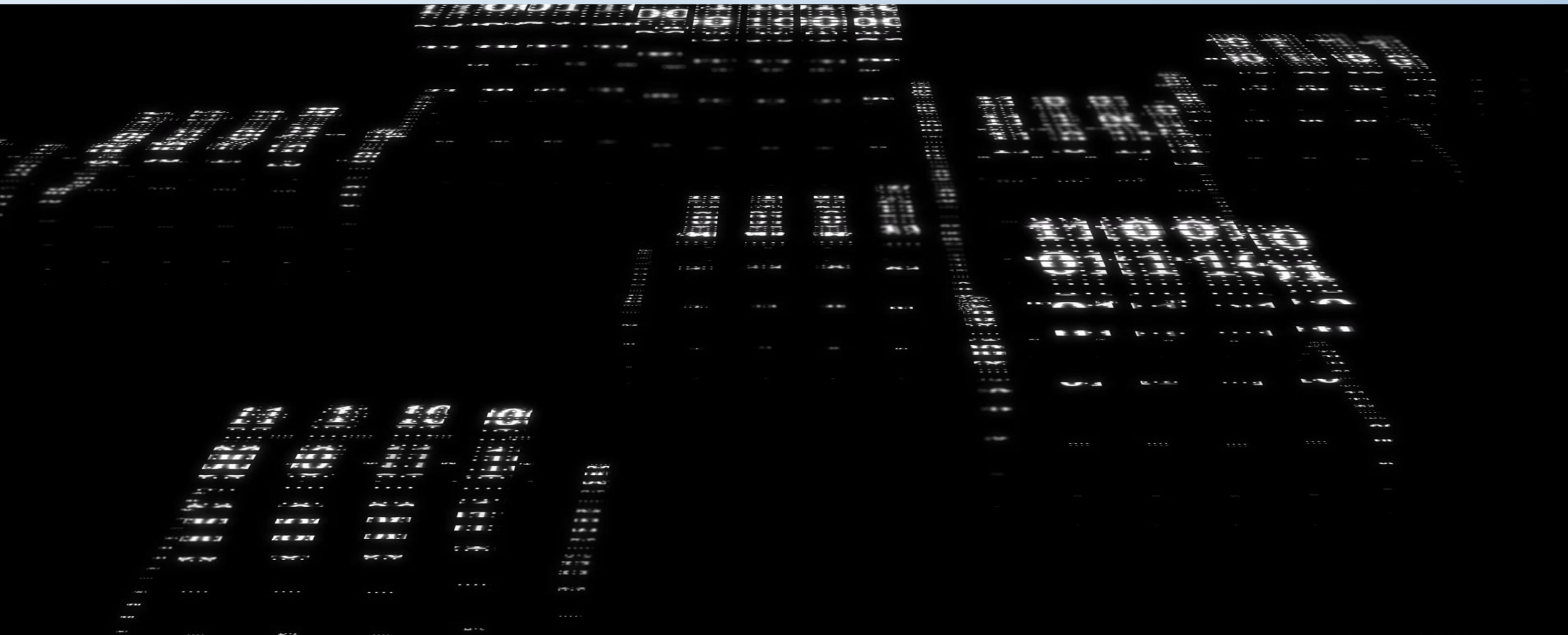
- Designing and developing the user interface (UI) and user experience (UX) components of the application or system.
- Includes creating layouts, implementing functionality, and ensuring a seamless and intuitive user interface.

# Data Architecture





# ETL (Extract, Transform & Load)




# Preparing and cleaning the data for analysis and modeling

- Data source and acquisition process.
- Data is acquired from Kaggle.com provided by Home credit Group.
  - Data contains files:
  - application\_{train/test}.csv
  - bureau.csv
  - bureau\_balance.csv
  - POS\_CASH\_balance.csv
  - credit\_card\_balance.csv
  - previous\_application.csv
  - installments\_payments.csv
  - HomeCredit\_columns\_description.csv
- Description of the dataset: features, target variable
  - Features: Age, Gender, Education Level, Marital Status, Income, Occupation, Number of Dependents, Credit Score, Home Ownership, Loan Purpose, Loan Amount, Loan Term, etc.
  - Target Variable: Target

# Data Preprocessing:

## Handling missing values

- Data obtained from source has categorical data in refined form and minimal bad entries
- However Null values in were filled by putting 0 values in columns where 0 and 1 does not impact our model predictions.
- The before and after of the handling of missing values is as illustrated on the image to the right

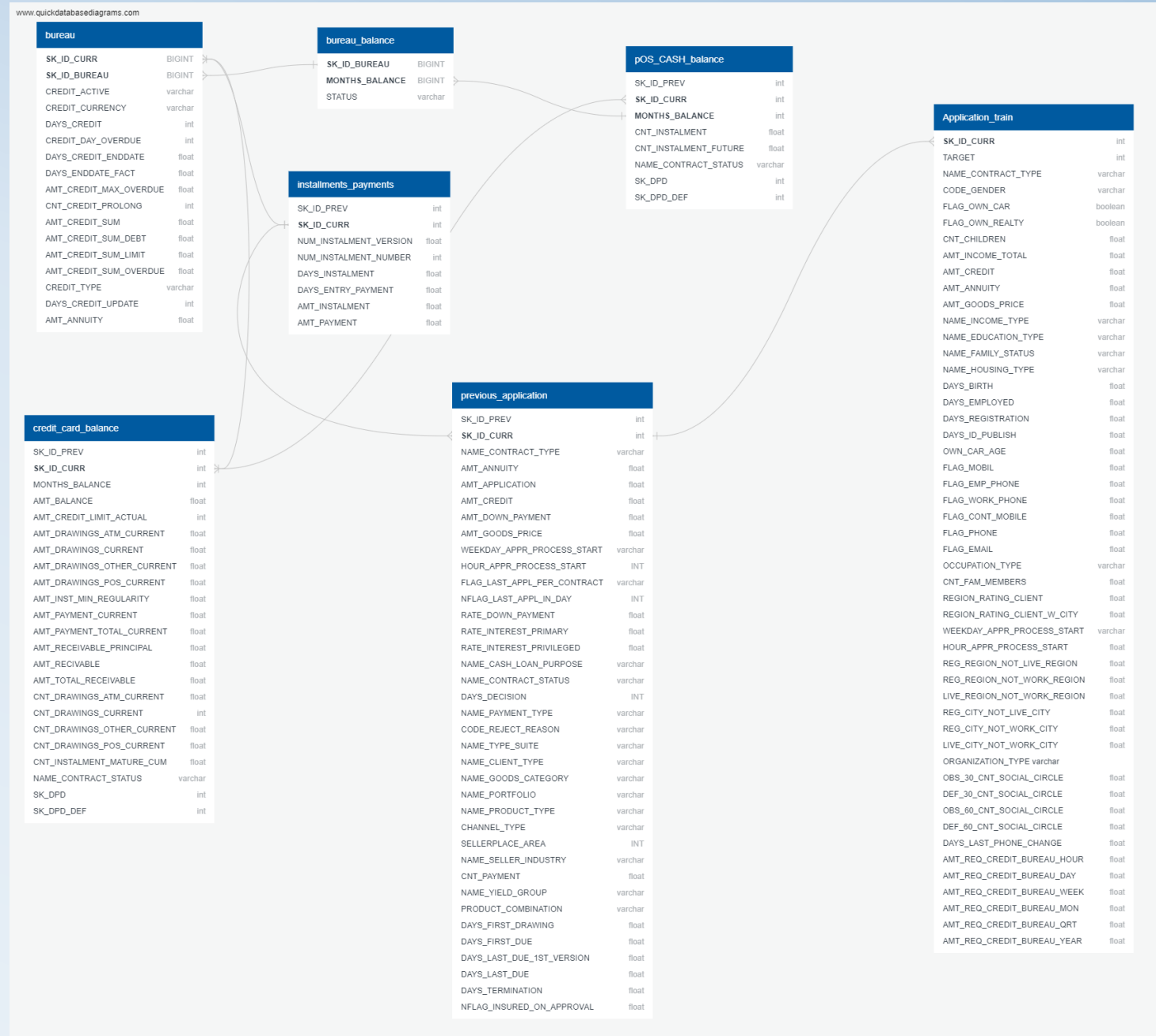


SK_ID_CURR	0	SK_ID_CURR	0
SK_ID_BUREAU	0	SK_ID_BUREAU	0
CREDIT_ACTIVE	0	CREDIT_ACTIVE	0
CREDIT_CURRENCY	0	CREDIT_CURRENCY	0
DAYS_CREDIT	0	DAYS_CREDIT	0
CREDIT_DAY_OVERDUE	0	CREDIT_DAY_OVERDUE	0
DAYS_CREDIT_ENDDATE	105553	DAYS_CREDIT_ENDDATE	0
DAYS_ENDDATE_FACT	633653	DAYS_ENDDATE_FACT	0
AMT_CREDIT_MAX_OVERDUE	1124488	AMT_CREDIT_MAX_OVERDUE	0
CNT_CREDIT_PROLONG	0	CNT_CREDIT_PROLONG	0
AMT_CREDIT_SUM	13	AMT_CREDIT_SUM	0
AMT_CREDIT_SUM_DEBT	257669	AMT_CREDIT_SUM_DEBT	0
AMT_CREDIT_SUM_LIMIT	591780	AMT_CREDIT_SUM_LIMIT	0
AMT_CREDIT_SUM_OVERDUE	0	AMT_CREDIT_SUM_OVERDUE	0
CREDIT_TYPE	0	CREDIT_TYPE	0
DAYS_CREDIT_UPDATE	0	DAYS_CREDIT_UPDATE	0
AMT_ANNUITY	1226791	AMT_ANNUITY	0
dtype: int64		dtype: int64	

# Data Preprocessing cont.

## Data cleaning and formatting

- After Initial cleaning and handling of missing values, CSV files are exported, a schema was created. The schema is illustrated as in the image on the right.





# Data Preprocessing cont.

## Connect to Sources(APIs):

APIs were created using Database queries, Final schema file is used to create tables in Postgresql. Tables are populated with data by using postgresqlpop.ipynb file which have functions to insert data in the database from CSV files.

The app.py file has query setup to obtain data from Postgresql via APIs

- Below is list of APIs:

"Bureau" : <http://127.0.0.1:5000/bureau>

"Bureau\_balance" : [http://127.0.0.1:5000/bureau\\_balance](http://127.0.0.1:5000/bureau_balance)

"Credit\_card\_balance" :  
[http://127.0.0.1:5000/credit\\_card\\_balance](http://127.0.0.1:5000/credit_card_balance)

"Installments\_payments" :  
[http://127.0.0.1:5000/installments\\_payments](http://127.0.0.1:5000/installments_payments)

"POS\_CASH\_balance" :  
[http://127.0.0.1:5000/pOS\\_CASH\\_balance](http://127.0.0.1:5000/pOS_CASH_balance)

"Previous\_application" :  
[http://127.0.0.1:5000/previous\\_application](http://127.0.0.1:5000/previous_application)

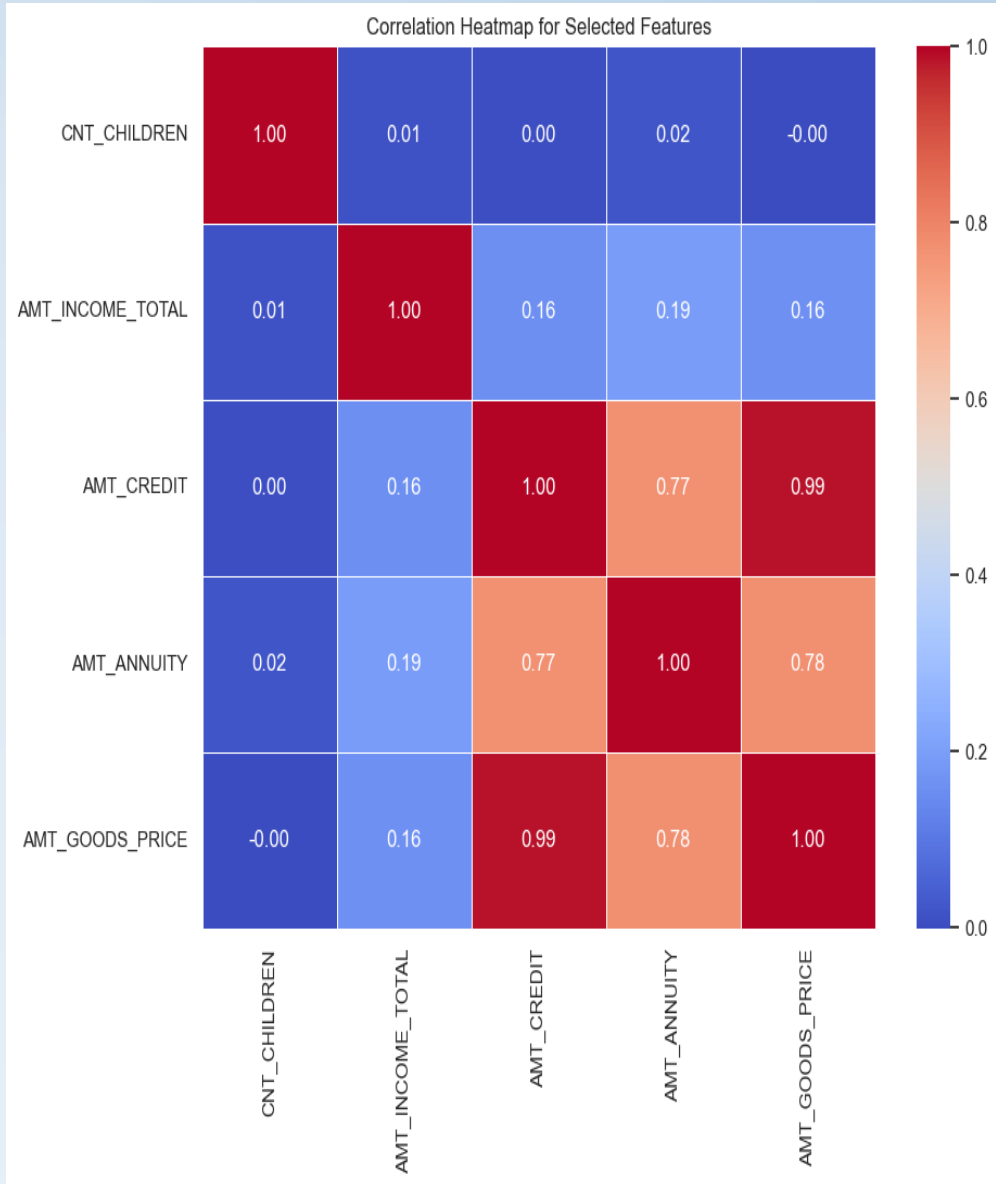
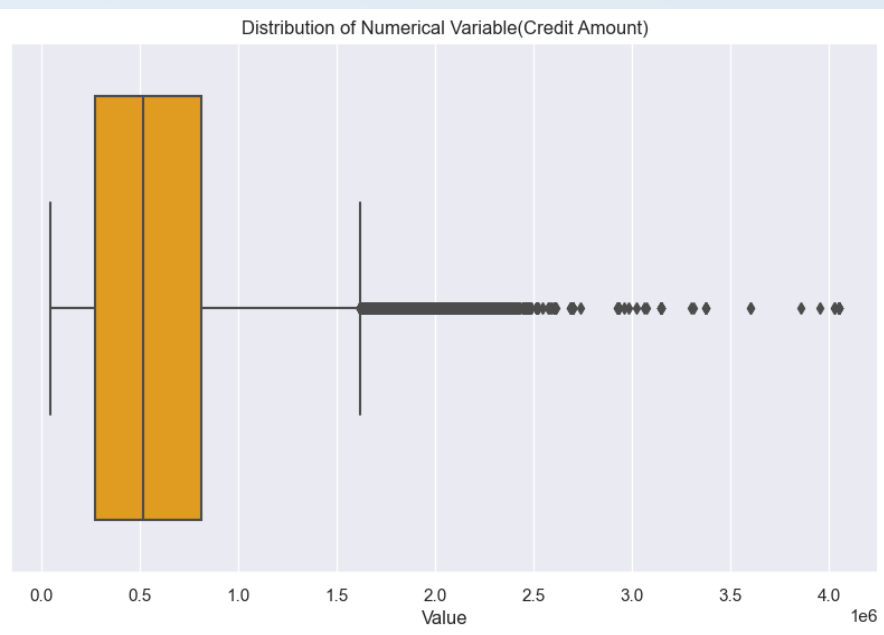
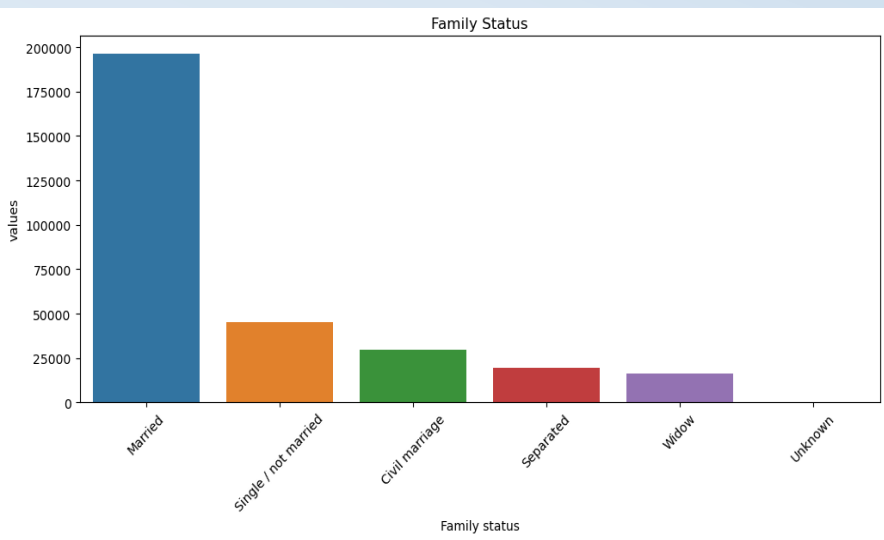
"Application\_train" : [http://127.0.0.1:5000/application\\_train](http://127.0.0.1:5000/application_train)

# Data Preprocessing cont.

## Data Retrieval from APIs:

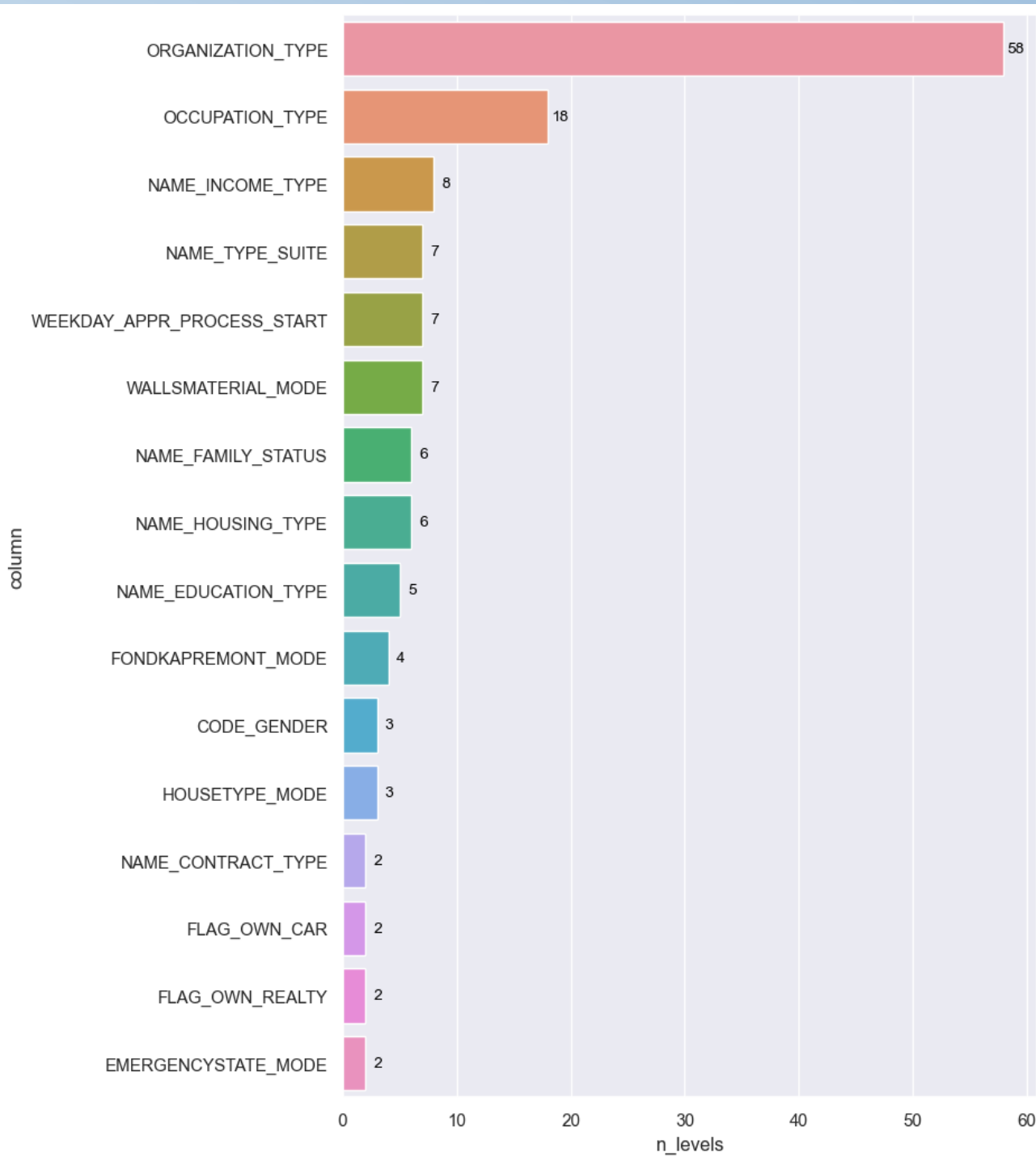
- Data is extracted from API and tables are merged together using model\_probab.ipynb & model\_predict.ipynb file for respective modelling purpose. -"POS\_CASH\_balance", "Installments\_payments", "Credit\_card\_balance", "Previous\_application" were merged on SK\_ID\_PREV and then the "Application\_train" & "Bureau" on SK\_ID\_CURR while Bureau is attached with Bureau\_balance at SK\_ID\_bureau.
- Data is then used for modelling process and Data analysis.

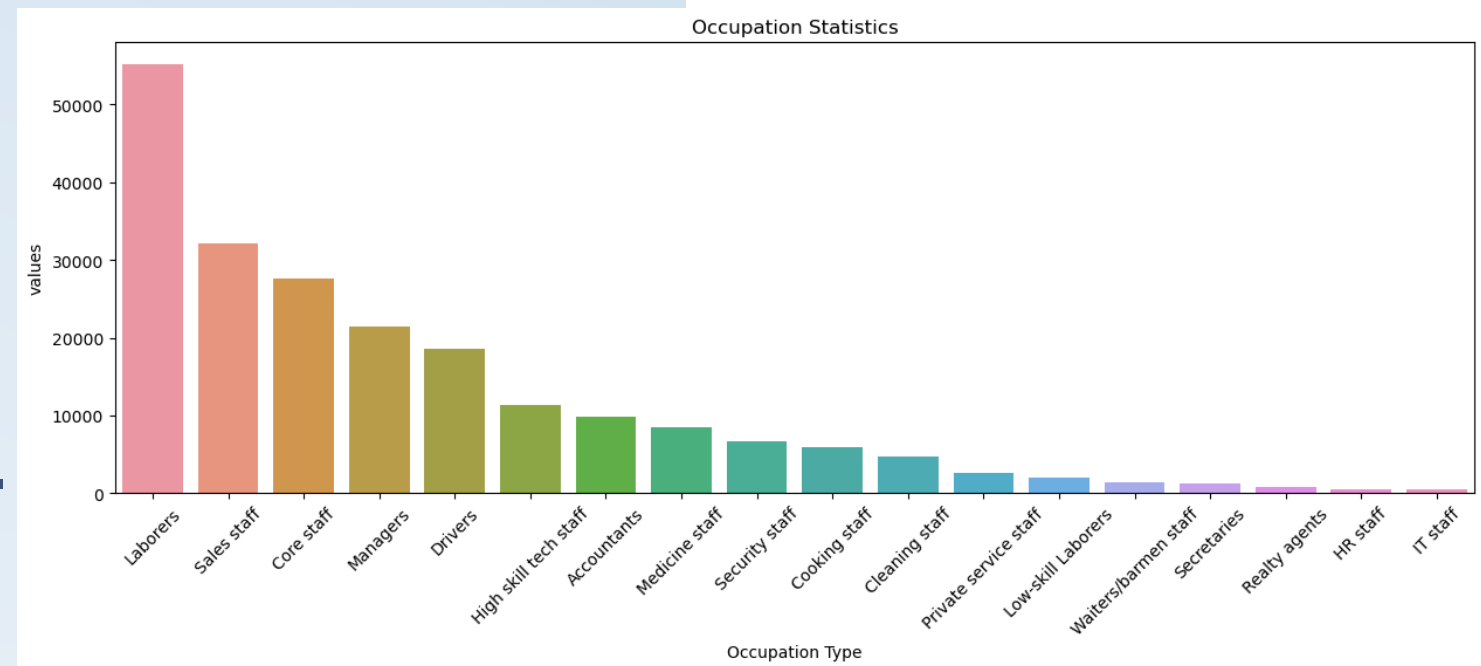
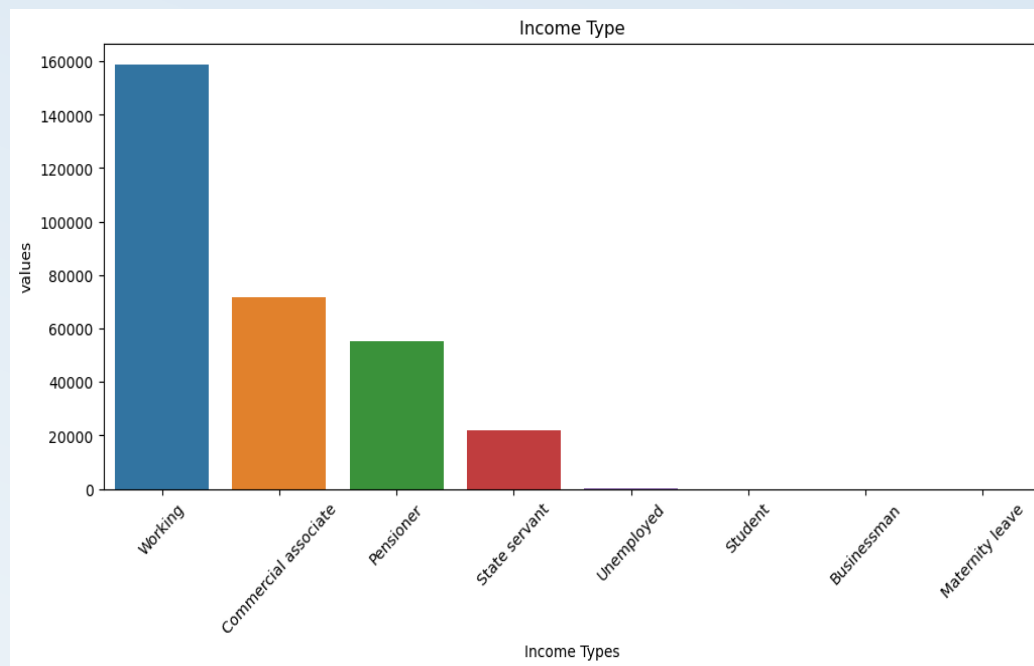
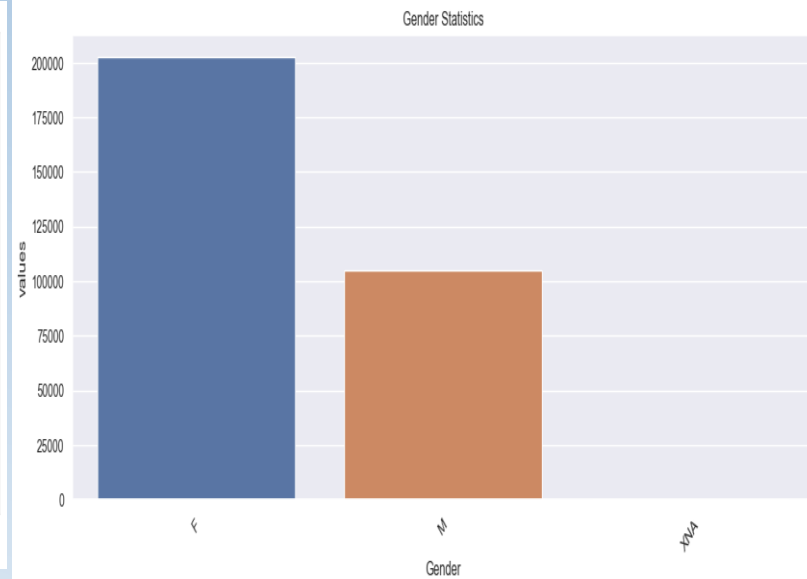
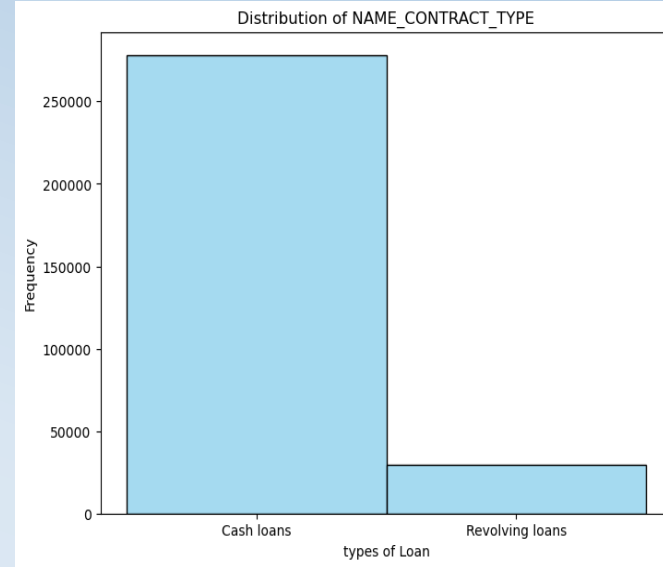
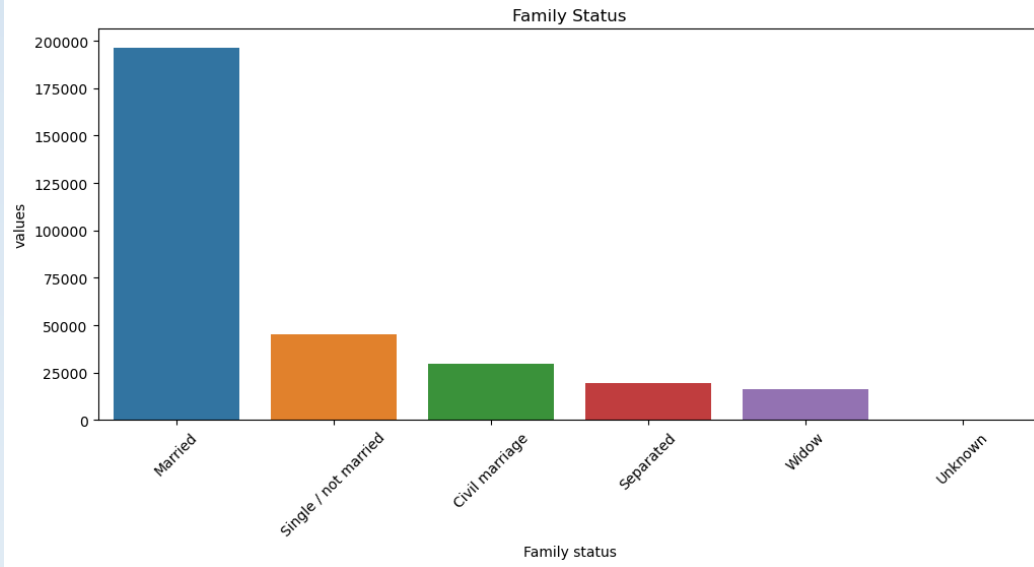
# Data Analysis and Visualizations



# Understanding Data

- The categorical Features were examined, and the unique number of levels was obtained.
- It is interesting to note that origination type had the highest number (58 ) unique levels.
- Followed by Occupation type with 18 unique levels then Income type and so on as depicted in the image on the right.
- The least number of unique levels was 2 in contract type, car and the likes.



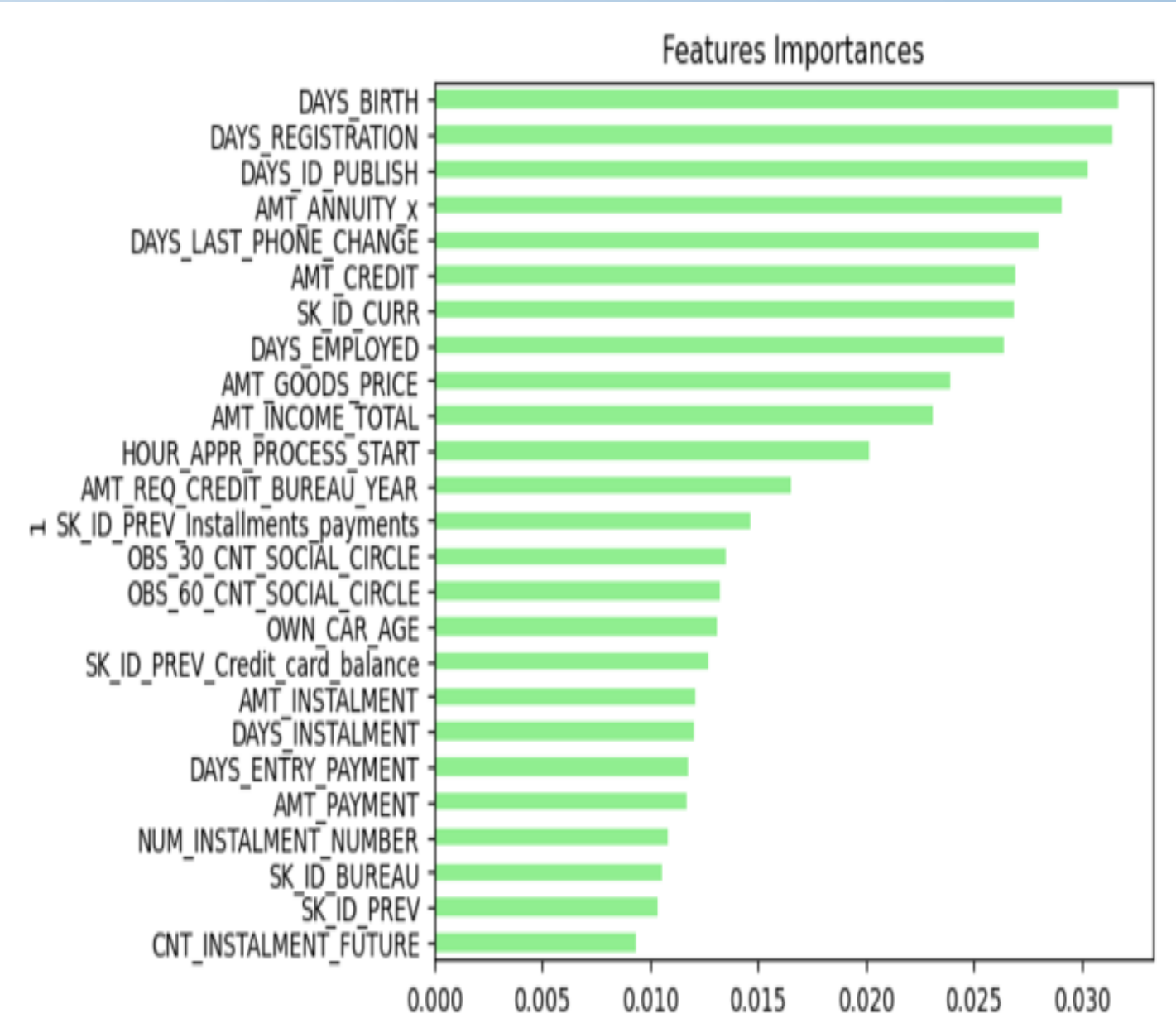


Some Interesting iteration of the Categorical Features into their various levels

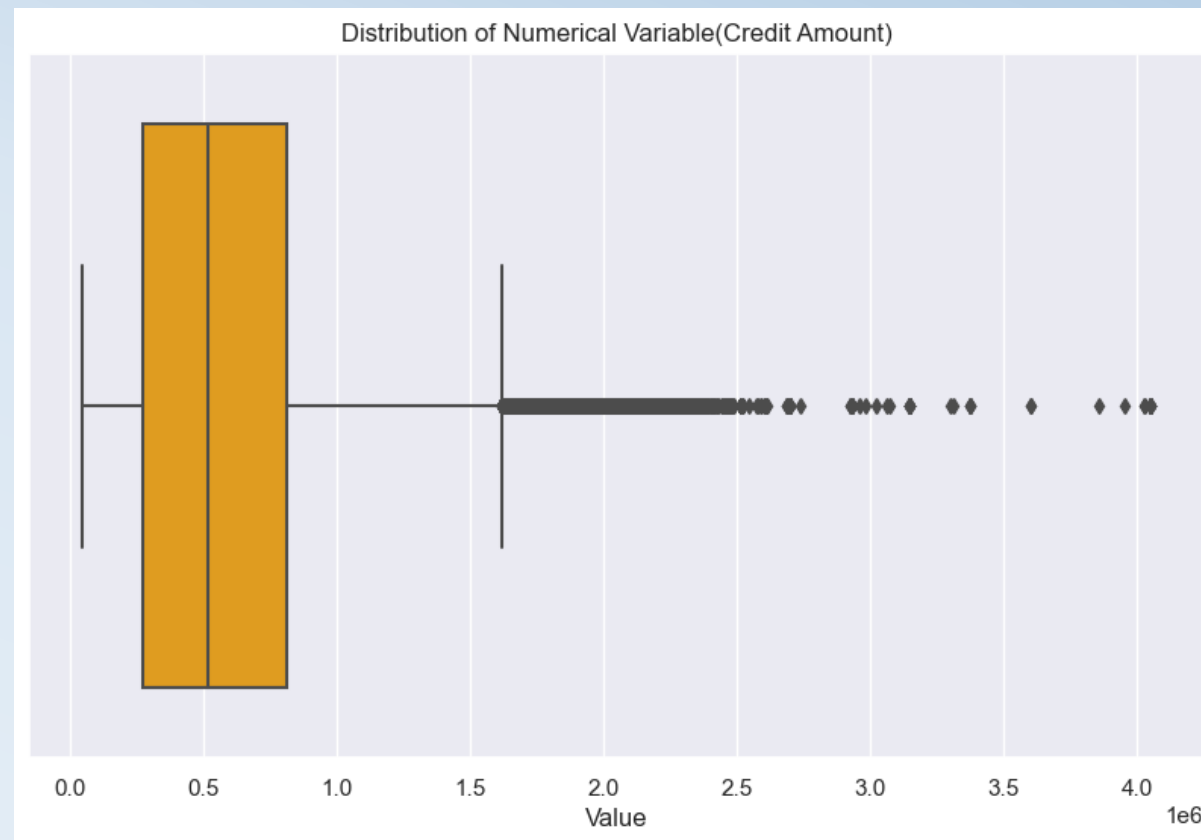
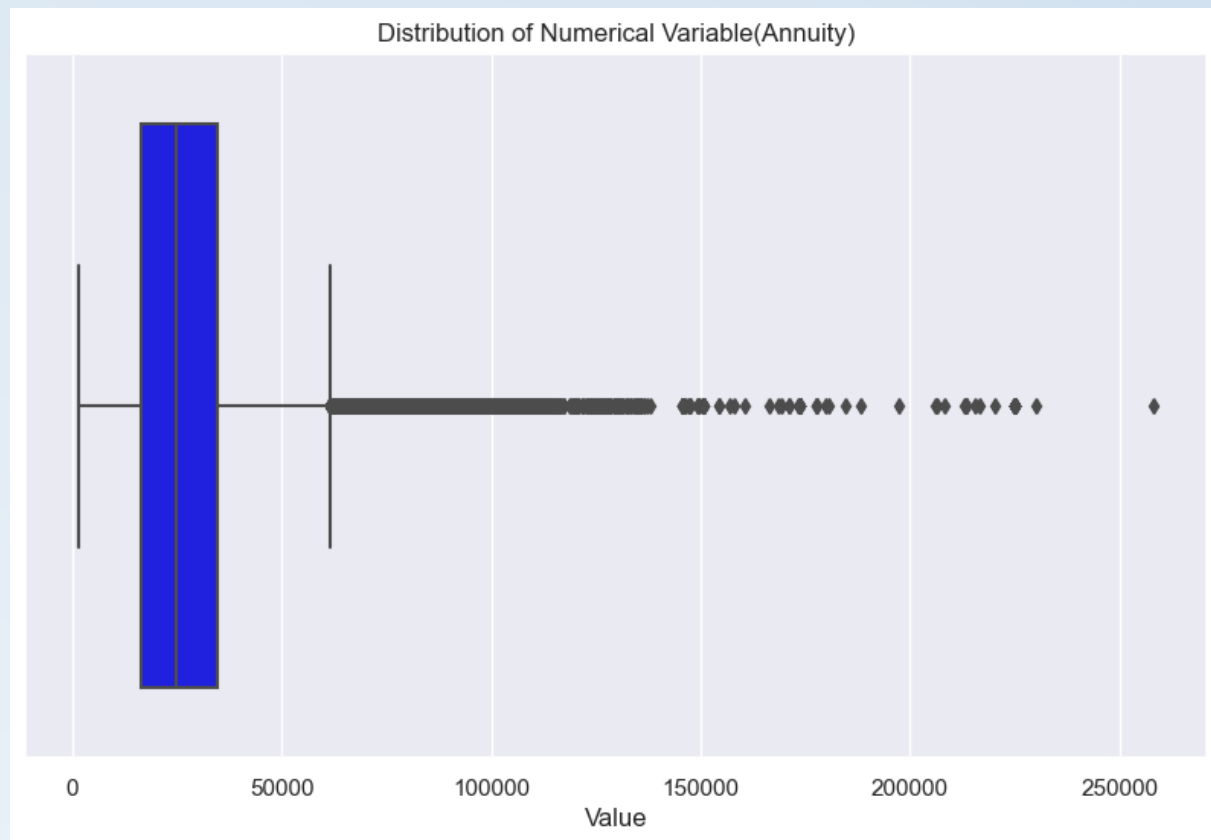


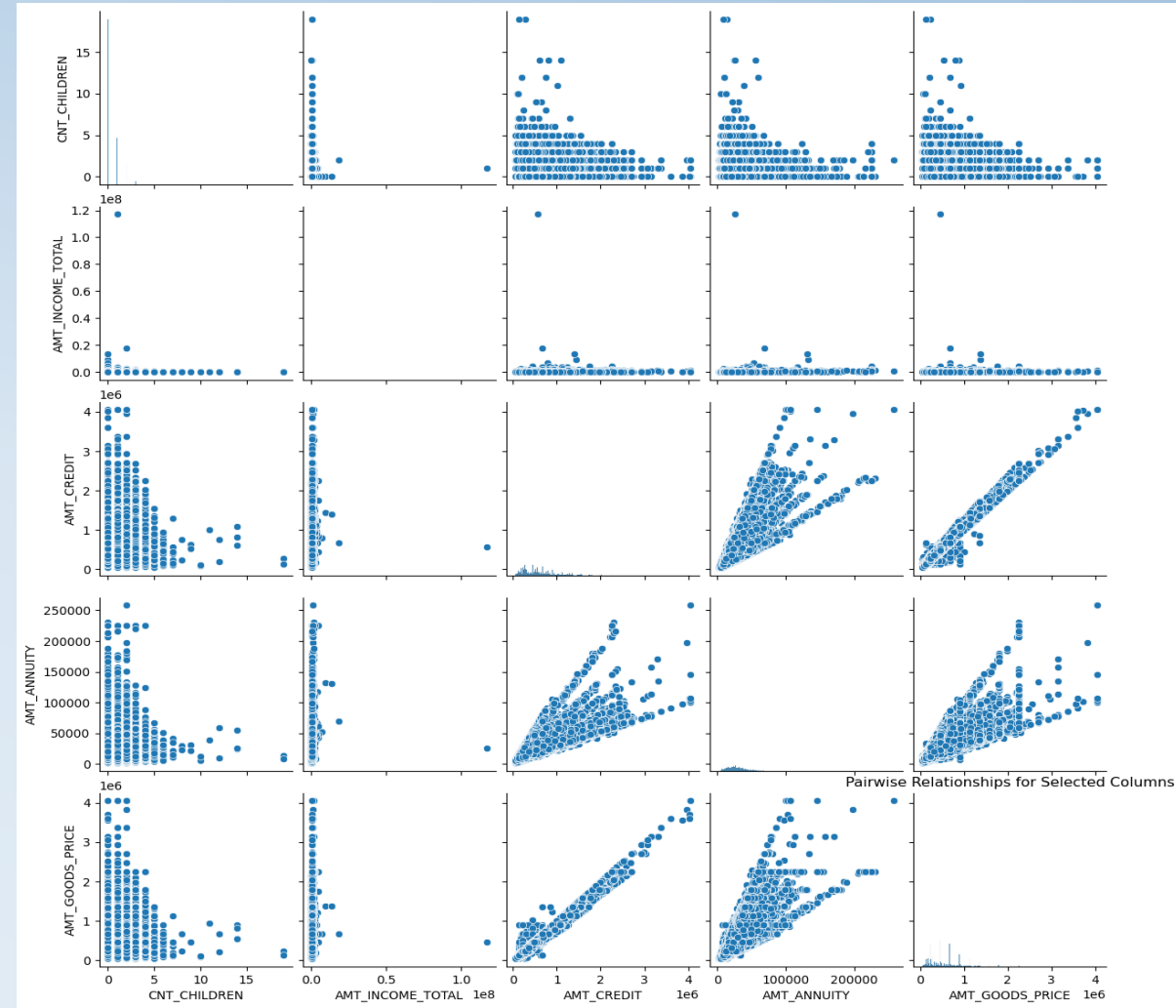
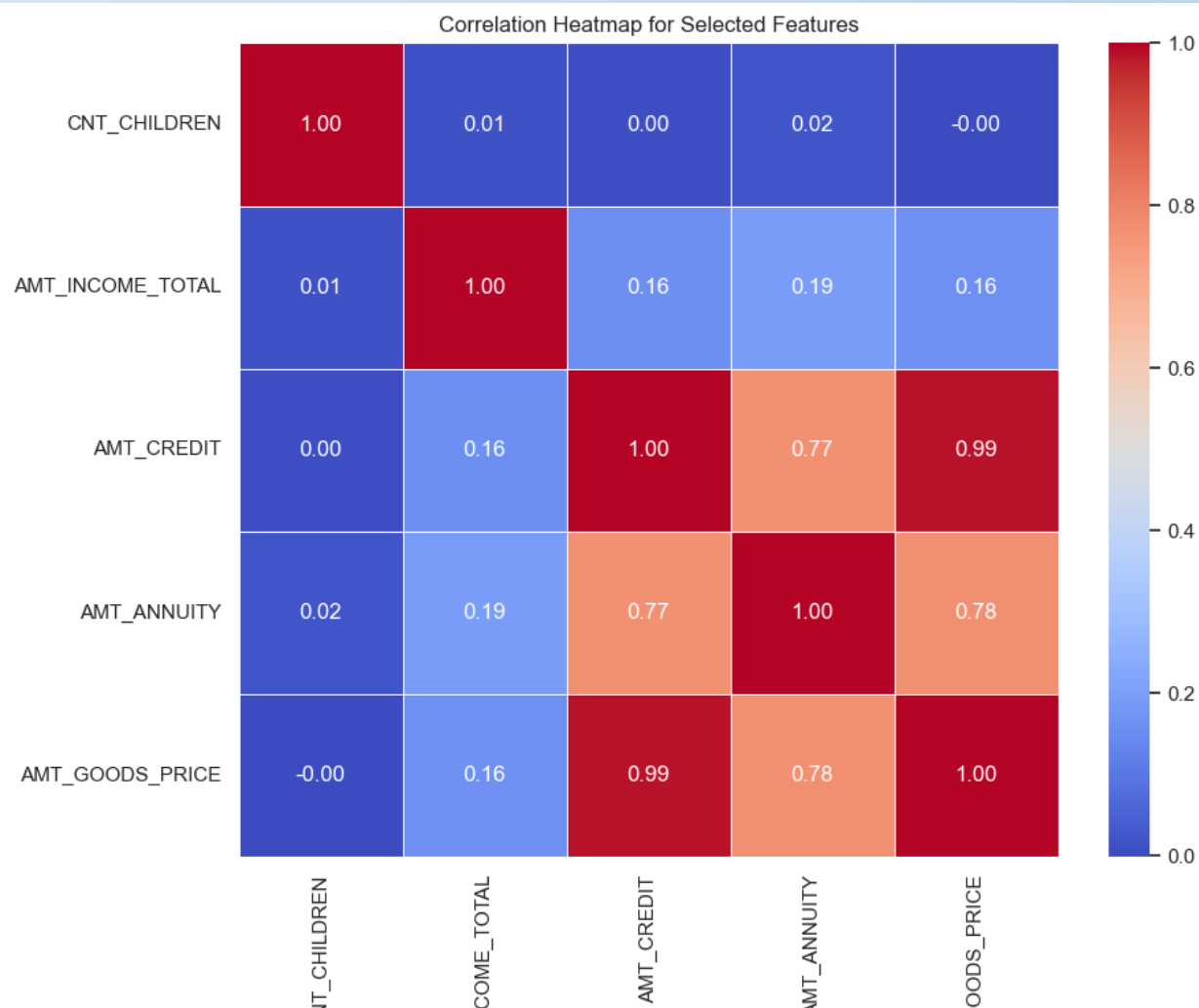
# Understanding the Data

The Numerical Features was analyzed as well. The image on the right hand shows some the numeric features and how important and relevant they are to building the model by using the “importances\_sorted” feature on the dataset and plotting the bar .



# Insight into some selected Numerical Features





Correlation amongst some randomly selected features

# Machine Learning and Model Integration



# Machine Learning

## Probability Prediction

- Model : Random Forest Classifier
- Performance metrics: Accuracy score and confusion matrix.
- Selecting the **top 15** important features and re-run the model. Performance metrics check.
- Assign probability and risk levels.

	SK_ID_CURR	Probability	Risk Level
0	215987	0.16	Very Low Risk
1	215990	0.10	Very Low Risk
2	215995	0.04	Very Low Risk
3	216004	0.25	Low Risk
4	216006	0.09	Very Low Risk
...	...	...	...
27030	331734	0.05	Very Low Risk
27031	331736	0.11	Very Low Risk
27032	331737	0.10	Very Low Risk
27033	331739	0.07	Very Low Risk
27034	331744	0.10	Very Low Risk

[27035 rows x 3 columns]

The first performance metrics:

Confusion Matrix:

```
[[47266    0]
 [ 1577   2277]]
```

Accuracy: 0.9691510172143974

After Hyperparameter tuning and Choosing top contributing features in model :

Confusion Matrix:

```
[[47266    0]
 [ 1518   2336]]
```

Accuracy: 0.9703051643192488

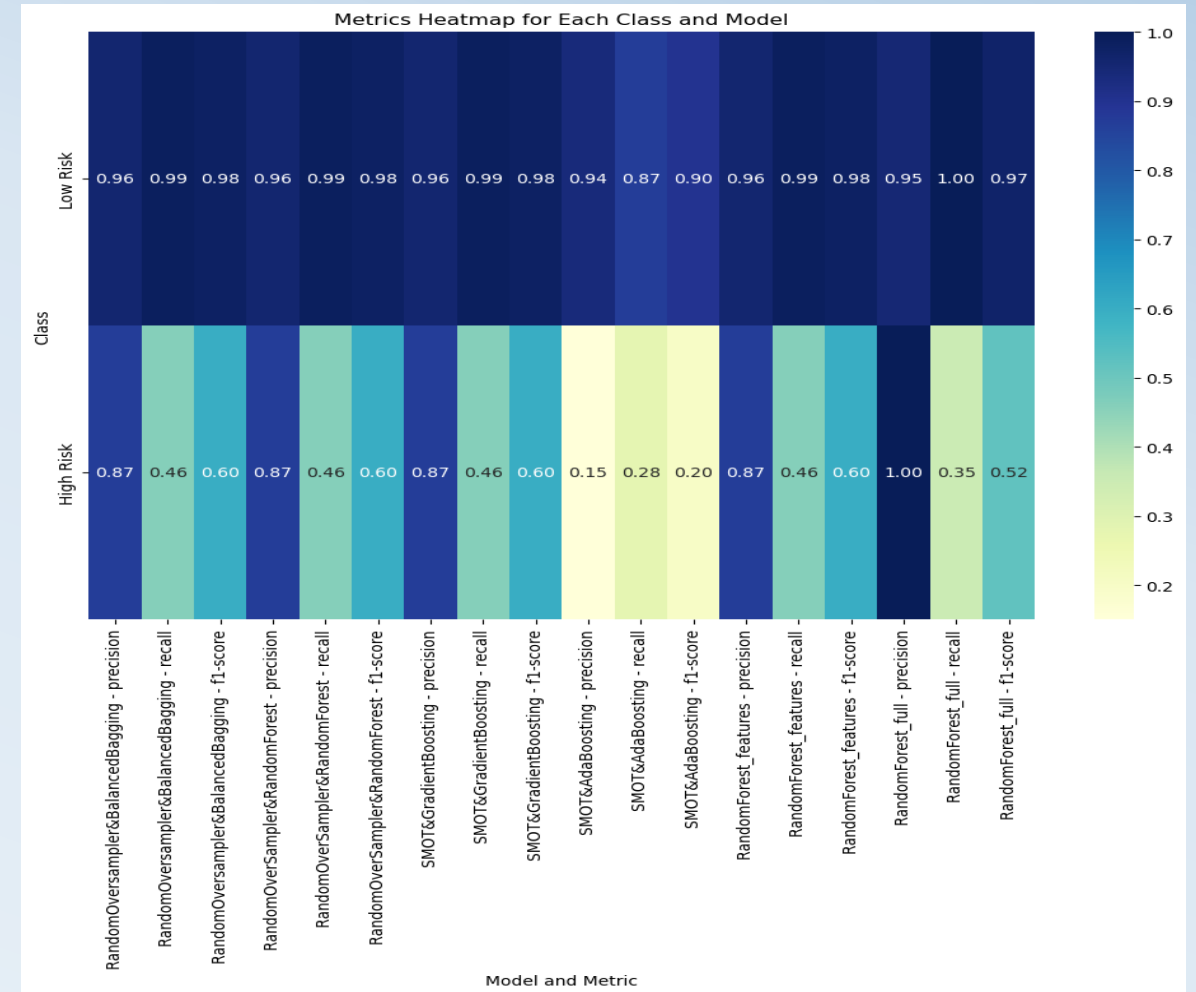


# Resampling and Ensemble Modeling

## Model Selection

- Our data has a high level of skewness.
- While trying to choose a reasonable model, resampling and ensembling methods used:
  - Random Over Sampler(ROS)- Random forest Classifier
  - ROS- Balanced Bagging Classifier
  - Synthetic Minority over-sampling Technique (SMOTE) - Gradient Boosting
  - SMOTE - AdaBoosting
- Models' predictions were made, and imbalanced classification reports were gathered.

Based on the classification reports a concise heatmap is generated as below:



# Final Model

- Here's the breakdown of the confusion matrix:
  - True Positives (TP): 2,980
  - True Negatives (TN): 49,676
  - False Positives (FP): 1,207
  - False Negatives (FN): 0
- Accuracy: Accuracy is the proportion of correctly classified instances out of the total number of instances. It is calculated as  $(TP + TN) / (TP + TN + FP + FN)$ . In this case, the accuracy is 0.9775, or 97.75%, indicating that the model correctly classified 97.75% of the instances.
- Class 0 (the negative class) has high precision, recall, and F1-score, indicating that the model performs well in identifying negative cases.
- Class 1 (the positive class) has perfect precision but lower recall, resulting in a lower F1-score. This suggests that while the model identifies positive cases with high precision, it misses some positive cases.
- The weighted average F1-score is 0.98, indicating strong overall performance across both classes.
- The macro average F1-score is 0.91, indicating a slightly lower performance when considering class imbalance.

Confusion Matrix:

```
[[49676    0]
 [ 1207   2980]]
```

Accuracy: 0.9775912964372575

Classification Report:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	49676
1	1.00	0.71	0.83	4187
accuracy			0.98	53863
macro avg	0.99	0.86	0.91	53863
weighted avg	0.98	0.98	0.98	53863

# Machine Learning:

## Amount Prediction

- The insight into the data shows some imbalances therefore the model selected for prediction should be able to improve model performance, handle imbalanced datasets, and reduce overfitting.
- Upon further research, we found out that resampling and ensembling are two powerful techniques commonly used in machine learning to improve model performance, handle imbalanced datasets, and reduce overfitting.
- Machine learning algorithms used:
  - Random Forest
  - XG Boost
  - Lasso Regression and
  - Neural Network Regression
  - AdaBoost
- We did this extra step to understand how our model is doing
- Performance metrics is illustrated in the image above

```
Random Forest RMSE: 0.21798580636574888
XGBoost RMSE: 0.2898705546802953
Lasso Regression RMSE: 0.09850914453283427
Neural Network Regression RMSE: 0.24162629190062188
AdaBoost RMSE: 0.23015328606206514
```

# Front End



# Front End

## Create Endpoints for Model Prediction:

- Defined endpoints or routes in our web application that corresponds to the functionalities of our machine learning model, that created a route to accept input data, preprocess it, and pass it to your model for prediction.
- When the model predicts the outcome, return the result as an HTTP response to the client.

## Handle Model Integration:

- Loaded the serialized model into memory when the web server starts up. This ensures that the model is readily available for prediction.
- Integrate the loaded model with the appropriate endpoint handler in your web application.
- When a request is received at the prediction endpoint, preprocess the input data as necessary and pass it to the model for prediction.
- Return the prediction result to the client, either as JSON data or through a web page rendered with the result.

## Deployment:

- Deployed our web application to a web server or cloud platform to make it accessible over the internet.
- Ensured that the server environment has the necessary dependencies installed, including the machine learning libraries required to load and use the model.

## Testing and Monitoring:

- Tested the integration thoroughly to ensure that the model behaves as expected in a web environment.
- Monitored the performance of our web application and model in production to identify and address any issues that arise.



## Final Output

The outcome of this project is summarized in the link below

[https://github.com/callmidrey/Project-4--Group-3/blob/main/templates/pre\\_evaluation.html](https://github.com/callmidrey/Project-4--Group-3/blob/main/templates/pre_evaluation.html)

Live Demo

# Limitations and suggestions

Live Demo

## Limitations

- Extremely large dataset: Top 99,999 entries are selected to run the model with available resources.
- Imbalanced Classes: Imbalance in the distribution of classes can skew model performance and affect the accuracy of predictions, especially in classification tasks.
- Missing Values: Presence of missing data points can introduce bias and reduce the reliability of statistical analyses and predictive models if not handled properly.
- Limited Feature Representation: The dataset may lack certain important features or variables that could provide deeper insights into the phenomenon under study, limiting the scope of analysis and interpretation.