# CREDIT RISK ASSESSMENT USING MACHINE LEARNING MODELS

GROUP:05

Team Member:
Labdhi Zatakia
Ishita Vaghela
Sam Leslie
Banshi Keshwala

# INTRODUCTION & MOTIVATION:

## Problem:

- Risk of customer defaults can cause financial loss.

## Importance:

- Credit approval decisions impact profitability.
- Predicting risk helps reduce losses.

## Value:

- Supports data-driven credit decisions.
- Efficient and reusable machine learning solution.

## Our Approach:

- Use customer data (income, assets, past behavior).
- Classify customers as **Good / Medium / Bad**.
- Provide **probability of default** for uncertain cases.

# DATA CLEANING/ INITIAL ANALYSIS:

**Application Data:**

- Includes: gender, car/house ownership, children, income, education, family status, age, employment days, phone/email flags, occupation
- **Missing values:** OCCUPATION_TYPE


**Merging:**

- We cleaned the data by removing rows with missing occupation information, and then we merged both datasets using the customer ID.

**Credit Record Data:**

Fields: ID, MONTHS_BALANCE (0 to -60), STATUS

- **No missing values**

# DATA TRANSFORMATION:

| | | |
|---|---|---|
| **Null Handling** | Removed rows | missing occupation types |
| **Feature Engineering** | Converted | 'DAYS_BIRTH' to 'Age &and 'DAYS_EMPLOYED' to 'MONTHS_EMPLOYED' |
| **Binning** | Categorized continuous variables | (Income, Age) |
| **Target Encoding** | Mapped : **Credit Status** | 1.GOOD 2.NEUTRAL 3.BAD |

# EXPLORATORY DATA ANALYSIS (EDA):

Examined the dataset to understand patterns and relationships:

- Outlier detection: Summarize numeric data for analysis
- Examining distribution of all numeric variables including age, income, and employment duration
- Correlation heatmap to see which variables move together

Key findings:

- No obvious outliers
- Data distribution presents as valid for features – imbalance in months_balance
- Family count and child count correlates strongly

# OUTLIER DETECTION:

- No obviously incorrect difference between mean/median for numeric data EXCEPT for months_balance

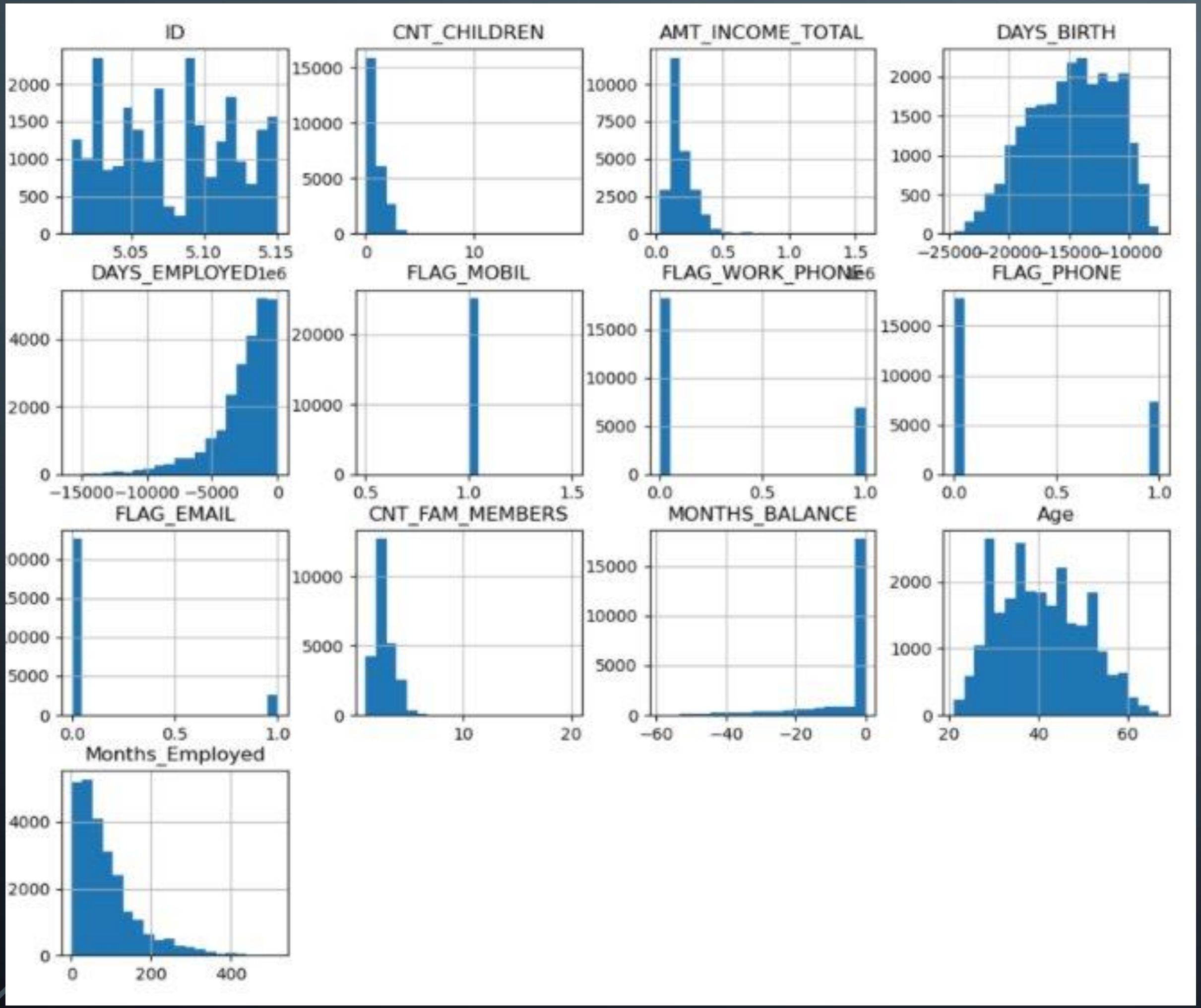
```
[38]: #outlier detection
credit_cleaned_df.select_dtypes(include='number').agg(['sum','mean','median','min','max']).round(4)
```

[38]:

| | ID | CNT_CHILDREN | AMT_INCOME_TOTAL | DAYS_BIRTH | DAYS_EMPLOYED | FLAG_MOBIL | FLAG_WORK_PHONE | FLAG_PHONE | FLAG_EMAI |
|---|---|---|---|---|---|---|---|---|---|
| **sum** | 1.276515e+11 | 12877.0000 | 4.896954e+09 | -3.718333e+08 | -6.597526e+07 | 25134.0 | 6882.0000 | 7359.0000 | 2530.000 |
| **mean** | 5.078838e+06 | 0.5123 | 1.948339e+05 | -1.479404e+04 | -2.624941e+03 | 1.0 | 0.2738 | 0.2928 | 0.100 |
| **median** | 5.079004e+06 | 0.0000 | 1.800000e+05 | -1.454700e+04 | -1.942000e+03 | 1.0 | 0.0000 | 0.0000 | 0.000 |
| **min** | 5.008806e+06 | 0.0000 | 2.700000e+04 | -2.461100e+04 | -1.571300e+04 | 1.0 | 0.0000 | 0.0000 | 0.000 |
| **max** | 5.150487e+06 | 19.0000 | 1.575000e+06 | -7.489000e+03 | -1.700000e+01 | 1.0 | 1.0000 | 1.0000 | 1.000 |

| | MONTHS_BALANCE |
|---|---|
| **sum** | |
| **mean** | -143982.0000 |
| **median** | -5.7286 |
| **min** | 0.0000 |
| **max** | -59.0000 |
| | 0.0000 |

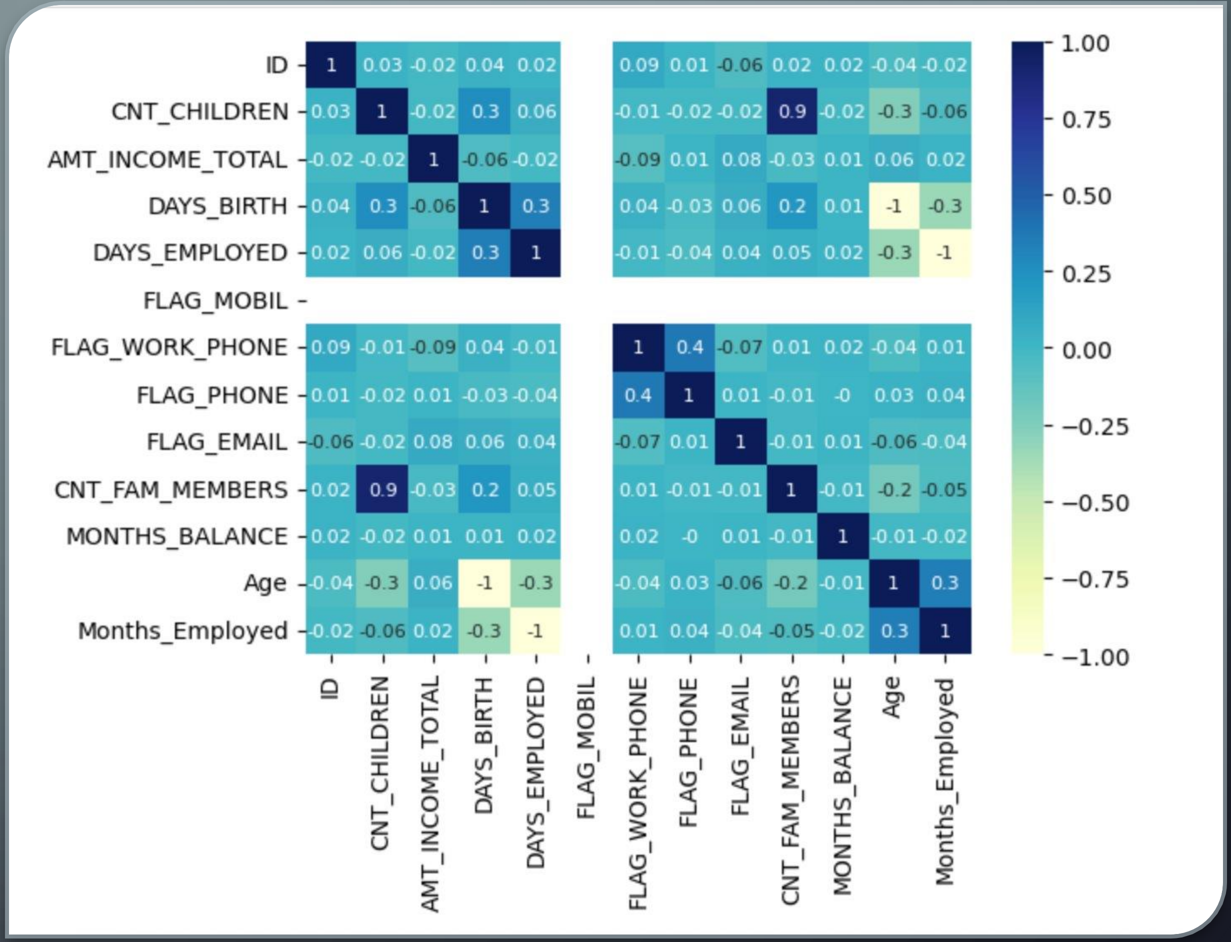
# DATA DISTRIBUTION:




- Outliers in children & family count, and income but typical dispersion.
- Months balance (related to STATUS, our response variable) is skewed left – indicating an imbalance in label data.

# CORRELATION HEATMAP

- Count of Children and Family highly correlated at 0.9
- Multiple cases of around 0.2-0.4 level correlation – indicating a possible lack of independence in the data



# MACHINE LEARNING MODELS:

**Naive Bayes**

- Used as a simple baseline classifier
- Fast and efficient for initial testing
- Helps compare performance against more complex models

**Decision Tree Classifier**

- Captures non-linear relationships in the data
- Easy to interpret and visualize
- Useful for understanding feature importance

**Random Forest Classifier**

- Ensemble of multiple decision trees
- More stable and accurate than a single tree
- Achieved the highest performance in our results

# NAÏVE BAYES MODEL

Process:

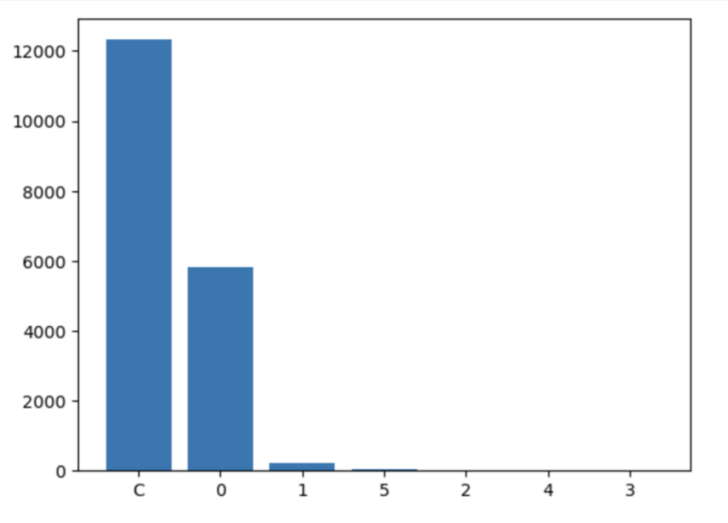
- Creating labels by binning STATUS column into 3 classes
- Binning Age, Months Employed and Income features
- Encoding categorical variables with dummies
- Selecting predictors
- Split data into training and testing groups, run model.
- Review results using: Accuracy score, Confusion matrix, Classification report

Barriers, experimentation and conclusion:

- First results extremely weak due to strong class imbalance (mostly "good" labels)
- Used SMOTE to create synthetic data, retrained model
- Chose less predictors, reduced binning size
- Not the strongest model due to class imbalance, unusual distributions and non-independant data

# LABELS AND BINNING, FEATURE SELECTION

Imbalanced Labels:




```
#creating labels
labels = {
     'C': 'good',
     '0': 'medium',
     '1': 'medium',
     '2': 'bad',
     '3': 'bad',
     '4': 'bad',
     '5': 'bad'}

credit_cleaned_df["
print(credit_cleane
```

```
LABELS
good      12319
medium     6038
bad          78
Name: count, dtype: int64
```

5 BINS :

```
emp_bins = [0, 12, 36, 60, 120, 500]      # in months
emp_labels = ['<1yr', '1-3yr', '3-5yr', '5-10yr', '10+yr']
```

```
#age and months bins
age_bins = [0, 25, 35, 50, 65, 120]
age_labels = ['18-25', '26-35', '36-50', '51-65', '65+']
```

```
inc_bins = [0, 20000, 40000, 60000, 100000, 200000, 9999999]
inc_labels = ['<20k', '20-40k', '40-60k', '60-100k', '100-200k', '200k+']
```

A lot of features for the first trial

```
cat_predictors = ["CODE_GENDER","FLAG_OWN_CAR","FLAG_OWN_REALTY","CNT_CHILDREN","NAME_INCOME_TYPE","NAME_EDUCATION_TYPE",
                  "NAME_FAMILY_STATUS","NAME_HOUSING_TYPE","OCCUPATION_TYPE","AGE_BIN","EMP_BIN","INCOME_BIN"]
num_predictors = ["FLAG_WORK_PHONE","FLAG_PHONE","FLAG_EMAIL"]
```

# INITIAL RESULTS

Extremely poor

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| bad | 0.00 | 0.00 | 0.00 | 19 |
| good | 0.67 | 1.00 | 0.80 | 3685 |
| medium | 0.53 | 0.00 | 0.01 | 1827 |
| accuracy | | | 0.67 | 5531 |
| macro avg | 0.40 | 0.33 | 0.27 | 5531 |
| weighted avg | 0.62 | 0.67 | 0.54 | 5531 |

```
[[   0   19    0]
 [   1 3676    8]
 [   0 1818    9]]

0.6662448020249503
```

# RETRIALS:

- Reduced binning size to 3

```
emp_bins = [0, 36, 120, 500]      # in months
emp_labels = ['<3yrs', '3-10yrs', '10+yrs']
```

```
#age and months bins
age_bins = [0, 35, 60, 120]
age_labels = ['18-35', '36-60','60+']
```

```
inc_bins = [0, 40000,100000,9999999]
inc_labels = ['<40k', '40-100k','100k+']
```

Used SMOTE to generate synthetic data to hopefully assist in the imbalance

```
cat_predictors = ["FLAG_OWN_CAR","FLAG_OWN_REALTY","CNT_CHILDREN","NAME_EDUCATION_TYPE",
                  "AGE_BIN","EMP_BIN","INCOME_BIN"]
outcome = "LABELS"
```

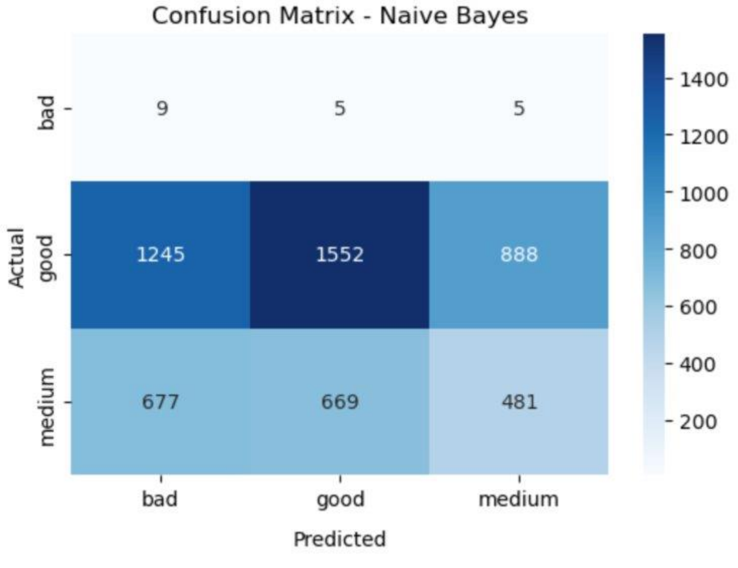
Chose only 7 features

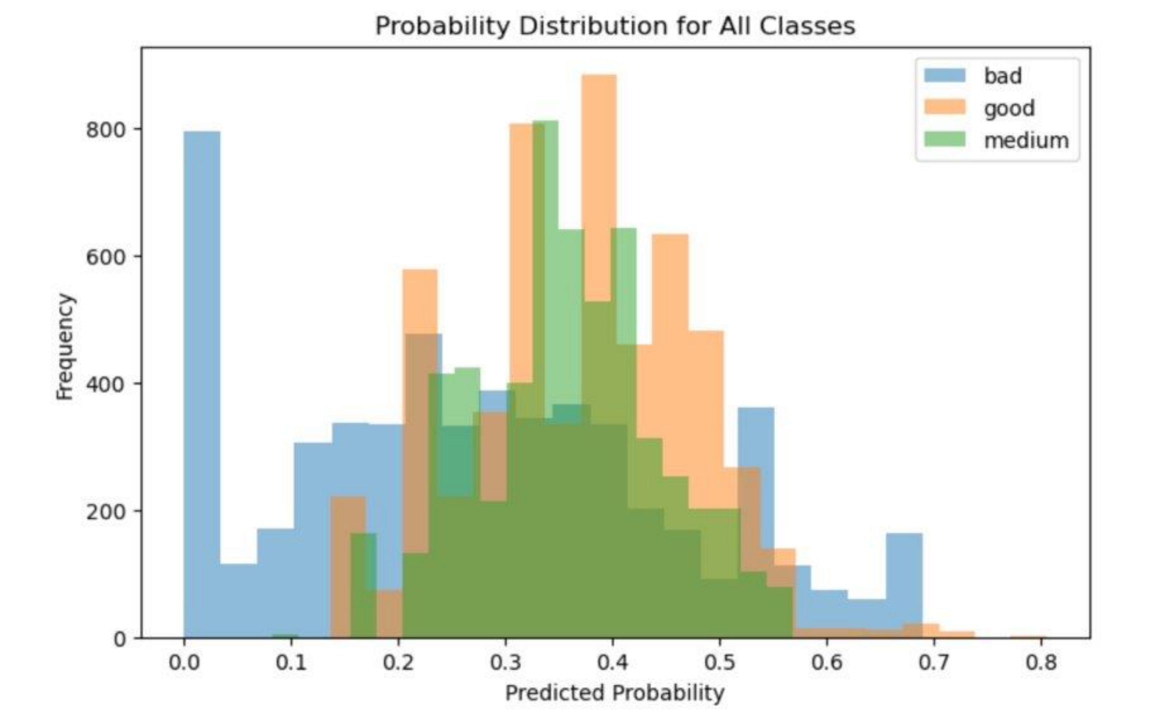
# AFTER RETRIALS:

Still poor performance

```
0.36883022961489786
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| bad | 0.00 | 0.53 | 0.01 | 19 |
| good | 0.69 | 0.42 | 0.53 | 3685 |
| medium | 0.34 | 0.21 | 0.26 | 1827 |
| accuracy | | | 0.36 | 5531 |
| macro avg | 0.34 | 0.39 | 0.27 | 5531 |
| weighted avg | 0.57 | 0.36 | 0.44 | 5531 |



# DECISION TREE:

Process:

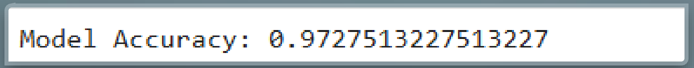
- Checking for missing values in Decision and removing them.
- Spliting the data into train and test and applying it to the decision tree model.
- 2nd attempt of applying the model with labelencoder and objective type columns.
- Checking accuracy and building the confusion matrix.

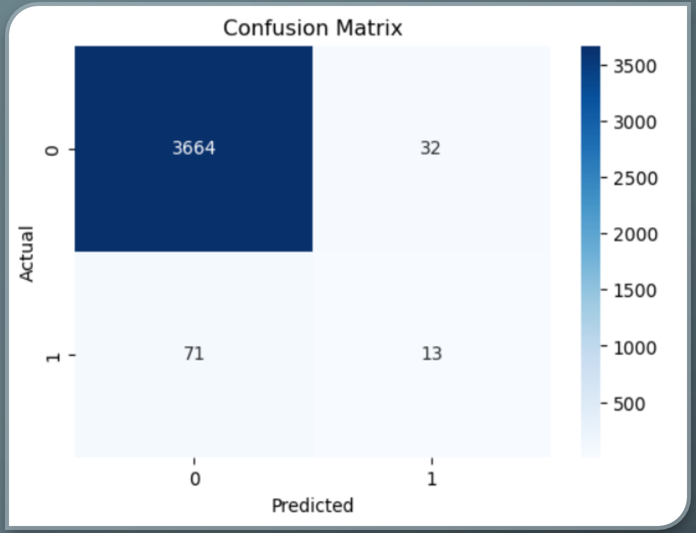
Challenges:

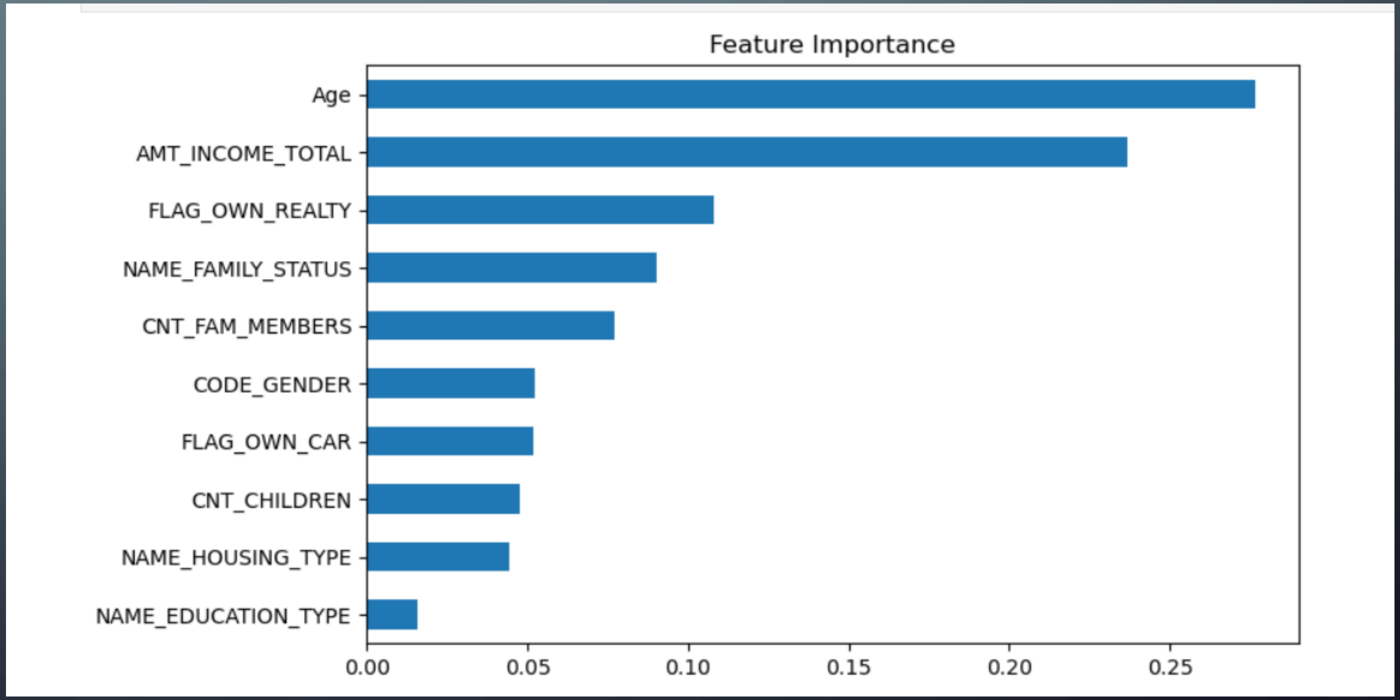
- First attempt: Model running but data type error
- Second attempt: tried different split and test but most optimal was 70-30.

# FINAL:

- Accuracy of model is very good.
- Confusion matrix shows Pridicted vs actual
- Feature which has highest importance: Age
- Feature which has lowest importance: Education type



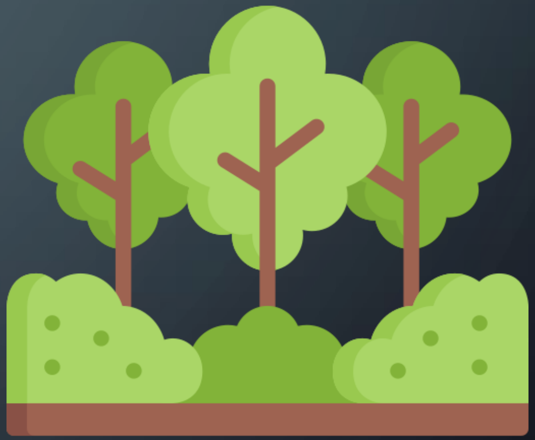



# RANDOM FOREST:

Process and challenges:

- Had to choose between Logistic Reg. And Random forest.
- Had to learn both but random forest works well in all data.
- What is random forest? How to implement it?
- Should it be balanced class or without balanced class?
- Accuracy for both and best of all 3 models.
- Interpretation about the model.



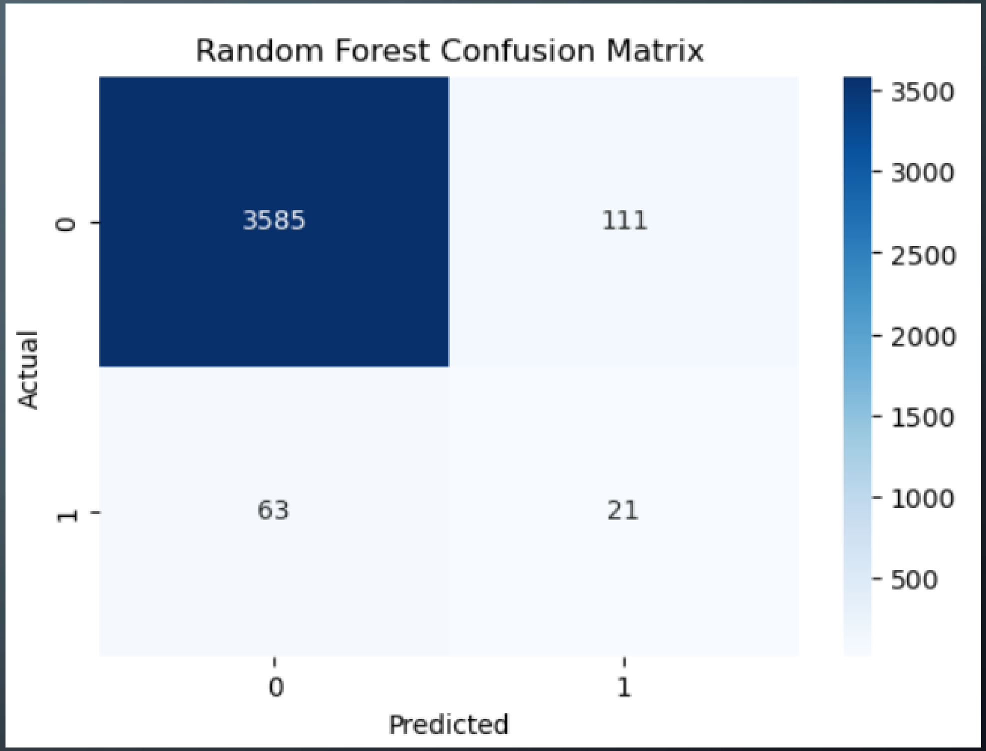
# WITH CLASS WEIGHT AS BALANCED:

- Accuracy is less due to balanced weight
- Confusion matrix shows pridicted vs actual

```
Random Forest Accuracy: 0.953968253968254
```

Random Forest - Classification report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 0.97 | 0.98 | 3696 |
| 1 | 0.16 | 0.25 | 0.19 | 84 |
| accuracy | | | 0.95 | 3780 |
| macro avg | 0.57 | 0.61 | 0.59 | 3780 |
| weighted avg | 0.96 | 0.95 | 0.96 | 3780 |



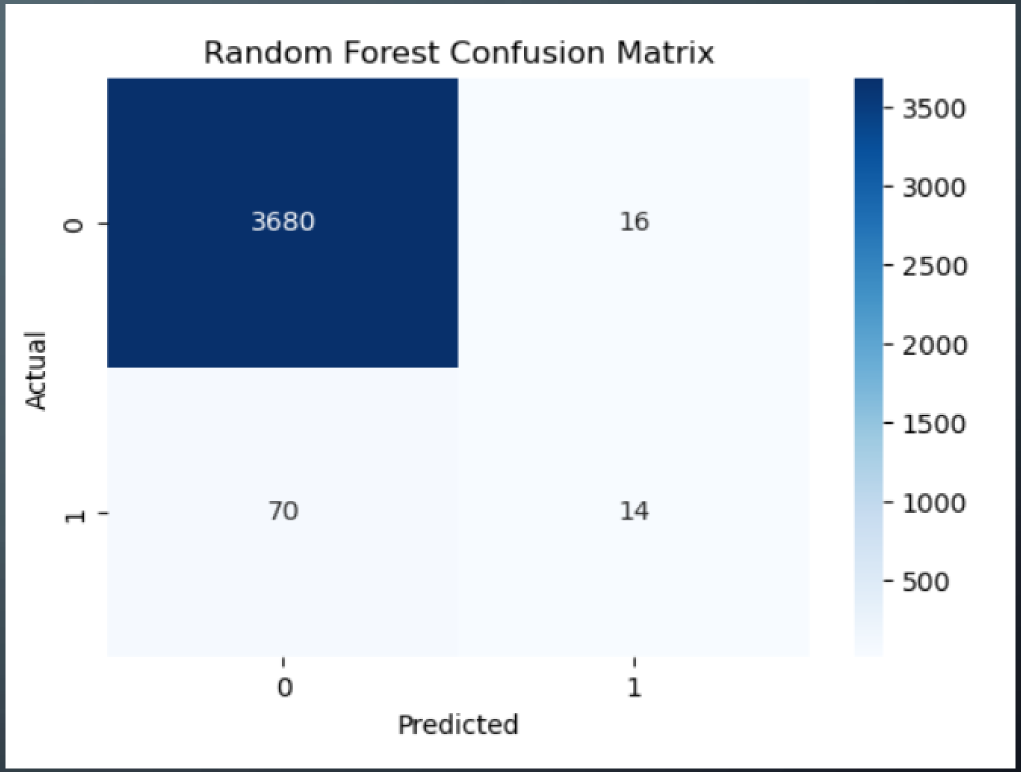
# WITHOUT CLASS WEIGHT AS BALANCED:

- Accuracy Changes a lot
- Confusion matrix gives idea about predicted vs actual

```
Random Forest Accuracy: 0.9772486772486773
```

Random Forest - Classification report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 1.00 | 0.99 | 3696 |
| 1 | 0.47 | 0.17 | 0.25 | 84 |
| accuracy | | | 0.98 | 3780 |
| macro avg | 0.72 | 0.58 | 0.62 | 3780 |
| weighted avg | 0.97 | 0.98 | 0.97 | 3780 |



# CONCLUSION:

- Random Forest achieved the **highest accuracy**
- Handled mixed data types and reduced overfitting
- Naive Bayes struggled with class imbalance
- Decision Tree performed well but less stable
- **Random Forest is the best model** for credit approval prediction

# ROLES:

-Sam leslie: Created confusion matrices

Calculated accuracy, precision, recall, and F1-score

Implemented Naive Bayes

-Labdhi zatakia: Implemented , Random Forest

Compared model performances and identified the best model

-Ishita vaghela: Implemented decision tree,
Designed graphs and visual explanations
Created the PowerPoint slides and presented the findings

-Banshi keshwala : Cleaned the raw dataset
Handled missing values and encoding
Prepared the final dataset for modeling

# REFERENCES:


- https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction
- https://scikit-learn.org/stable/modules/tree.html

*Thank you.*