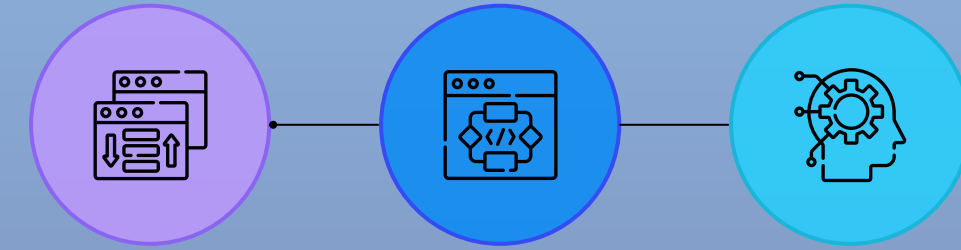


# Credit Score Predictor

Machine Learning Model Applications

Project by:  
Álvaro  
Aurélie  
Ceci  
Tung



**CREDIT SCORE PREDICTOR**  
MACHINE MACHINE LEARNING PROJECT

Machine learning

# Introduction

## Landing Page



### ***Unlock Your Dreams with Ease!***

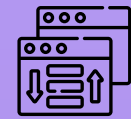
Tired of the endless hoops to jump through for a loan? Whether it's a new home, car, or a little indulgence, we've got you covered. Our cutting-edge algorithm offers a lightning-fast and hassle-free way to see if you're eligible—no more waiting in bank lines or drowning in paperwork.

### ***Discover Your Eligibility Instantly!***

Say goodbye to the red tape and hello to your dreams. Simply click below and fill out the form to check if you qualify for your next big goal. It's time to make those aspirations a reality!

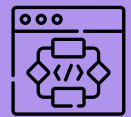
# Project Overview

## Introduction



### Objective

Predict individual credit scores using machine learning models.



### Data-Driven Approach

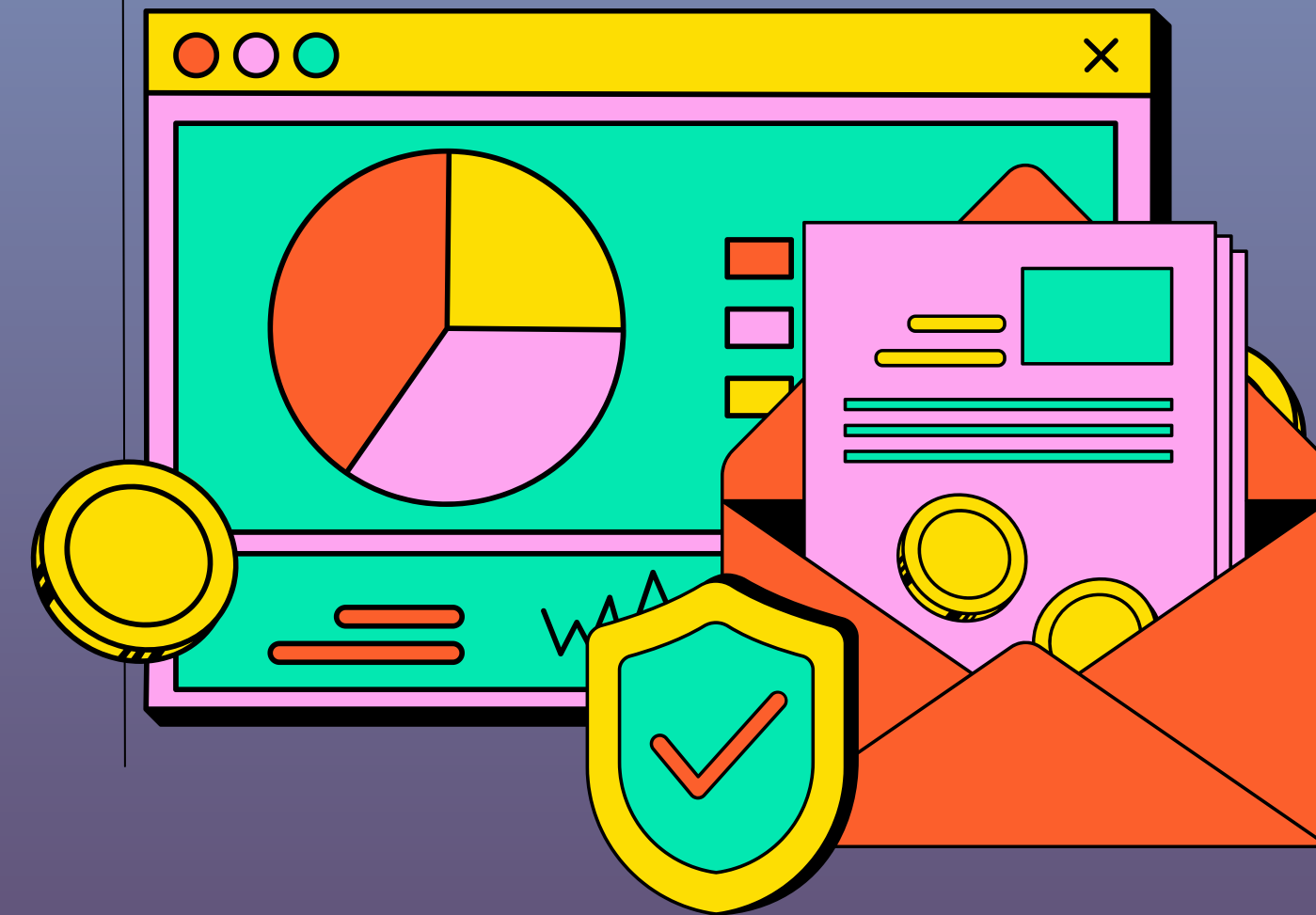
Utilize a selected dataset containing relevant features.

Apply advanced machine learning techniques to analyze and predict outcomes.

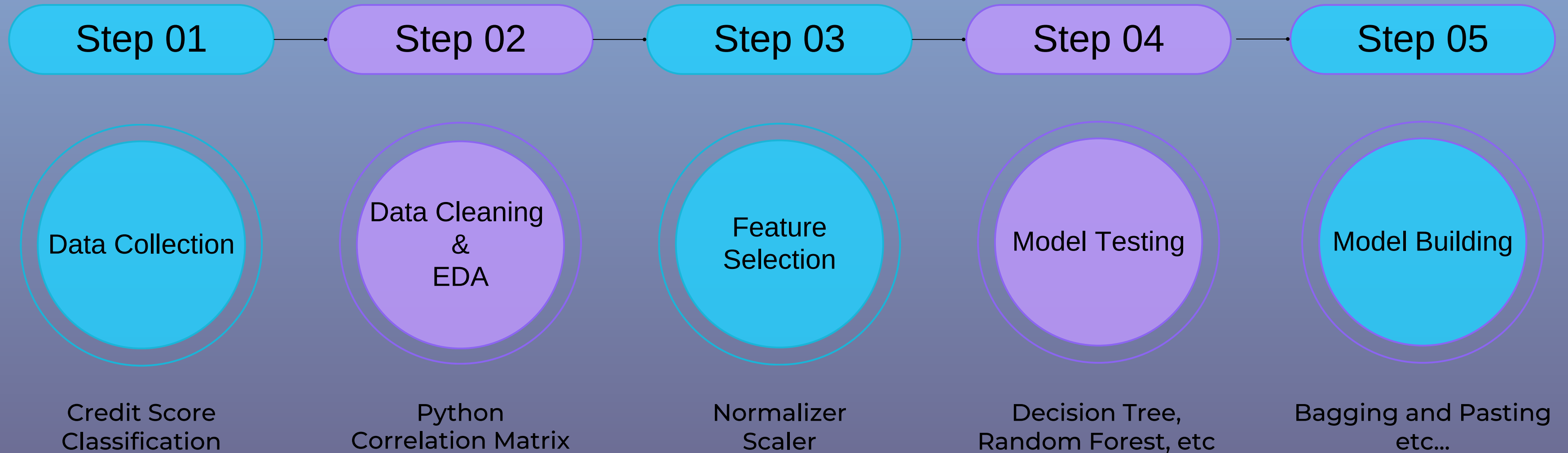


### Outcome

Enhance the accuracy and reliability of credit score predictions.



# Data Analysis



# Data Analysis

## Step 01 - Data Collection

Credit Score Classification:

There are three credit scores that banks and credit card companies use to label their customers:

1. Good
2. Standard
3. Poor

A person with a good credit score will get loans from any bank and financial institution. For the task of Credit Score Classification, we need a labelled dataset with credit scores.





# Data Analysis

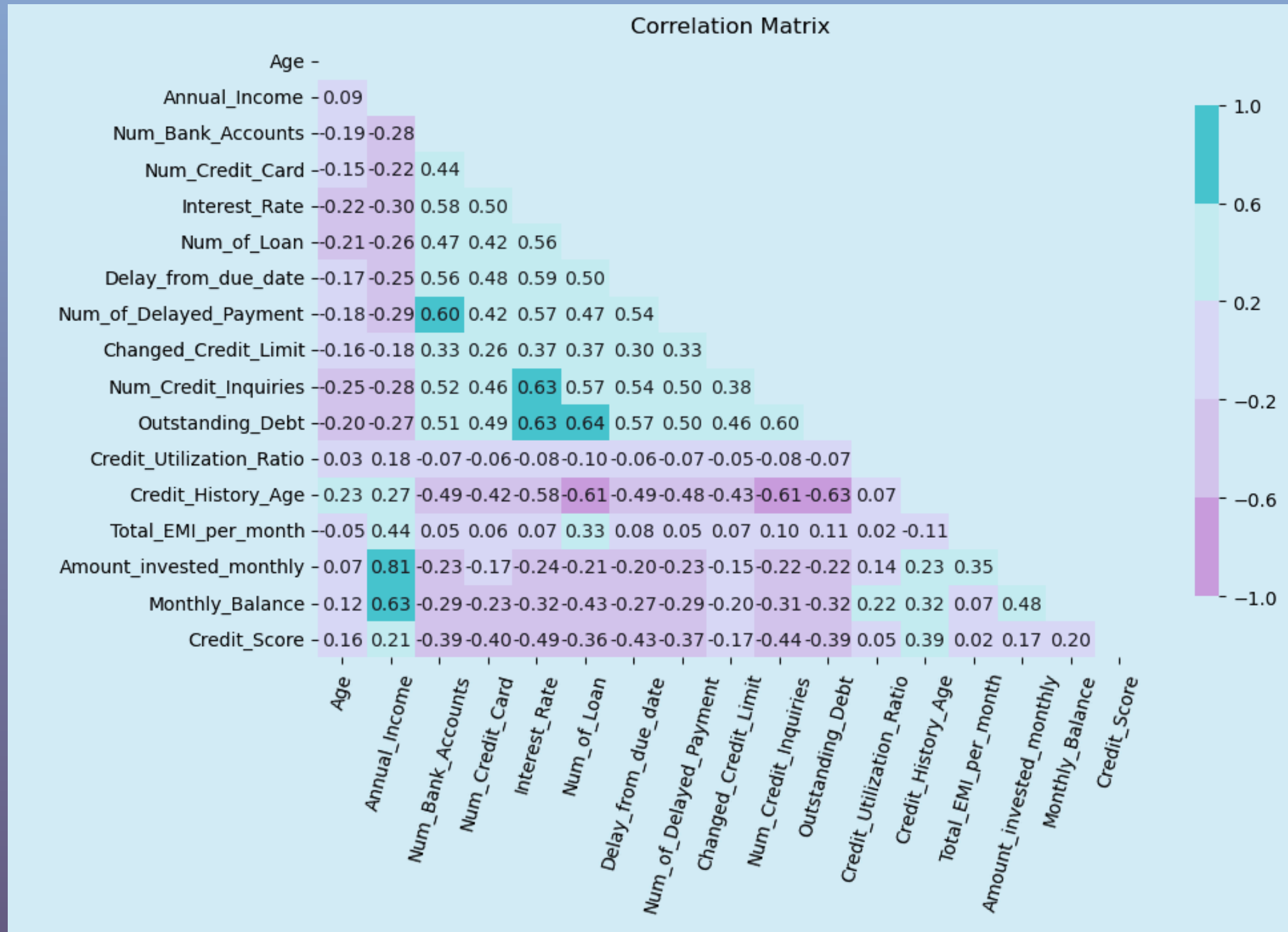
## Step 02 - Data Cleaning & EDA

### Python Cleaning Techniques

- Eliminate Null Values: Swiftly remove missing data to enhance dataset integrity and analytical precision.
- Assess Unique Values: Investigate distinct entries to gain insights and ensure data consistency.
- Refine Column Labels: Standardize and clarify labeling for seamless data manipulation and improved readability.

### Exploratory Data Analysis

- Correlation Matrix



# Data Analysis

## Step 03 - Feature Selection



### **Normalize for Consistency:**

Transform features to a common scale without distorting differences in the ranges of values, ensuring each feature contributes evenly to the model.

### **Scale for Precision:**

Use scaling techniques like Min-Max Scaler or Standard Scaler to enhance model performance by reducing model bias towards features with larger scales.

### **Result:**

Improved data uniformity and model accuracy, ensuring robust and equitable feature influence.

# Data Analysis

## Step 04 - Model Testing

### KNN:

Initial prediction rate of 45% leads to further model testing

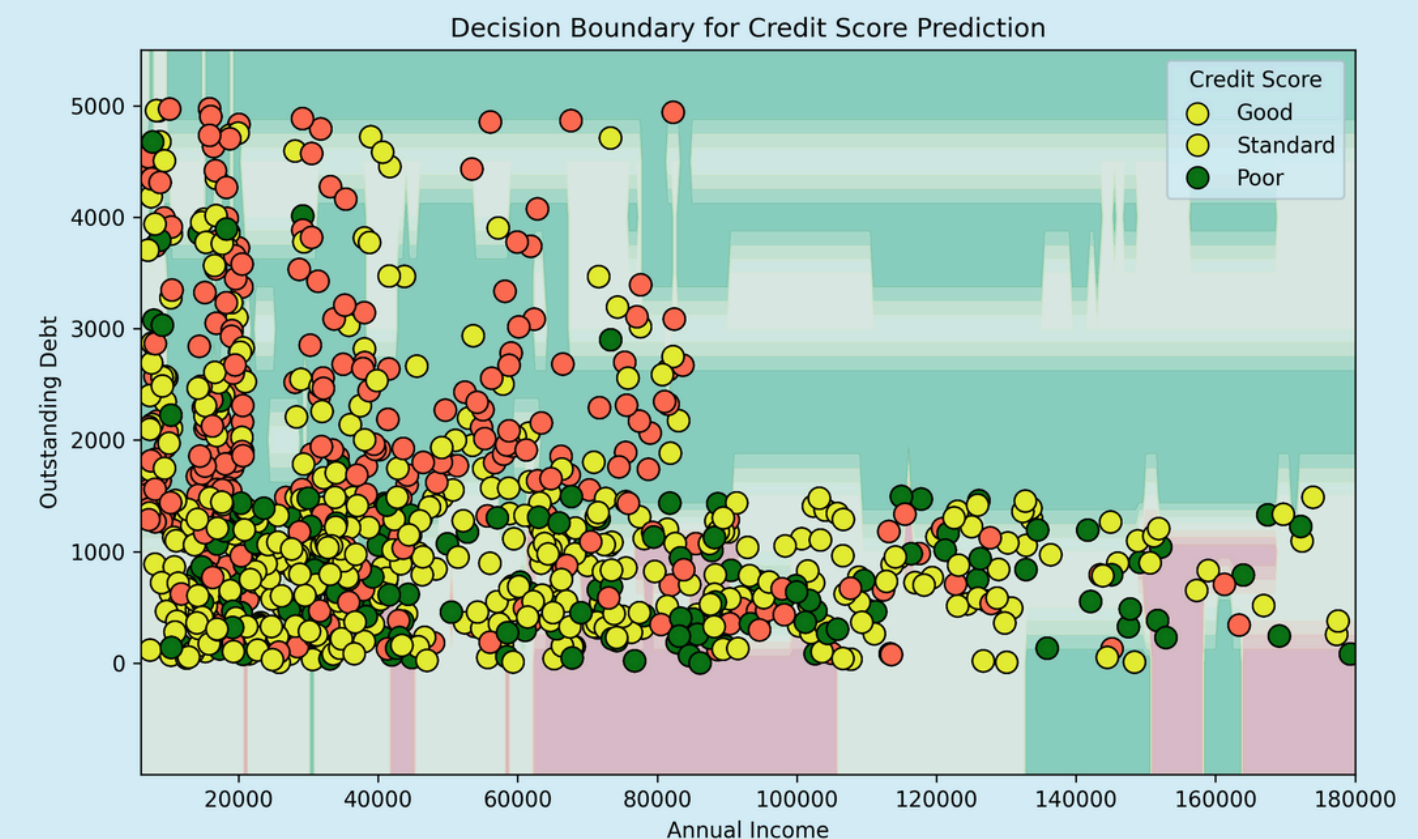
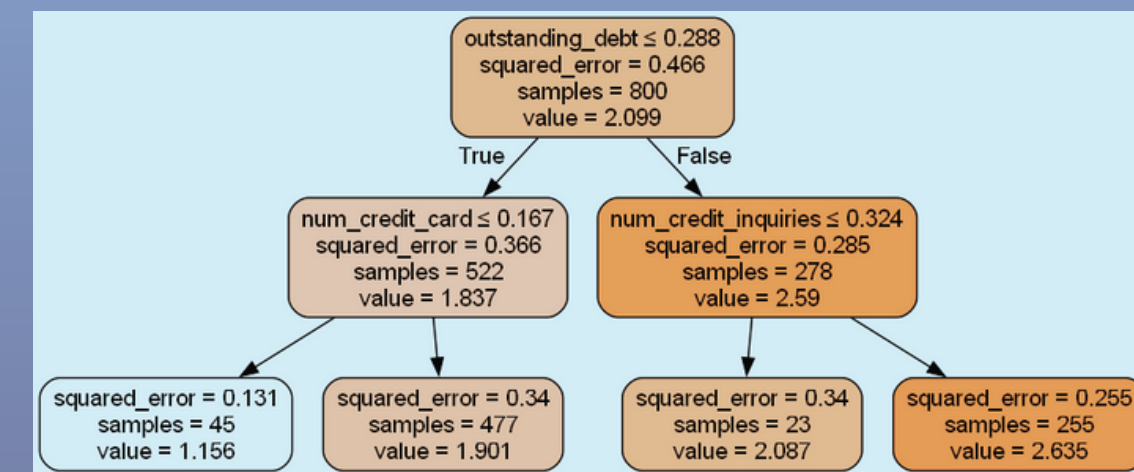
### Decision Tree and Decision Boundary Plot:

Use scaling techniques like Min-Max Scaler or Standard

### Result:

Improved data uniformity and model accuracy, ensuring robust and equitable feature influence.

```
1 print(f"The accuracy of the model is {knn.score(X_test, y_test)*100: .2f}%")  
5] ✓ 0.0s  
· The accuracy of the model is 45.00%
```





# Data Analysis

## Step 04 - Model Testing

### **Linear Regression**

MAE : 0.57

MSE : 0.52

RMSE : 0.72

Accuracy : 0.11

### **Random Forest Regressor**

MAE : 0.42

MSE : 0.33

RMSE : 0.57

Accuracy : 0.44

### **Random Forest Classifier**

MAE : 0.42

MSE : 0.33

RMSE : 0.57

Accuracy : 0.80

### **Gradient Boost Regressor**

MAE : 0.36

MSE : 0.32

RMSE : 0.57

Accuracy : -0.65

### **Gradient Boost Classifier**

MAE : 0.39

MSE : 0.62

RMSE : 0.79

Accuracy : 0.53

# Data Analysis

## Step 05 - Model Building

### Model Selection

We evaluated the following models:

- KNN Classifier  $R^2 \sim 0.9$
- Random Forest Regressor  $R^2 \sim 0.8$

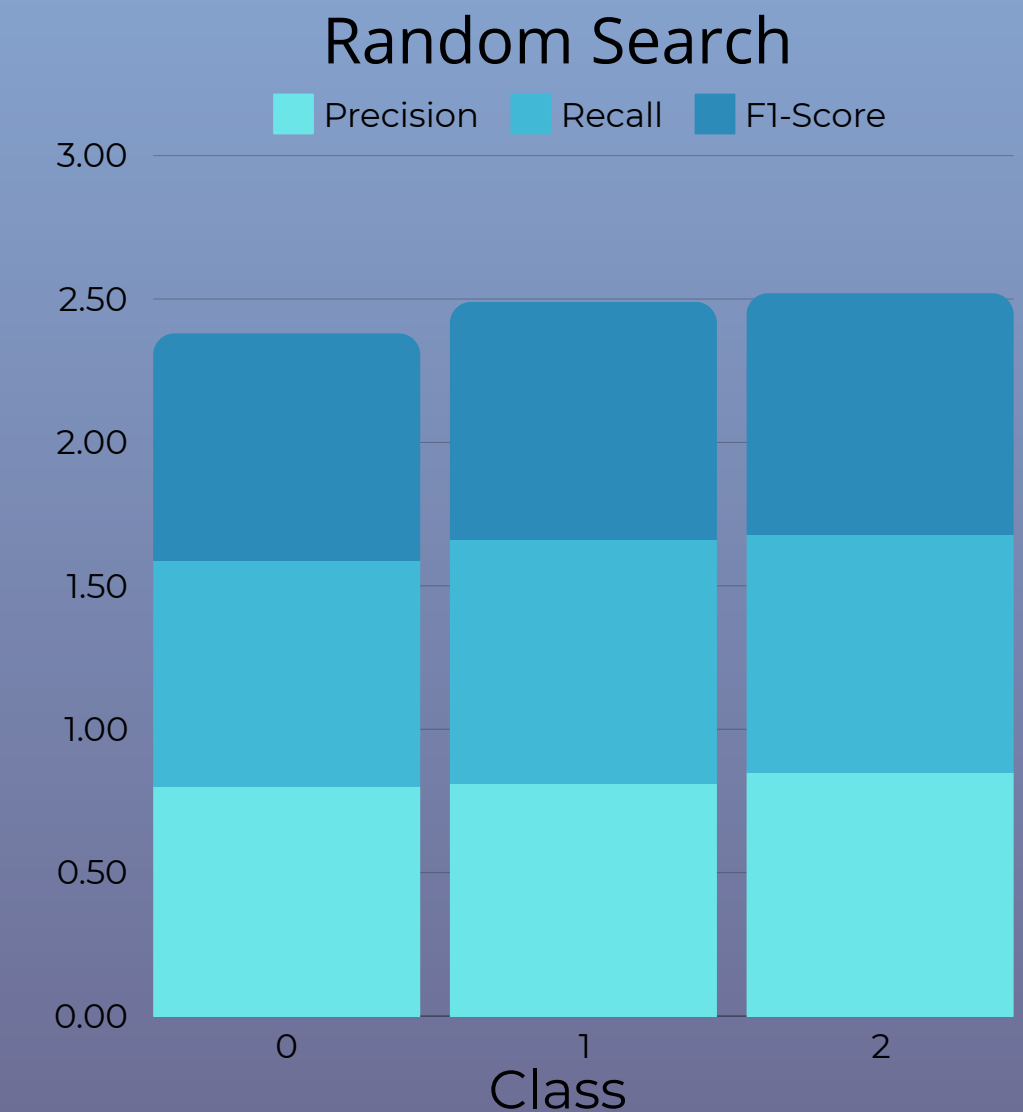
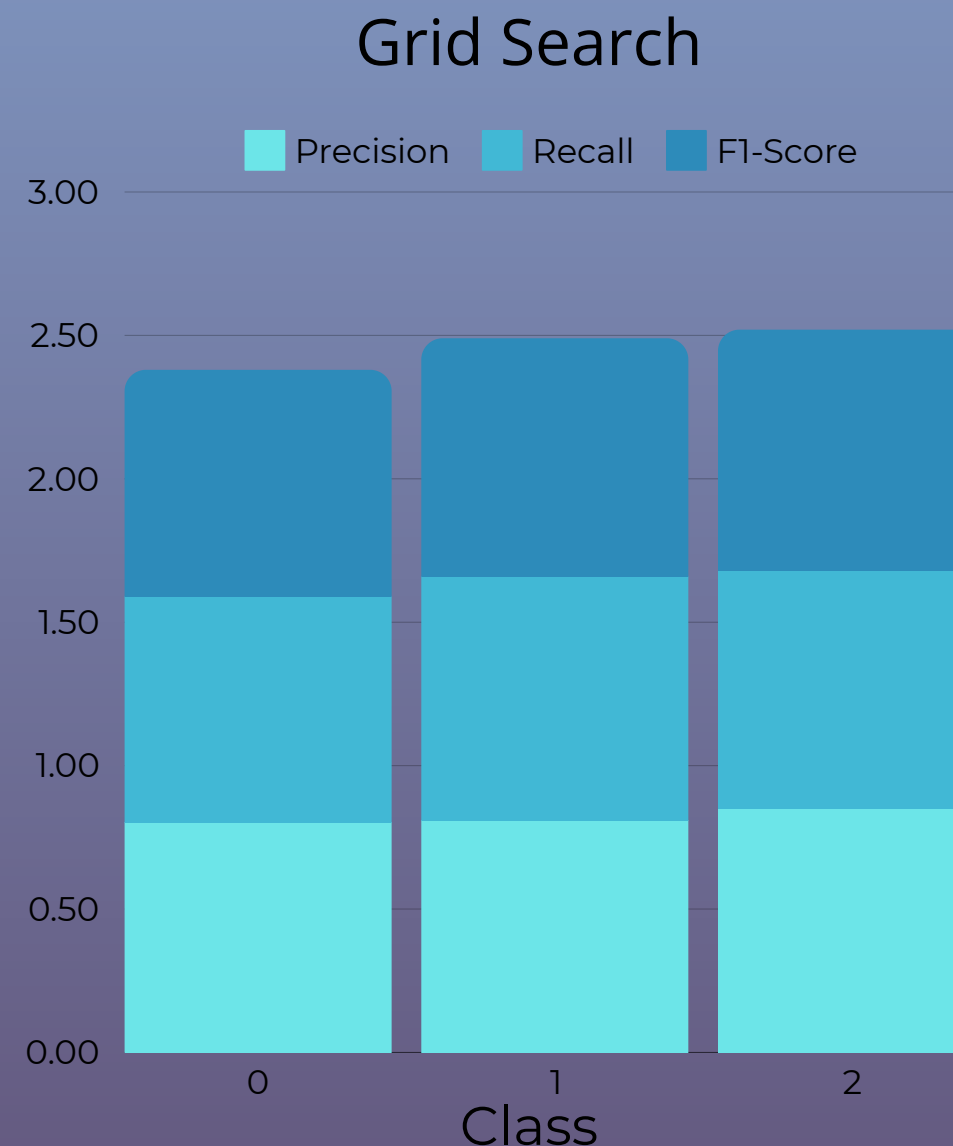
### Hyperparameter Tuning

To optimize the **Random Forest Classifier**, we employed two hyperparameter tuning techniques:

- Grid Search
- Randomized Search.

### Conclusion

The Random Forest Classifier, optimized through Grid Search and Randomized Search, demonstrated the most effective credit score classification model, achieving an accuracy of 83%, highlighting its efficiency.



# Data Analysis

## Step 04 - Model Testing

### KNN Classifier 70% train/30% test

Accuracy: 0.974

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.96	0.97	5322
1	0.97	0.98	0.98	15873
2	0.96	0.98	0.97	8805
accuracy			0.97	30000
macro avg	0.98	0.97	0.97	30000
weighted avg	0.97	0.97	0.97	30000

### KNN Classifier 80% train/20% test

Accuracy: 0.973

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.96	0.97	3527
1	0.98	0.97	0.97	10599
2	0.96	0.98	0.97	5874
accuracy			0.97	20000
macro avg	0.97	0.97	0.97	20000
weighted avg	0.97	0.97	0.97	20000

# Data Analysis

## Step 05 - Model Building

### Model Selection

We evaluated the following models:

- KNN Classifier  $R^2 \sim 0.9$
- Random Forest Regressor  $R^2 \sim 0.8$

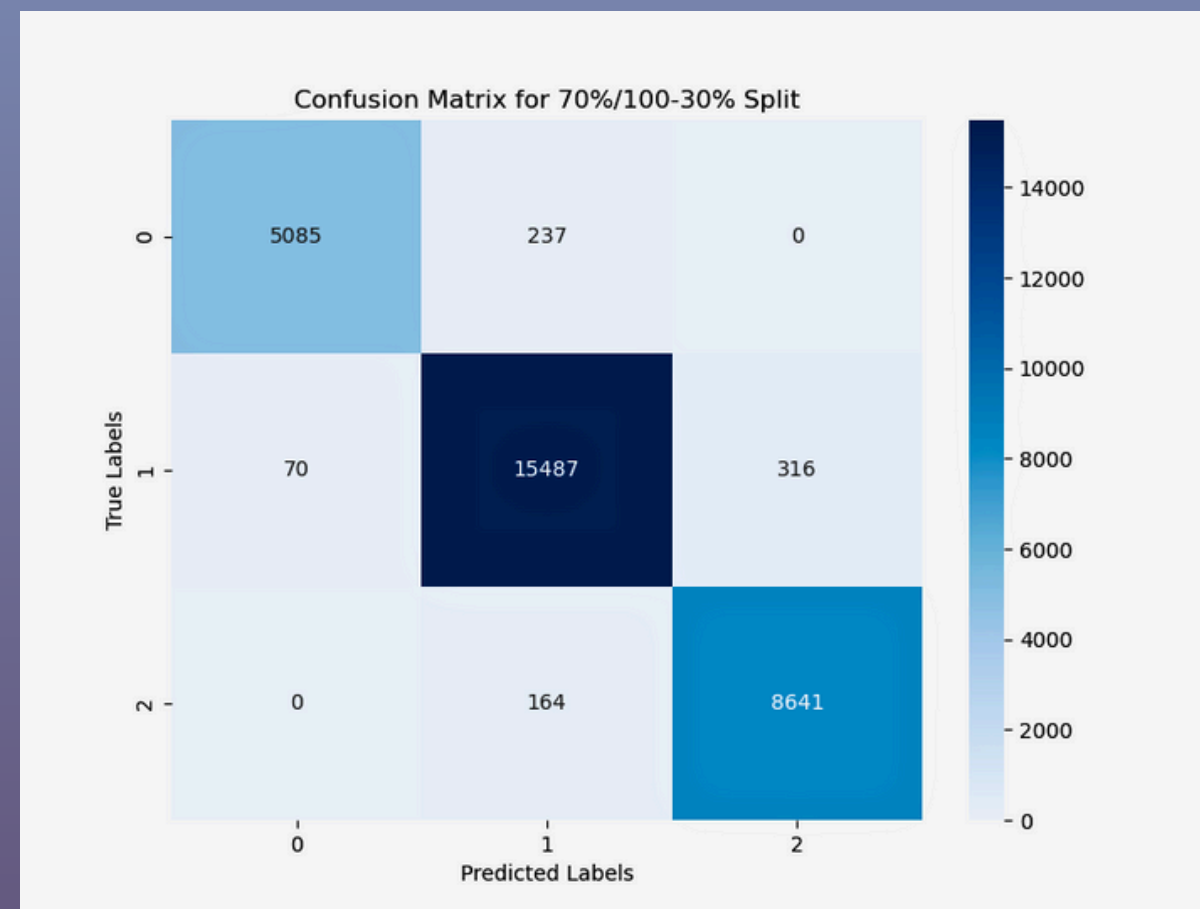
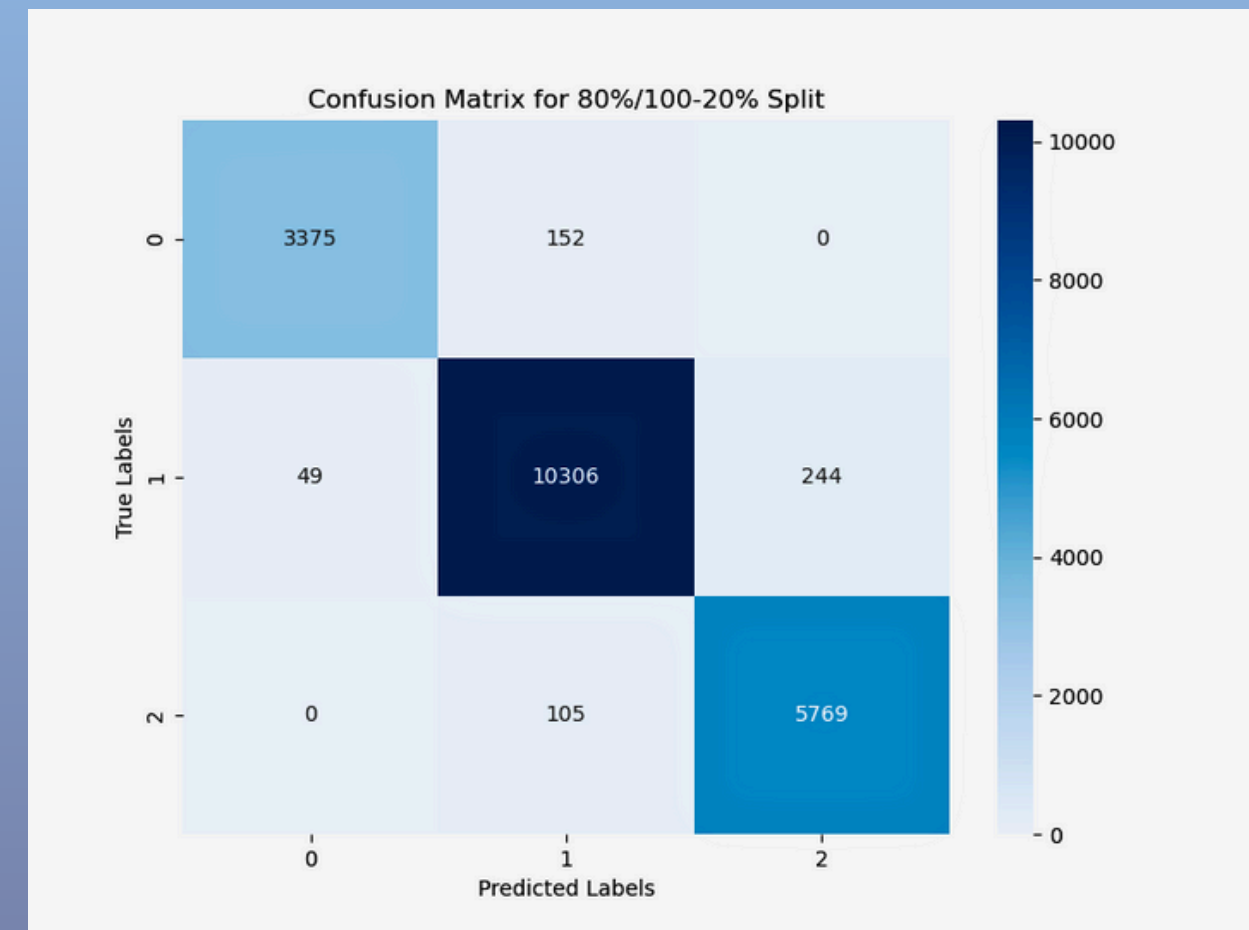
### Hyperparameter Tuning

To optimize the KNN Classifier Model, we employed three hyperparameter tuning techniques:

- Grid Search
- Randomized Search
- Bayesian Test.

### Conclusion

The **KNN Classifier Model**, post encoding of categorical columns, standardization and normalization and hyperparameter implementation through Grid, Randomization and Bayesian Tests shows a very promising accuracy result of 0.97 or **97%**. This leads to the thought it might be overfit slightly.





# Key Findings and Insights

Regressor vs Classifier:  
Classifier models are more efficient  
with less tuning



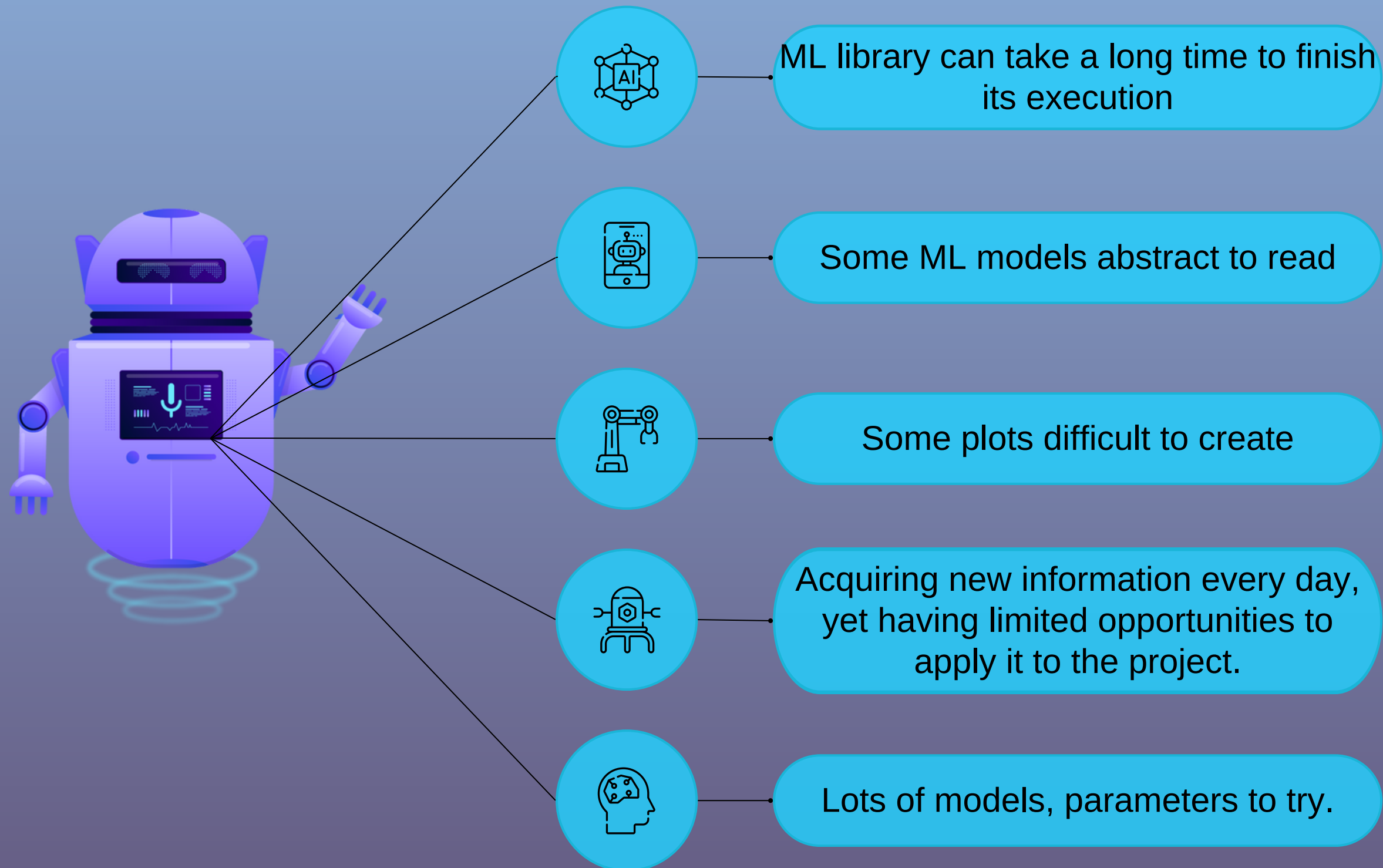
For tasks with clearly defined  
categories, classifiers are your go-to  
option for efficient modeling with  
minimal tuning effort.



Achieving the highest accuracy in  
predictive modeling is a journey  
characterized by "trial and error." With  
each iteration, models are refined and  
improved, demanding patience and  
perseverance to discover the ideal  
configuration.



# Challenges and Learnings



# THANK YOU!

## Credit Score Predictor

Machine Learning Model Applications

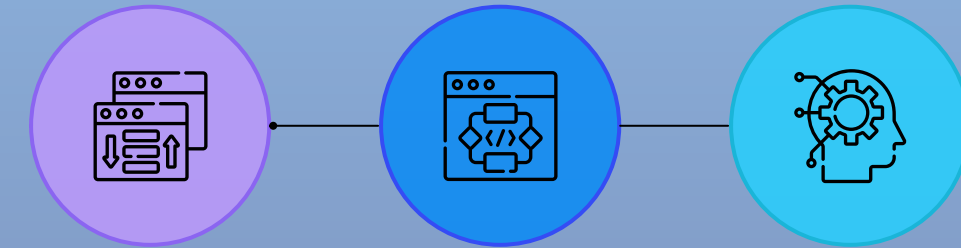
Project by:

Álvaro

Aurélié

Ceci

Tung



**CREDIT SCORE PREDICTOR**  
MACHINE MACHINE LEARNING PROJECT