A PROJECT ON

“CREDIT CARD APPROVAL PREDICTION”

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF
DIPLOMA IN BIG DATA ANALYSIS

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
‘Plot no R/2’, Market yard road,
Behind hotel Fulera,Gultekdi
Pune – 411037.
MH-INDIA

SUBMITTED BY:

Sarvesh Kotgire (75670)

Shubham Acharya (75273)

UNDER THE GUIDENCE OF:
Mrs.Manisha Hingne
Faculty Member
Sunbeam Institute of Information Technology, PUNE.

SUNBEAM

CERTIFICATE

This is to certify that the project work under the title ‘Credit Card Approval Prediction’ is done by Sarvesh Kotgire & Shubham Acharya in partial fulfillment of the requirement for award of Diploma in Big Data Analysis Course.

Mrs. Manisha Hingne
Project Guide

Mrs. Pradnya Dindorkar
Course Co-ordinator

Date:

ACKNOWLEDGEMENT

Our journey through this project has been enriched by the invaluable guidance and unwavering support of the following remarkable individuals:

We extend our heartfelt appreciation to Mr. Nitin Kudale, Center Coordinator at SIIT, Pune, for his insightful input and encouragement. A special thank you to Mrs. Pradnya Dindorkar, Course Coordinator at SIIT, Pune, for her continuous support and guidance throughout our project's development. We are deeply grateful to Mrs. Manisha Hingne, our esteemed Project Guide, whose expertise and mentorship played a vital role in shaping the direction of our work.

Their expertise, encouragement, and genuine interest in our project provided us with the necessary foundation to navigate the complexities of our work. Their constructive feedback and timely advice were instrumental in refining our project at every step, ultimately leading it to its current form.

We also extend our gratitude to the entire faculty and staff members of Sunbeam Institute of Information Technology, Pune, whose continuous support created an environment conducive to learning and growth.

In our journey, their belief in our potential and their commitment to our development have been our guiding lights. We are truly grateful for the opportunities they have provided us and the impact they have had on our project's success.

Sarvesh Kotgire
DBDA March 2023 Batch,
SIIT Pune

Shubham Acharya
DBDA March 2023 Batch,
SIIT Pune

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Introduction And Objectives:

In an increasingly interconnected and fast-paced world, credit cards have become more than just pieces of plastic in our wallets. They have evolved into powerful tools that offer unparalleled convenience, flexibility, and financial empowerment. In today's landscape, where seamless transactions and instant access to funds are paramount, credit cards have emerged as essential companions for modern living. This begs the question: how can we further enhance the credit card experience to align with the needs and aspirations of today's consumers?

Imagine a world where credit card applicants receive recommendations tailored to their unique financial situations and aspirations. A world where the fear of a credit score dip due to rejected applications becomes a thing of the past. This is where the concept of a "Credit Card Approval System" steps in—a solution that not only recognizes the significance of credit cards in today's world but also addresses the concerns and uncertainties that often accompany credit applications.

The Credit Card Approval System represents a groundbreaking innovation in the credit application process by leveraging the capabilities of data analytics and artificial intelligence. This sophisticated system is designed to assess an individual's financial profile comprehensively, offering insights into the likelihood of their credit card application being approved. Imagine a scenario where you receive tailor-made suggestions that take into account factors such as your income, credit history, and other relevant financial details. This not only enhances the accuracy of the approval prediction but also minimizes the risk of any adverse effects on your credit score.

Within the following pages, we will take a profound dive into the inner workings of this remarkable system. We will delve into its myriad benefits, intricate mechanics, and most importantly, the transformative influence it is poised to exert on credit cards and personal finance.

1.2 Problem statement

In today's interconnected world, credit cards have become an integral part of modern financial transactions, offering convenience, flexibility, and purchasing power to individuals. However, the process of applying for a credit card is often accompanied by uncertainties, including concerns about potential negative impacts on an individual's credit score due to application rejections. Additionally, traditional credit assessment methods are time-consuming and manual, leading to delays in decision-making.

To tackle these challenges and offer individuals a frictionless and secure credit card application process, there arises a necessity for a sophisticated machine learning and AI-driven system. The primary objective of this initiative is to create a Credit Card Approval Prediction System that harnesses state-of-the-art technologies to forecast the likelihood of an applicant's credit card application receiving approval or being declined. Importantly, this predictive assessment is carried out in a manner that guarantees no negative repercussions on the individual's credit score as a result of inquiring about the prediction.

1.3 Objectives

Automated Credit Assessment: Design and implement a machine learning model that can analyze user-provided data, including financial history, income levels, and other relevant factors, to predict the likelihood of credit card approval.

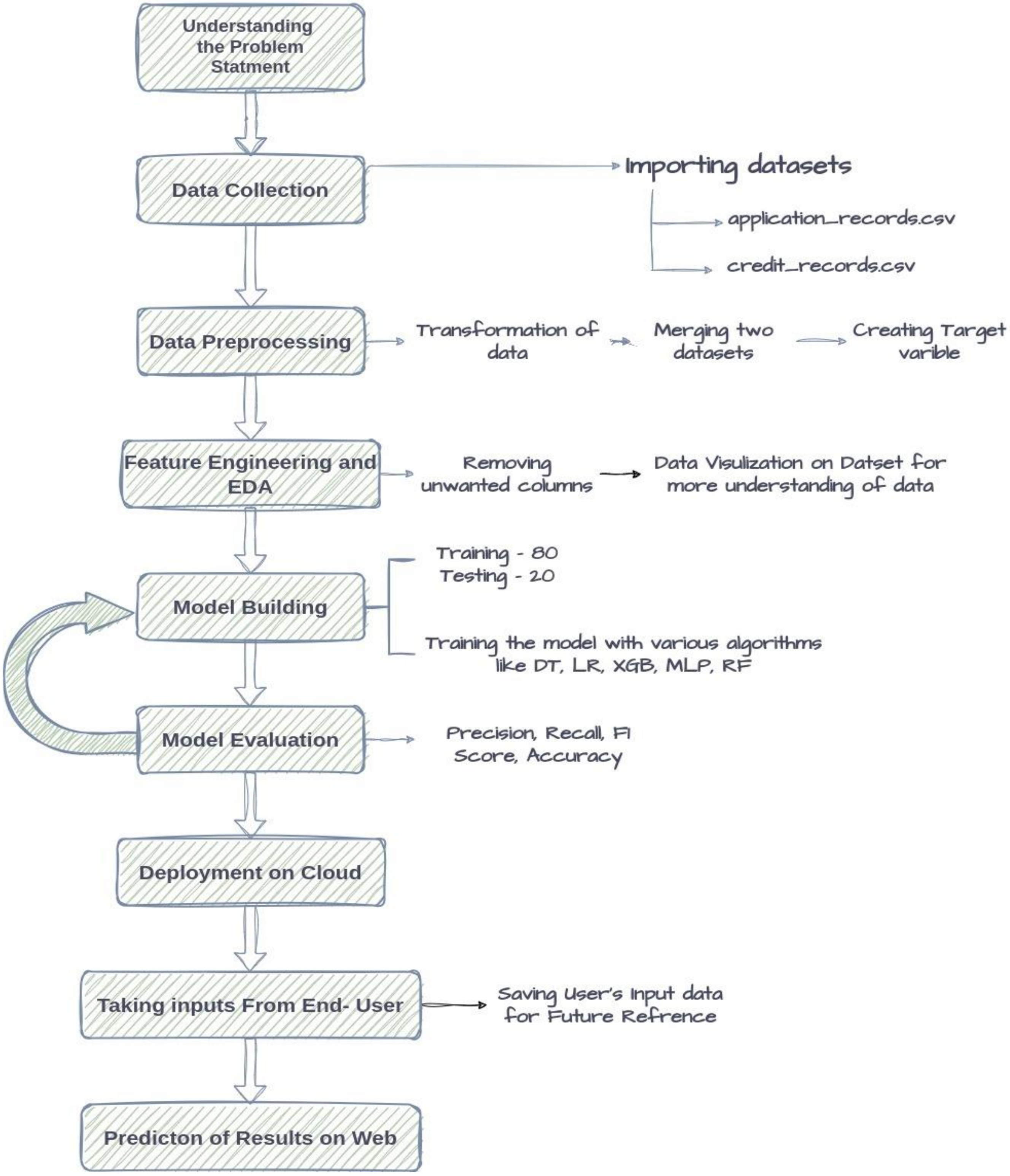
Minimized Credit Score Impact: Develop an innovative approach that allows individuals to check their credit card approval prediction without affecting their credit score, eliminating the fear of application rejections leading to credit score reductions.

User-Friendly Interface: Create an intuitive user interface that enables users to input their information easily and view the prediction outcome, enhancing the overall user experience.

Accurate Predictive Model: Train the machine learning model using a comprehensive dataset of historical credit card application data, optimizing its accuracy and minimizing false predictions.

Real-time Processing: Develop the system to provide instant predictions, reflecting the need for timely decisions in the fast-paced financial landscape.

Project Flow Diagram



Understanding the Problem Statment

Data Collection

Importing datasets

- application_records.csv
- credit_records.csv

Data Preprocessing

Transformation of data → Merging two datasets → Creating Target varible

Feature Engineering and EDA

Removing unwanted columns → Data Visulization on Datset for more understanding of data

Model Building

- Training - 80
- Testing - 20
- Training the model with various algorithms like DT, LR, XGB, MLP, RF

Model Evaluation

Precision, Recall, F1 Score, Accuracy

Deployment on Cloud

Taking inputs From End- User

Saving User's Input data for Future Refrence

Predicton of Results on Web

2. DATA EXPLORATION AND ANALYSIS:

2.1 Understanding the Datasets

From the well-known data science related website – Kaggle we found a data source distributed among two separate datasets. The datasets were with following attributes:

1) application_record.csv

This dataset has 438557 entries, 18 columns in it.

The 18 columns, after renaming them to proper meaningful names, are as follows:

```
Data columns (total 18 columns):

 #   Column              Dtype
---  ------              --------------
 0   ID                  int64
 1   Gender              object
 2   Own_car             object
 3   Own_property        object
 4   Children_cnt        int64
 5   Income              float64
 6   Income_source       object
 7   Education           object
 8   Family_status       object
 9   Housing_type        object
 10  Age                 int64
 11  Experience          int64
 12  Mobile              int64
 13  Work_phone          int64
 14  Phone               int64
 15  Email               int64
 16  Occupation_type     object
 17  Family_size         float64
```

```
dtypes: float64(2), int64(8), object(8)
```

2) credit_record.csv

It has 3 columns:

```
#   Column               Dtype
---  ------             --------------
 0   ID                  int64
 1   MONTHS_BALANCE      int64
 2   STATUS              object
dtypes: int64(2), object(1)
```

2.2 Observations and primary issues found

1) Both the datasets have ‘ID’ as primary key in them.
2) The column names were quite meangless and need renaming.
3) The columns ‘Gender’, ‘Own_car’, ‘Own_property’, ‘Mobile’, ‘Work_phone’, ‘Phone’, and ‘Email’ have binary records in them in the form of Yes or No.
4) The values in columns ‘Age’ and ‘Experience’ are given in days not years.
5) There are many categorical columns present in the dataset.
6) The second dataset has two columns ‘MONTHS_BALANCE’ and ‘STATUS’ which cannot be comprehended without any domain knowledge.
7) There are null values present in the first dataset.
8) There is no target variable.
9) The datasets need to be merged so as to do the processing in a proper manner.

3. METHODOLOGY

3.1 Packages used

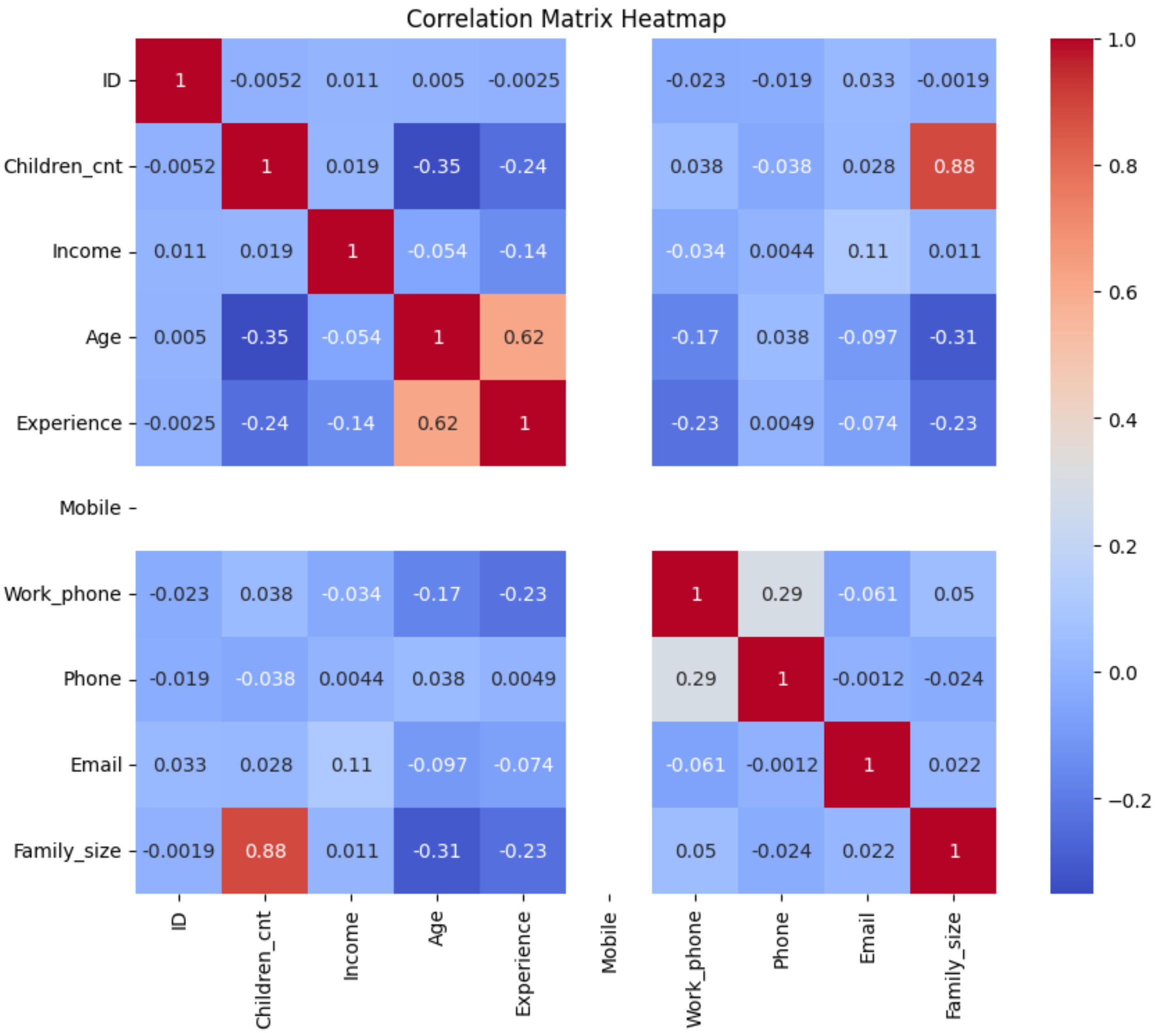
1) Pandas
2) Numpy
3) Matplotlib
4) Seaborn
5) Sci-kit learn
6) Math

3.2 Processing on application_record.csv

3.2.1 Preprocessing

The missing values are handled by dropna() and fillna() based on proportion of them in the particular columns. The fillna() method is used when that proportion is very high (In the ‘Occupation_type’ column).

The Correlation Matrix heatmap using Seaborn was used to understand the data better.



Based on the correlation data, it was found that the column ‘Children_cnt’ is highly correlated with the column ‘Family_size’. Hence the former column was removed.

Similarly, the column ‘Mobile’ was found to be having only one value in it. It was dropped as well.

3.2.2 Encoding

The columns in the list - ['Gender', 'Own_car', 'Own_property', 'Income_source', 'Education', 'Family_status', 'Housing_type', 'Occupation_type'] were categorical. A LabelEncoder was applied to convert these categorical columns into numerical for better processing.

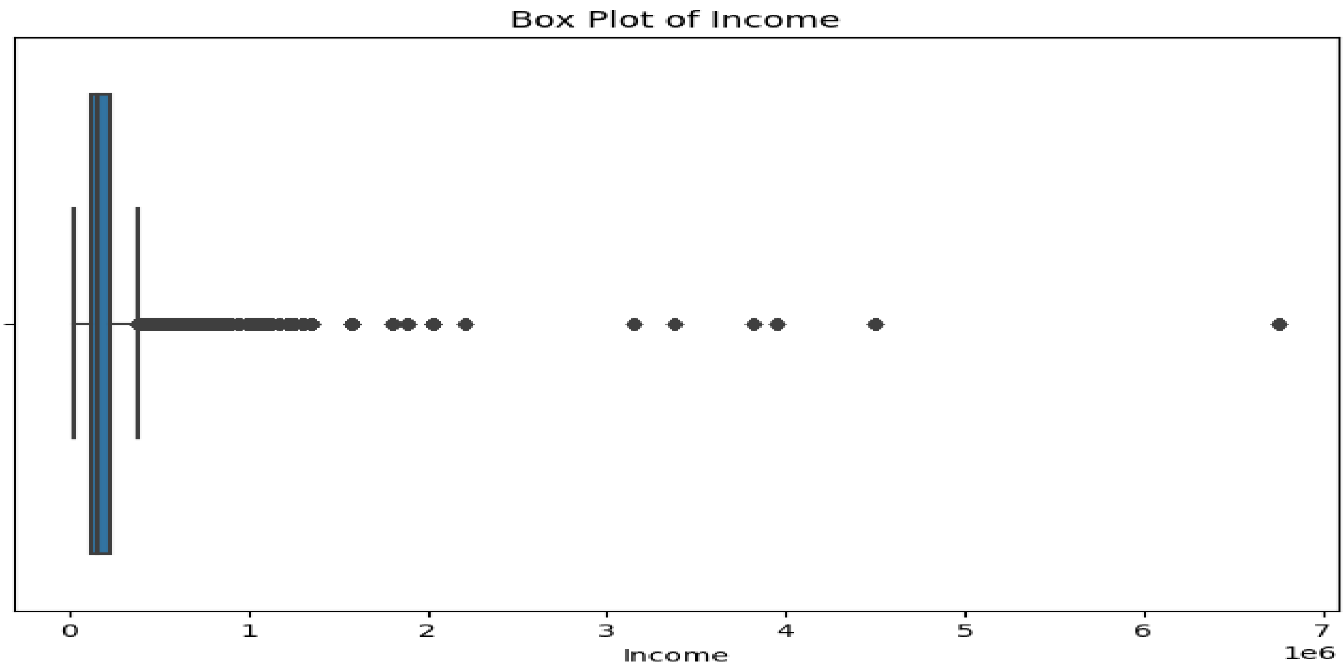
3.2.3 Feature Transformation

The columns ‘Experience’ and ‘Age’ were converted from days to years format. Later using ‘math’ package, the ‘Age’ and ‘Experience’ columns were converted to proper integers using floor() function.

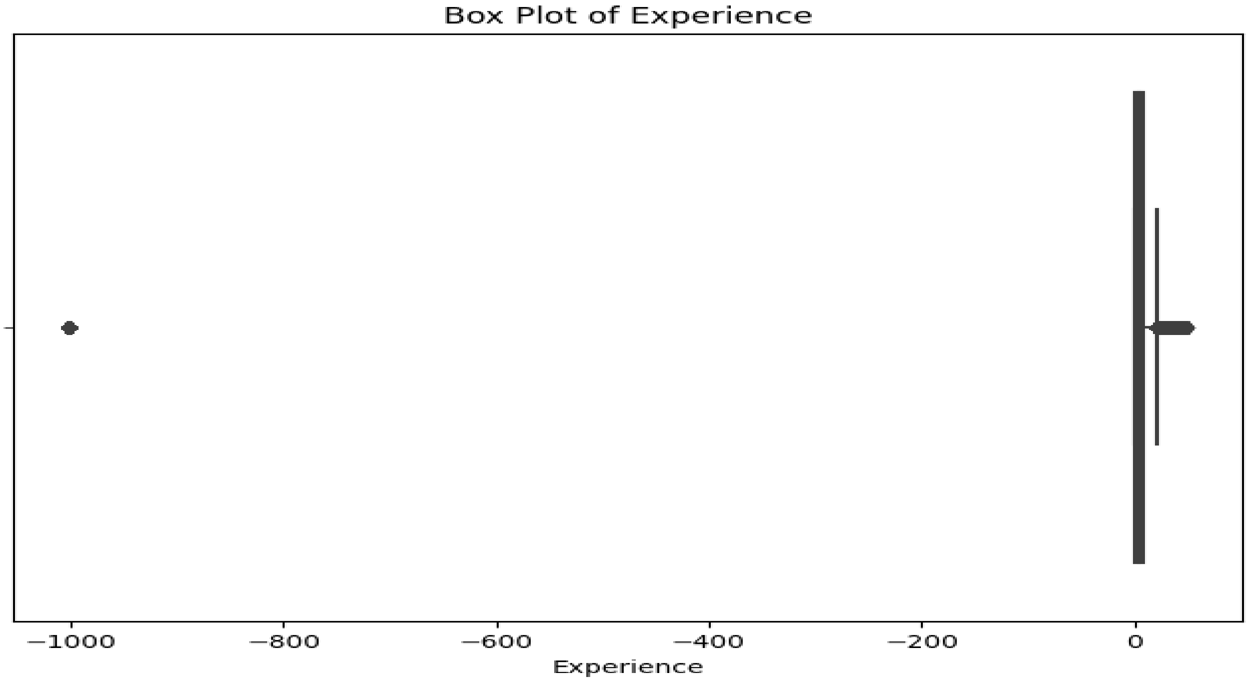
3.2.4 Feature Scaling

Numerical columns with binary values do not require scaling. There was one ‘Income’. Whether to scale it or not was decided on TRY AND ERROR METHOD based on model performance.

3.2.5 Outliers detection and handling



There were almost 20000 outliers found in the ‘Income’ column. The IQR method was used to remove them. The upper and lower limits were decided on the following formulae:
upper_bound = Q3 + 1.5 * IQR
lower_bound = Q1 - 1.5 * IQR
These were successfully removed.



The ‘Experience’ column had 73800 values as outliers. As experience cannot be below 0, these outliers below lower threshold were handled by replacing them with 0.

3.3 Processing on credit_record.csv

To understand the column ‘STATUS’, following information was gathered.
X: No loan for the month
C: paid off that month
0: 1-29 days past due
1: 30-59 days past due
2: 60-89 days overdue
3: 90-119 days overdue
4: 120-149 days overdue
5: Overdue or bad debts, write-offs for more than 150 days
A new feature was extracted names ‘STATUS_BINARY’ which basically distributed the records as ‘GOOD’ and ‘BAD’ customers based on the ‘STATUS’ column. The C and X

were given ‘GOOD’ status while the rest were given ‘BAD’.

Afterwards the ‘STATUS_BINARY’ column was further recreated in two columns named ‘Good’ and ‘Bad’ which consisted the individual’s number of records in the particular fields. These two columns were finally used to create a new and a very important column ‘Good_Rate’ which was created to know the chances of a customer being good. The formula we used is - df['Good']+1) / (df['Bad']+df['Good']+1.

The value of ‘Good_Rate’ lies between 0 and 1 and it approaching 1 means better the chances of that individual to get the status of good customer.

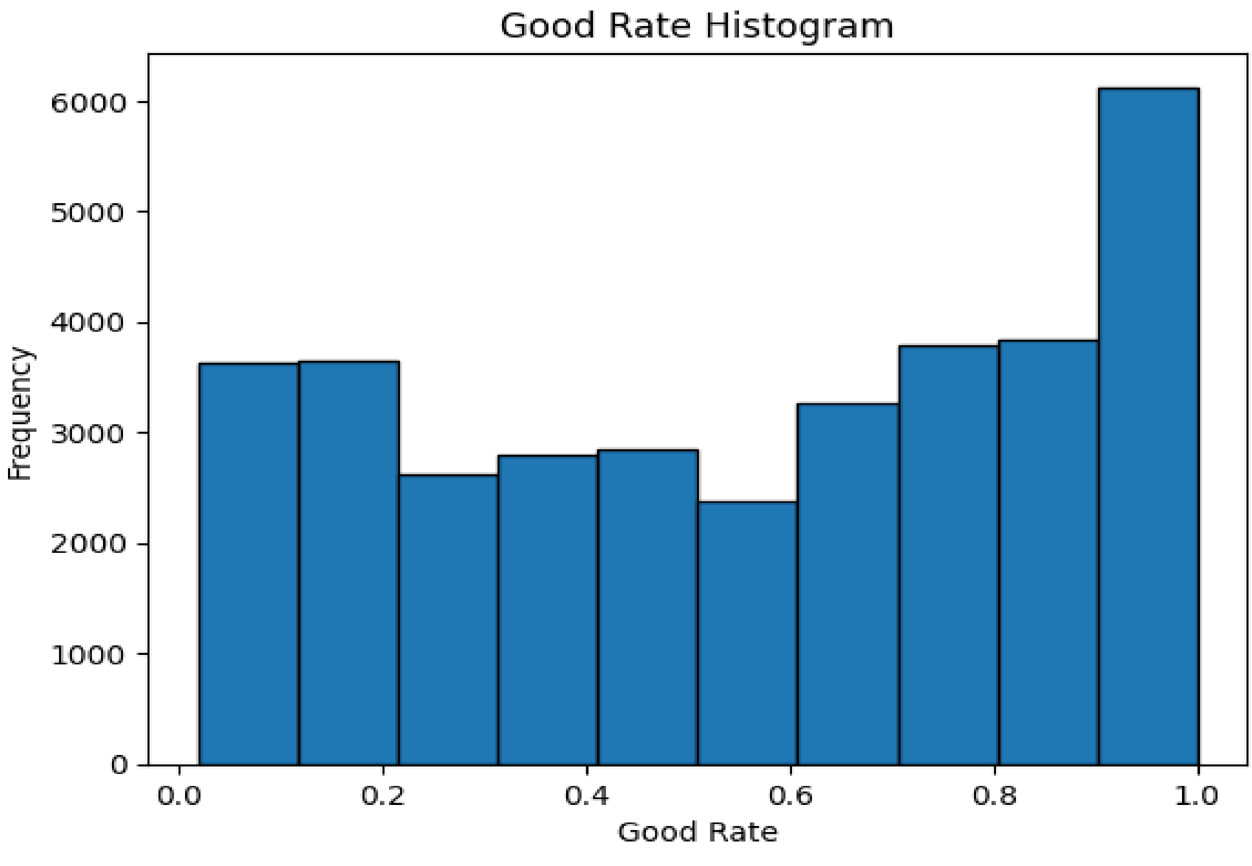
3.4 Merging

Inner join was used to merge the datasets based on the ‘ID’ column.

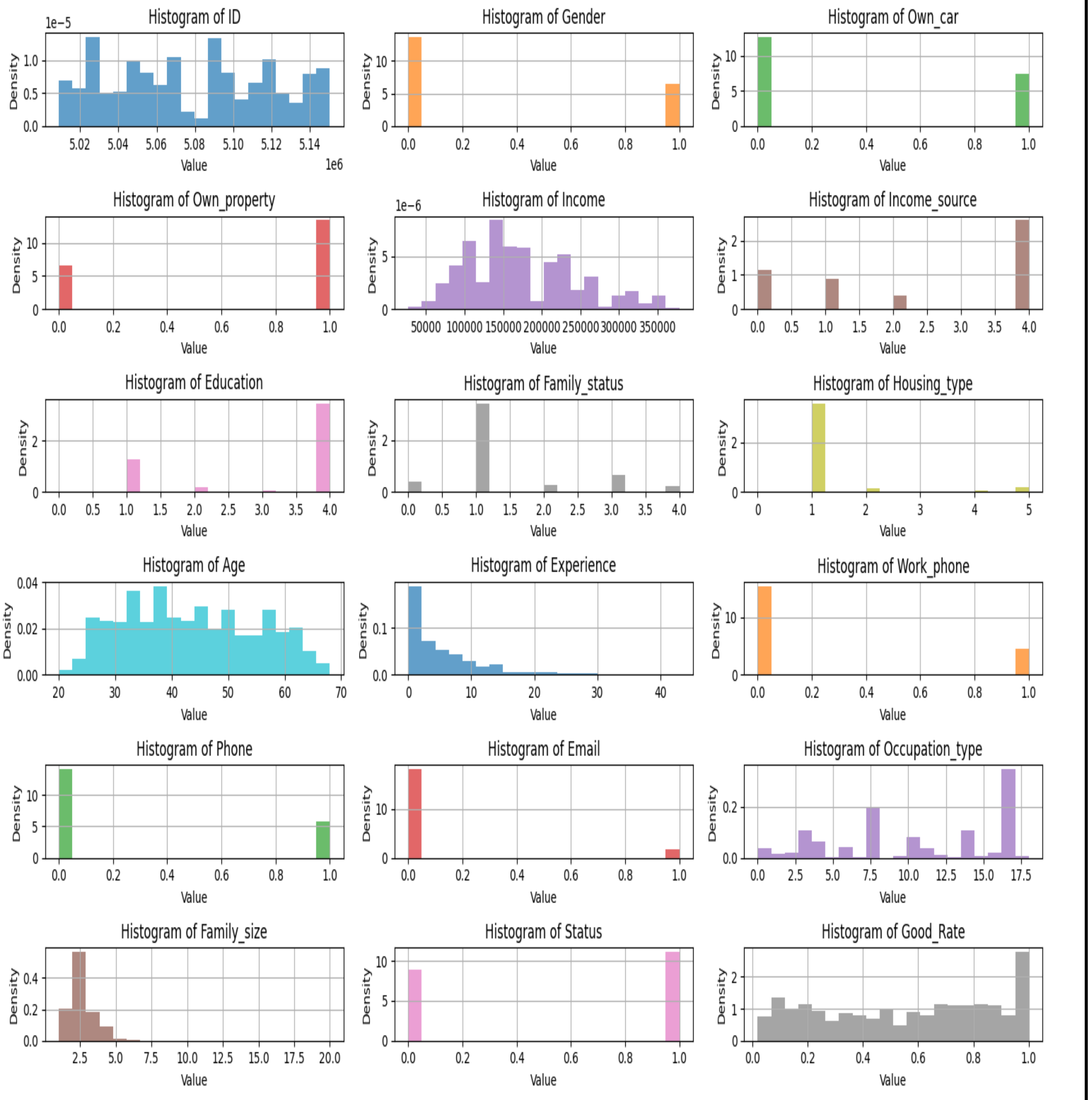
Upon merging, the final dataset had total 34928 records with 18 columns.

The target variable was ‘Status’ which has binary nature for 0 being credit card rejected and 1 being it is accepted.

3.5 EDA



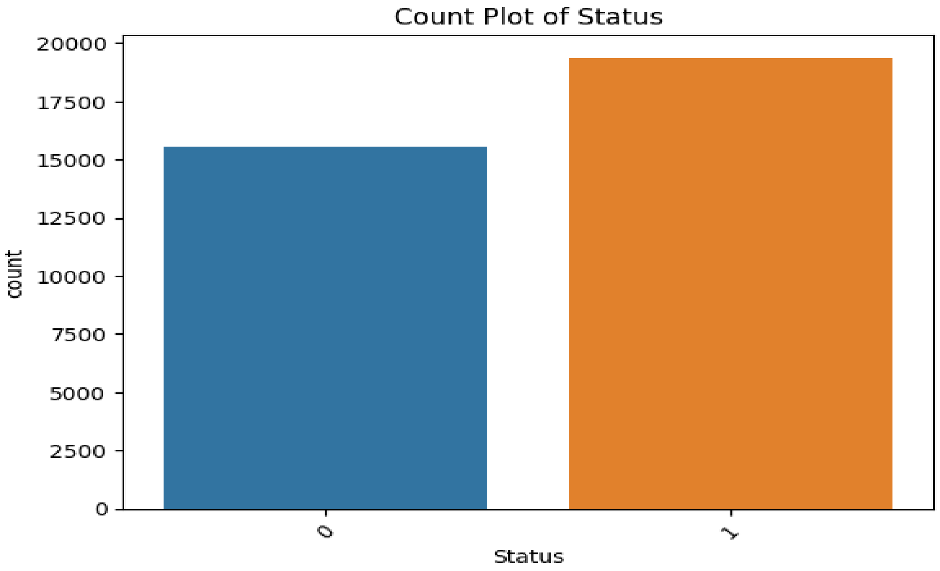
Here we can clearly see that the ‘Good_Rate’ column has values between 0 and 1.



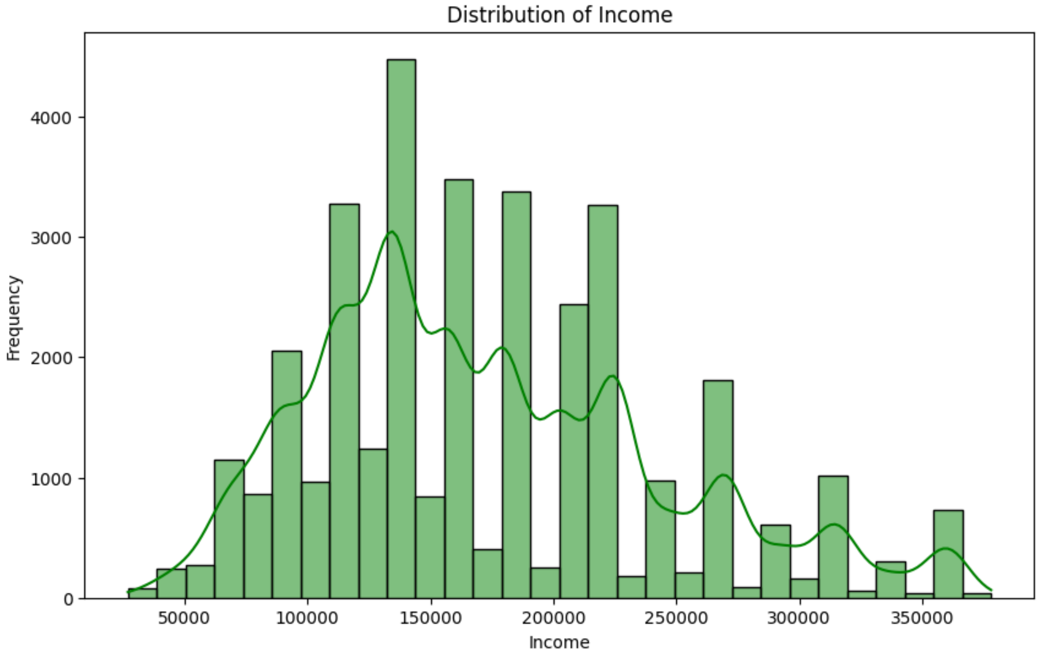
Various histograms were plot to analyse the different columns. Based on the plots we can decide which value is dominating the categorical values, which values are more in the rate column, what is the average family size in the dataset, the age distribution.

It also clearly tells us that one housing type is a way too dominating hence we can drop the entire column.

We can also see that the columns ‘Phone’, ‘Email’, ‘Work_phone’ are majorly dominated by the one set of value which leads to underrepresentation of the other value. We can drop these columns.

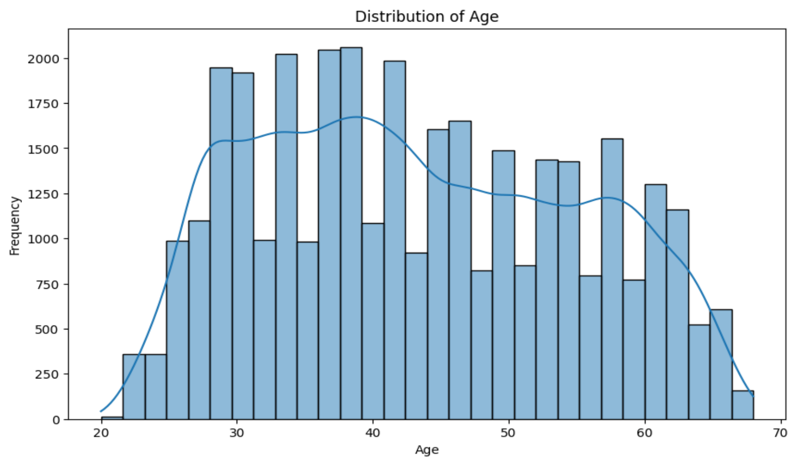


This graph tells us the distribution of the output variable. Fortunately no value is completely outperforming the other value.

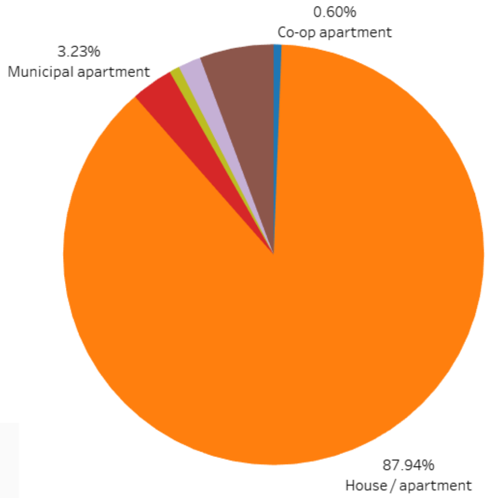


The above graph shows the distribution of income in the dataset. We can infer that the annual income between 90000 to 275000 is more frequent. It further tells us that the project is mostly aimed towards the tier 2 cities with medium income.

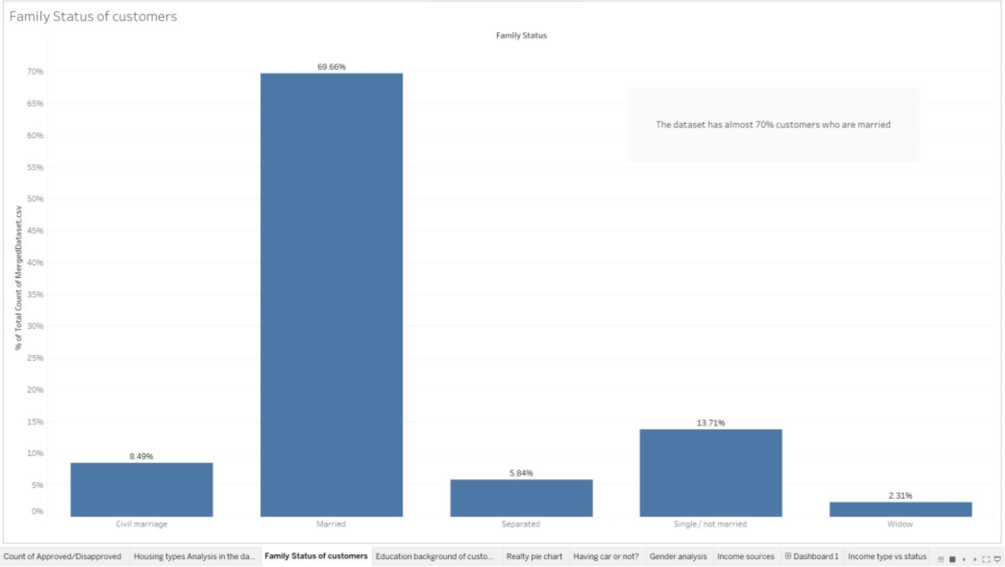
The following graph shows the distribution of age. It pretty much matches with the normal distribution of age.



Housing types Analysis in the dataset



The dataset majorly has information about those customers who live in House/Apartment


Family Status of customers

The dataset has almost 70% customers who are married

Based on EDA, following columns were dropped: Own_Car, Housing_type, Email, Phone, Work_phone.

4. MODEL TRAINING

4.1 Splitting the dataset

This is the point where the project takes a branched route. In one case, we split the dataset with keeping the ‘Good_Rate’ column which is taken as a input from user during predictions (this is done by providing options to the user and assigning ‘Good_Rate’ value on the basis of selected value) and then all the remaining processes till the model creation is done. In the other case we didn’t take the ‘Good_Rate’ column to accommodate the user with no prior credit history to his name.

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=123)
```

4.2 Oversampling, transforming, scaling

4.2.1 Oversampling using ADASYN

```
from imblearn.over_sampling import SMOTE , ADASYN
sm = ADASYN()
x_train, y_train = sm.fit_resample(x_train,y_train)
```

ADASYN (Adaptive Synthetic Sampling) is an oversampling technique used to address the class imbalance problem in machine learning. In this project, Class imbalance occured since in few columns the number of instances in one class significantly outweighs the number of instances in another class, which can lead to biased model performance.

ADASYN is specifically designed to handle imbalanced datasets by generating synthetic samples for the minority class.

4.2.2 Transformation

Power transformations are a type of data transformation commonly used to stabilize variance and make the data distribution more closely resemble a normal distribution.

```
from sklearn. preprocessing import PowerTransformer
power = PowerTransformer(method='yeo-johnson')
x_train = power.fit_transform(x_train)
x_test = power.fit_transform(x_test)
```

The Yeo-Johnson method is a power transformation that works with both positive and negative values, including zero.

4.2.3 Scaling

```
from sklearn import preprocessing
normalizer = preprocessing.StandardScaler()
x_train = normalizer.fit_transform(x_train)
x_test = normalizer.fit_transform(x_test)
```

Standardization is a preprocessing step that transforms the data to have a mean of 0 and a standard deviation of 1. This can be helpful for machine learning algorithms that assume a Gaussian distribution and are sensitive to the scale of the features.

4.3 Algorithms applied

(1) Random Forest: is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

(2) Naive Bayes is a simple yet powerful classification algorithm based on the principles of Bayes' theorem. It is widely used for various machine learning tasks, particularly in natural language processing, text classification, and spam filtering. Despite its "naive" assumption of feature independence, the algorithm often performs surprisingly well in practice and is computationally efficient. It performs well on the data with low correlation as well.

(3) XGBoost: or extreme gradient boosting is one of the well-known gradient boosting_techniques (ensemble) having enhanced performance and speed in tree-based (sequential decision trees) machine learning algorithms. XGBoost was created by Tianqi Chen and initially maintained by the Distributed (Deep) Machine Learning Community (DMLC) group. It is the most common algorithm used for applied machine learning in competitions and has gained popularity through winning solutions in structured and tabular data. It is open-source software. Earlier only python and R packages were built for XGBoost but now it has extended to Java, Scala, Julia and other languages as well.

(4) Logistic Regression:

Logistic Regression is a statistical algorithm used for binary and multiclass classification tasks. Despite its name, logistic regression is a classification algorithm rather than a regression algorithm. It's widely used due to its simplicity, interpretability, and efficiency. The primary goal of logistic regression is to predict the probability that a given input belongs to a certain class.

(5) Decision Tree: algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record’s attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

(6) MLP Classifier

Multilayer Perceptron (MLP) Classifier:

The Multilayer Perceptron (MLP) is a type of artificial neural network that's commonly used for classification and regression tasks. It's a versatile and powerful model that can capture complex relationships in data.

Advantages:

- Can capture complex nonlinear relationships in data.
- Performs well on a wide range of tasks, from simple to highly complex.
- Suitable for large datasets and high-dimensional feature spaces.
- Can automatically learn features from raw data.

The models were trained using above algorithms. Both test and train accuracies were calculated to check if there is any overfitting. It wasn’t the case.

4.4 Model Evaluation

Models were evaluated on the following parameters:

1. Accuracy
2. Precision
3. Recall
4. F1 score

The result obtained is as follows:

Algo	Accuracy	Precision	Recall	F1
RF	0.99	1.00	0.98	0.99
NB	0.98	1.00	0.96	0.98
XGB	0.99	1.00	0.98	0.99
LR	0.98	1.00	0.96	0.98
DT	0.99	1.00	0.98	0.99
MLP	0.98	1.00	0.96	0.98

4.5 Result

- The performance metrics showcase the algorithms' effectiveness in classifying the dataset. The accuracy metric measures the overall correctness of predictions, while precision indicates the ratio of correctly predicted positive observations to the total predicted positives. Recall, also known as sensitivity or true positive rate, highlights the ratio of correctly predicted positive observations to the total actual positives. The F1 score combines both precision and recall, providing a balanced measure that considers false positives and false negatives. The high accuracy, precision, recall, and F1 score across multiple algorithms demonstrate their robustness in handling the classification task. The results indicate strong predictive capabilities and consistent performance, suggesting that these algorithms are suitable candidates for our application.

- We also calculated and compared testing and training accuracies. In all cases, it was found to be close values to each other indicating that the model is likely performing well and generalizing effectively to new, unseen data. This similarity between training and test accuracy indicates that the model has not overfit or underfit the training data.

- During the first few test runs the accuracies were in the range of 50% to 65%. The usage of ADASYN oversampling, transformer and standardisation helped increase this accuracy significantly.

4.6 Saving models

Models were saved using pickle package.

```
import pickle

models = {
    "modelDT.pkl": model_dt,
    "modelXGB.pkl": model_xgb,
    "modelRF.pkl": model_rf,

    "modelNB.pkl": model_nb,
    "modelLR.pkl": model_lr,
    "modelMLP.pkl": model_mlp
}

for filename, model_instance in models.items():
    with open(filename, 'wb') as file:
        pickle.dump(model_instance, file)
```

5. GUI and Model Deployment

Flask Framework:

Flask is a lightweight and popular web framework for building web applications and APIs using the Python programming language. It is designed to be simple, flexible, and easy to use, making it an excellent choice for both beginners and experienced developers. Flask provides the essential tools and components needed to create web applications without imposing a rigid structure.

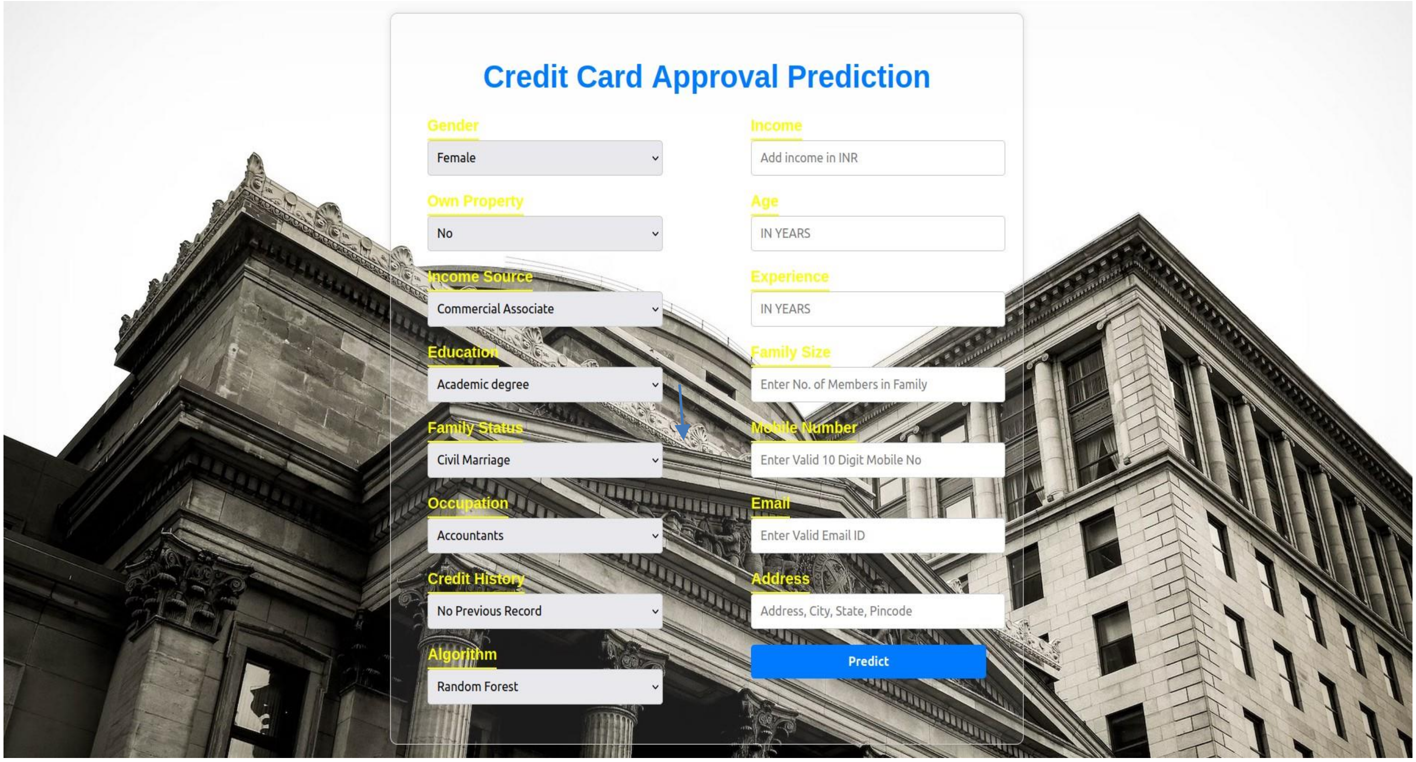
Flask Usage in UI Creation:

Flask can be used to create user interfaces (UIs) for web applications. While Flask is not a dedicated UI framework like React or Angular, it can serve HTML templates and dynamic content to users, making it suitable for creating UIs in combination with HTML, CSS, and JavaScript.

In essence, Flask can be used to create user interfaces by generating dynamic HTML templates, serving static files, handling form submissions, and integrating with frontend technologies. It's a versatile tool for creating web applications with a user-friendly and interactive interface.

We again used pickle to load the saved models and used in the Flask application.

As an additional functionality, we incorporated a file handling function which saves all the inserted data of user in a csv file. We can use this data in many ways.



Credit Card Approval Prediction

Gender: Female

Own Property: No

Income Source: Commercial Associate

Education: Academic degree

Family Status: Civil Marriage

Occupation: Accountants

Credit History: No Previous Record

Algorithm: Random Forest

Income: Add income in INR

Age: IN YEARS

Experience: IN YEARS

Family Size: Enter No. of Members in Family

Mobile Number: Enter Valid 10 Digit Mobile No

Email: Enter Valid Email ID

Address: Address, City, State, Pincode

The User Interface that we created it shown above.
Finally, the web application was deployed on AWS cloud.

6. CONCLUSION

6.1 Conclusion

In the rapidly evolving landscape of financial services, where access to credit plays a pivotal role in individuals' financial well-being, the development of accurate and efficient credit card approval prediction systems without any negative effects becomes increasingly significant. This project aimed to address this need by leveraging machine learning techniques to create a predictive model capable of assessing credit card approval applications.

Through a systematic approach that encompassed data collection, preprocessing, feature engineering, model selection, and evaluation, we successfully developed a credit card approval prediction system. By utilizing a diverse set of algorithms including Random Forest, Naive Bayes, XGBoost, Logistic Regression, Decision Tree, and Multilayer Perceptron, we were able to build a range of models that exhibited impressive performance across multiple evaluation metrics.

Our results underscored the effectiveness of these algorithms in accurately predicting credit card approval outcomes. With high accuracy, precision, recall, and F1 scores, our models demonstrated their robustness in classifying credit card applications, thereby aiding financial institutions in making informed decisions while ensuring the creditworthiness of applicants.

Furthermore, the project offered insights into the importance of feature engineering and data preprocessing in enhancing model performance. We explored the impact of various techniques, such as handling missing values, encoding categorical variables, and scaling features, on the overall predictive power of the models.

The deployment phase involved creating a user-friendly interface that allowed applicants to input their information and receive instant predictions about their credit card approval status. Additionally, the application ensured that these predictions did not impact individuals' credit scores, providing a valuable and secure resource for users.

As the financial sector continues to evolve, the development of predictive models holds immense potential to streamline credit assessment processes, minimize human bias, and enhance the efficiency of decision-making. While our project achieved notable success, there remain opportunities for further refinement, such as exploring ensemble methods, fine-tuning hyperparameters, and incorporating additional data sources.

In conclusion, this project represents a significant step forward in the realm of credit card approval prediction. The amalgamation of machine learning techniques, data preprocessing, and user-friendly deployment underscores the convergence of technology and finance for the betterment of both financial institutions and applicants.

6.2 Future Work

1) Exploring Ensemble Methods:

While the individual algorithms employed in this project demonstrated strong performance, the potential benefits of ensemble methods could be explored further. Techniques like Random Forest, Gradient Boosting, and Stacking involve combining multiple models to improve overall predictive accuracy. Experimenting with ensemble techniques could lead to even more robust and accurate predictions.

2) Incorporating Additional Data Sources:

The power of predictive models often relies on the quality and diversity of data. Exploring additional data sources, such as alternative credit data, socioeconomic indicators, or external economic trends, could provide valuable insights that enhance the predictive capabilities of the model. Integrating such data may help capture more complex patterns and nuances.

3) Model Interpretability:

Exploring methods for model interpretability, such as feature importance analysis, SHAP values, or LIME (Local Interpretable Model-agnostic Explanations), can provide valuable insights into the factors that influence credit card approval decisions.

4) Dynamic Model Updates:

Financial landscapes are dynamic, and credit assessment criteria may change over time. Developing mechanisms to periodically retrain and update the model with the most recent data ensures that the model's predictive performance remains accurate and aligned with evolving industry standards.

5) Cross-Validation and External Validation:

While the models' performance was evaluated using train-test splits, exploring cross-validation techniques and validating the models on external datasets can provide a more comprehensive understanding of their generalization capabilities.

6) Deployment Scalability:

As user demand grows, scalability becomes essential. Transitioning from single-instance deployments to cloud-based solutions that can handle higher traffic loads while maintaining low latency is a significant consideration for the future.