INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 08 ISSUE: 07 | JULY - 2024 SJIF RATING: 8.448 ISSN: 2582-3930

LOAN AMOUNT ANALYSIS AND PREDICTION USING MACHINE LEARNING

[1] Manoj K M, [2] Dr. Geetha M
[1] Student, Department of MCA, BIET, Davanagere.
[2]Associate Professor, Department of MCA, BIET, Davanagere.
Email:manojkm2001dvg@gmail.com

ABSTRACT

This research paper presents a comprehensive analysis and prediction of loan sanction amounts using multiple machine learning models. The dataset includes various customers attributes such as income, expenses, credit score, and employment type. We preprocess the data, perform exploratory data analysis (EDA) [1], and evaluate the performance of several regression models, including Linear Regression, Bagging Regressor, Decision Tree Regressor, and Random Forest Regressor.

Keywords: *Linear Regression, Bagging Regression, Decision Tree, Random Forest Regression.*

I. INTRODUCTION

The Prediction models use data mining, statistical analysis, and probability to forecast outcomes based on predictors. These models, which range from simple linear equations to complex neural networks [2], improve over time with more data, enhancing accuracy and reducing decision-making risks and time. In banking, prediction models are crucial for minimizing risk in loan approvals by evaluating parameters such as credit score, income, age, marital status, and gender, there by reducing defaulters and streamlining the process. Manual loan approvals are prone to errors and delays, potentially leading to financial losses for banks and economic instability. The project aims to develop a loan amount analysis and prediction model to optimize the approval process [3].

This model seeks to automate the loan approval process, thereby increasing efficiency, reducing human error, and enhancing overall communication within banking departments. By employing advanced machine learning algorithms such as the Random Forest Regressor, the system aims to deliver higher accuracy in predicting loan eligibility and default risk compared to traditional methods [4]. The ultimate goal is to provide a quick and reliable solution for both applicants and bank emp oyees, ensuring a smoother and more profitable operation for financial institutions.l

II. LITERATURE SURVEY

The literature on loan amount analysis and prediction encompasses a variety of approaches and methodologies aimed at improving the accuracy and efficiency of loan approval processes [5]. Traditional systems have often relied on basic statistical methods and simple linear regression models, which are limited in their ability to handle complex relationships and diverse data sets. These methods typically involve straight forward imputation techniques for missing data, such as mean, median, or mode imputation, which may not adequately preserve the data's integrity and predictive power. Furthermore, categorical feature encoding in these systems is often simplistic, potentially leading to poor model performance.

Recent advancements have introduced more sophisticated data preprocessing techniques, including the use of advanced imputation methods and robust categorical feature encoding strategies like label encoding and specific replacements for binary features. Visualization tools, such as heatmaps, are utilized to better understand and address missing data. The integration of multiple machine learning models, including Linear Regression, Bagging Regressor, Decision Tree Regressor, and Random Forest Regressor, has significantly enhanced the predictive accuracy and reliability of loan amount predictions.

Studies have shown that leveraging historical data and advanced analytics can lead to more precise predictions and better risk management. Machine learning models, particularly ensemble methods like Random Forest, have demonstrated superior performance over traditional linear regression models by effectively capturing complex patterns and interactions within the data. These models also facilitate automated decision-making processes, reducing the time and administrative effort required for loan approvals.

Furthermore, the application of machine learning in loan prediction systems has enabled more objective and data-driven decision-making, minimizing human biases and inconsistencies. The incorporation of a wide range of data points, such as credit scores, income, age, marital status, and gender, allows for a more comprehensive assessment of borrower credit worthiness [6]. This approach not only improves the accuracy of loan amount predictions but also enhances the overall efficiency of the loan approval process.

Moreover, advanced risk modeling techniques have been employed to better assess and manage credit risk, leading to lower default rates and improved portfolio performance. These techniques enable lenders to tailor loan offers to individual borrower profiles, providing personalized and flexible lending options that better meet borrowers' needs. The scalability of modern loan prediction systems ensures they can handle large volumes of applications and adapt to changing market conditions.

In conclusion, the literature highlights a significant shift from traditional, simplistic methods to advanced machine learning approaches in loan amount analysis and prediction [7]. These advancements offer substantial improvements in accuracy, efficiency, and risk management, ultimately benefiting both lenders and borrowers by streamlining the loan approval process and enhancing financial outcomes.

III. METHODOLOGY

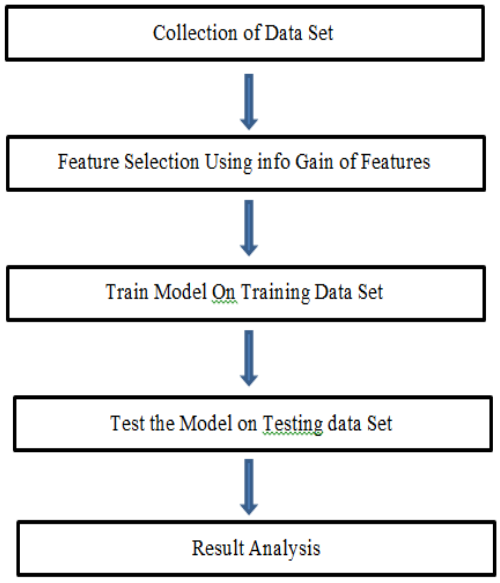

Fig: 3.1 Dataset Training & Result Analysis

- **Collection of Data Set:** The dataset is loaded from a CSV file using pd.read_csv("classified-data.csv").
- **Feature Selection Using Info Gain of Features:** Although the program does not explicitly use Information Gain for feature selection, it processes and selects relevant features by handling missing values, encoding categorical variables, and identifying numerical and categorical features.

- **Train Model on Training Data Set:** The data is split into training and testing sets using train_test_split. Various models are trained on the training set (X_train, y_train), including Linear Regression, Bagging Regressor, Decision Tree Regressor, and Random Forest Regressor.
- **Test the Model on Testing Data Set:** After training, the models are tested on the testing set (X_test) to evaluate their performance. The program calculates the R^2 score for each model, which is a measure of how well the predictions match the actual values in the testing set.
- **Result Analysis:** The R^2 scores for different models are printed to compare their

performance. Visualizations such as Count plots of predictions are created to analyze the model's performance.

-

IV. TECHNOLOGIES USED

4.1 RANDOM FOREST REGRESSION:

The random forest algorithm is a machine learning method that creates many decision trees and combines their results to make a prediction. It works by taking random samples of the data and picking random subsets of features to build each tree. This randomness helps prevent the model from overfitting, which means it doesn't get too tailored to the training data and performs better on new, unseen data. Random forests are good at handling complex datasets and can provide insights into which features are most important for making predictions. Overall, this method is known for its accuracy and reliability in predicting outcomes.

Formula:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} f_t(x)$$

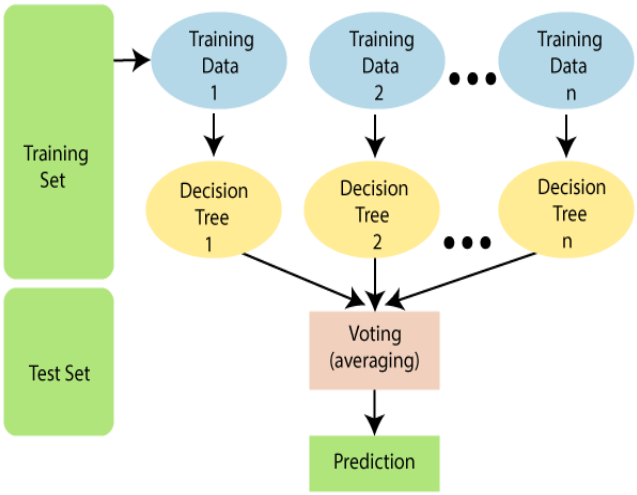

Fig 4.1: Flowchart of Random Forest Algorithm

4.2 DECISION TREE

Decision tree regression is a machine learning method used for predicting continuous values. It works by splitting the data into smaller and smaller groups based on different features, creating a tree-like structure. At each split, the algorithm chooses the feature and value that best divide the data to minimize prediction errors. This process continues until the groups are small enough or a stopping condition is met, such as a maximum tree depth. The final prediction is made by averaging the values in the leaf nodes of the tree. Decision tree regression is easy to understand and interpret, making it a popular choice for various regression tasks.

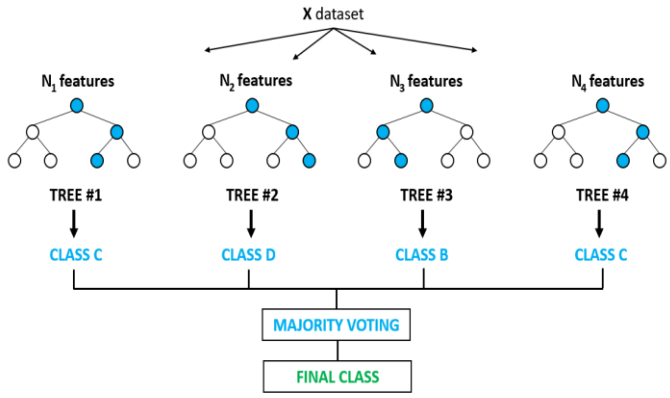

Fig 4.2: Flowchart of Decision TreeAlgorithm

4.3 BAGGING REGRESSION

Bagging regression is a machine learning technique that improves the accuracy of predictions by combining the results of multiple regression models. It works by creating many different subsets of the original data through random sampling with replacement. Each subset is used to train a separate regression model. The final prediction is made by averaging the predictions from all these models. This approach reduces errors and helps prevent overfitting, making the overall model more robust and reliable. Bagging is particularly useful when the individual models are prone to high variance.

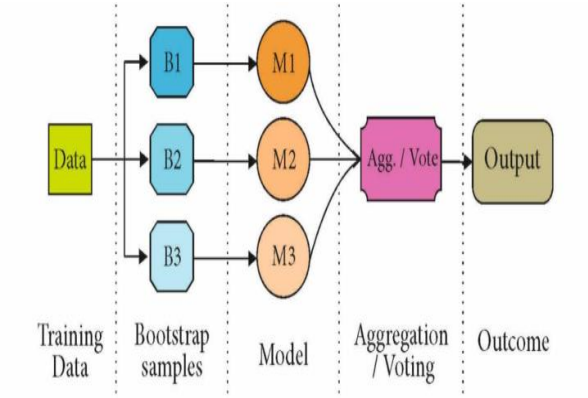

Fig 4.3: Flowchart of Bagging Regression Model

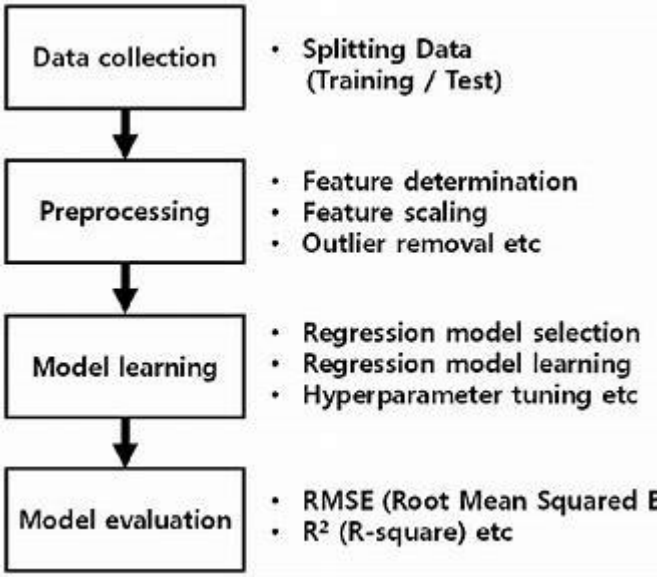

Fig 4.4: Flowchart of Linear Regression Model

4.4 LINEAR REGRESSION

Linear regression is a simple and widely-used algorithm for predicting a continuous outcome based on one or more input features. It works by finding the best-fit line (in the case of one feature) or a hyper plane (for multiple features) that minimizes the differences between the predicted and actual values. The algorithm calculates coefficients for each feature that indicate their contribution to the outcome. These coefficients are adjusted during training to reduce prediction errors. Linear regression is easy to understand and implement, making it a fundamental tool in statistics and machine learning for tasks like forecasting and trend.

Formula:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

V. IMPLEMENTATION

The loan amount analysis and prediction lies in the advancement and integration of cutting-edge technologies and methodologies. One potential avenue is the utilization of big data analytics and artificial intelligence to process vast amounts of data from diverse sources, enabling more accurate risk assessment and personalized loan offers. Additionally, incorporating machine learning algorithms such as reinforcement learning could optimize loan approval processes by dynamically adapting to changing market conditions and borrower profiles in real-time. Further more, the application of blockchain technology could enhance transparency and security in loan transactions, reducing the risk of fraud and improving trust between lenders and borrowers. Moreover, with the increasing popularity of peer-to-peer lending platforms and digital financial services, there is an opportunity to develop innovative algorithms and models tailored to these emerging market segments. Overall, the future of loan amount analysis and prediction holds immense potential for leveraging technology to drive efficiency, inclusivity, and sustainability in the lending ecosystem.

VI. RESULTS

1. Load Libraries

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
        from sklearn.preprocessing import LabelEncoder
        from sklearn.model_selection import train_test_split
        from sklearn import metrics
        from sklearn.linear_model import LinearRegression
```

Fig: 6.1 Loading Libraries

Data is being processed so that it can be determined how many values are missing from each column. The count of missing values present in the non-numerical attributes are processed and computed using the statistics.

2.Loading the dataset

```
In [2]: df = pd.read_csv("classified-data.csv")
In [3]: df.head()
```

Out[3]:

	Customer ID	Name	Gender	Age	Income (USD)	Income Stability	Profession	Type of Employment	Location	Loan Amount Request (USD)	...	Credit Score	No. of Defaults	Has Active Credit Card	Property ID	Property Age
0	C-36995	Frederica Shealy	F	56	1933.05	Low	Working	Sales staff	Semi-Urban	72809.58	...	809.44	0	NaN	746	1933.05
1	C-33999	America Calderone	M	32	4952.91	Low	Working	NaN	Semi-Urban	46837.47	...	780.40	0	Unpossessed	608	4952.91
2	C-3770	Rosetta Verne	F	65	988.19	High	Pensioner	NaN	Semi-Urban	45593.04	...	833.15	0	Unpossessed	546	988.19
3	C-26480	Zoe Chitty	F	65	NaN	High	Pensioner	NaN	Rural	80057.92	...	832.70	1	Unpossessed	890	NaN
4	C-23459	Afton Venema	F	31	2614.77	Low	Working	High skill tech staff	Semi-Urban	113858.89	...	745.55	1	Active	715	2614.77

Fig: 6.2 Loading Dataset

The diagram shows the first five rows of a pandas DataFrame displaying customer information related to loan applications. It includes columns like Customer ID, Name, Gender, Age, Income, Income Stability, Profession, Type of Employment, Location, Loan Amount Request, Credit Score, Number of Defaults, Active Credit Card status, Property ID, and Property Age. The data contains a mix of numerical and categorical values, with some missing entries (NaN). Each row represents a unique customer's details. The DataFrame is likely used for analyzing loan eligibility and credit risk.

```
In [5]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 24 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Customer ID                 30000 non-null  object
 1   Name                        30000 non-null  object
 2   Gender                      29947 non-null  object
 3   Age                         30000 non-null  int64
 4   Income (USD)                25424 non-null  float64
 5   Income Stability            28317 non-null  object
 6   Profession                  30000 non-null  object
 7   Type of Employment          22730 non-null  object
 8   Location                    30000 non-null  object
 9   Loan Amount Request (USD)   30000 non-null  float64
 10  Current Loan Expenses (USD) 29828 non-null  float64
 11  Expense Type 1              30000 non-null  object
 12  Expense Type 2              30000 non-null  object
 13  Dependents                  27507 non-null  float64
 14  Credit Score                28297 non-null  float64
 15  No. of Defaults             30000 non-null  int64
 16  Has Active Credit Card      28434 non-null  object
 17  Property ID                 30000 non-null  int64
 18  Property Age                25150 non-null  float64
 19  Property Type               30000 non-null  int64
 20  Property Location           29644 non-null  object
 21  Co-Applicant                30000 non-null  int64
 22  Property Price              30000 non-null  float64
 23  Loan Sanction Amount (USD)  29660 non-null  float64
dtypes: float64(8), int64(5), object(11)
memory usage: 5.5+ MB
```

Fig: 6.3 Identifying missing values

Data is being processed so that it can be determined how many values are missing from each column. The count of missing values present in the non-numerical attributes are processed and computed using the statistics.

```
In [15]: df["Gender"].value_counts()
Out[15]: M    15106
         F    14894
         Name: Gender, dtype: int64

In [16]: plt.pie(df.Gender.value_counts(), autopct="%.1f%%", radius=1.5, labels=['Male','Female'])
         plt.legend()
         plt.show()
```

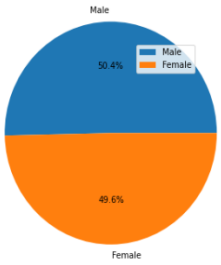

Fig: 6.4 Gender Distribution

A pie chart is a circular analytical chart, which is divided into region to symbolize numerical percentage. In this above Pie plot shows the

percentage values of male and female customers, applied for loan. The 50.4% of the male customers are applied for the loan and 49.6% of female customers are applied to the loan. According to this pie plot more number of male customers is applied to the loan.

```
In [17]: df["Income Stability"].value_counts()

Out[17]: Low     27434
         High     2566
         Name: Income Stability, dtype: int64
```

```
In [18]: fig = plt.figure(figsize=(4,4))
         sns.countplot(x = 'Income Stability',data = df)
         plt.show()
```

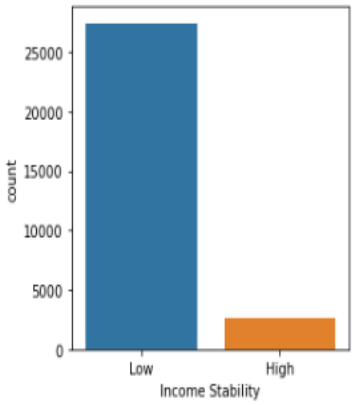

Fig: 6.5 Income Stability

In Python, we can normally plot bar charts for numerical variables. But when it comes to the case of categorical variables then we cannot normally plot the count for each category. Here comes Sea born Cat plot in the picture. It allows you to plot the count of each category for non-numerical/categorical variables. In this above count plot says that most of the customers have low-income stability. And few of customers have high income stability. This is explored in EDA.

```
In [19]: df["Profession"].value_counts()

Out[19]: Working               16926
         Commercial associate   7962
         Pensioner              2740
         State servant          2366
         Unemployed                2
         Businessman               2
         Student                   1
         Maternity leave           1
         Name: Profession, dtype: int64
```

```
In [20]: fig = plt.figure(figsize=(15,4))
         sns.countplot(x = 'Profession',data = df)
         plt.show()
```

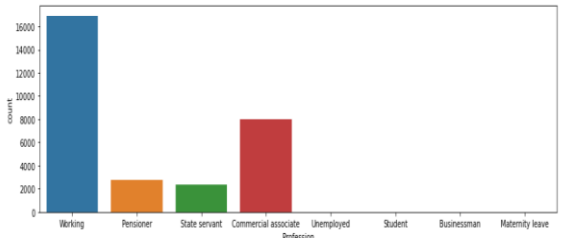

Fig: 6.6 Professions

In this plot we can says that the more number of customers are working (16926) are applied for loan. And some customers are pensioners (2740), and some customers are Commercial associates (7962), then some customers are State Servants (2366), Then finally few number of customers are Business mans, unemployed, and only 1 Student and 1Maternity leave. This can be explored in this plot.

```
In [21]: df["Type of Employment"].value_counts()

Out[21]: Laborers                 12848
         Sales staff               3736
         Core staff                3230
         Managers                  2495
         Drivers                   1606
         Accountants               1379
         High skill tech staff     1307
         Medicine staff             864
         Security staff             579
         Cooking staff              566
         Private service staff      342
         Cleaning staff             341
         Low-skill Laborers         162
         Secretaries                161
         Waiters/barmen staff       149
         Realty agents               86
         IT staff                    77
         HR staff                    72
         Name: Type of Employment, dtype: int64
```

```
In [22]: fig = plt.figure(figsize=(20,5))
         sns.countplot(x = 'Type of Employment',data = df)
         plt.show()
```

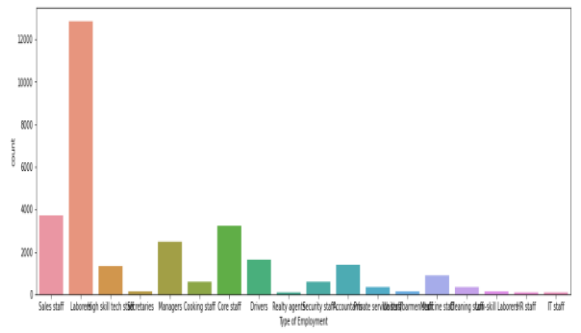

Fig: 6.7 Type of Employment

In this plot we can says that the more number of laboring customers (12848) are applied for loan. And some customers are sales staffs (3736), And some customers are core staffs (3230), then some customers are managers (2495), then finally few number of customers are drivers (1606), accountants (1379), high skill tech staff (1307), medicine staffs (864), security staffs (579), cooking staffs (566), private service staffs (342), cleaning staffs (341), low-skill laborers (162), secretaries (161), waiters (149), reality agents (86), IT staffs(77), HR staffs(72). But most numbers of customers are laborers.

```
In [23]: df["Location"].value_counts()
Out[23]: Semi-Urban    21563
         Rural          5338
         Urban          3099
         Name: Location, dtype: int64

In [24]: fig = plt.figure(figsize=(4,4))
         sns.countplot(x = 'Location',data = df)
         plt.show()
```

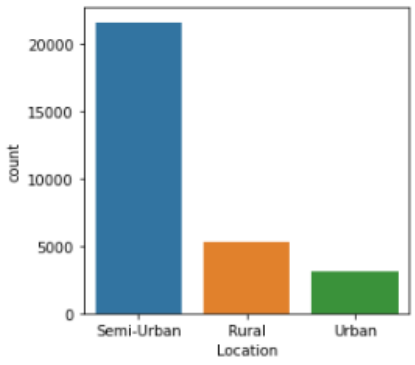

Fig: 6.8 Location

In this above count plot says that most of the customers are from semi-urban (21563). And few numbers of customers are from rural (5338) and urban (3099). This is explored in EDA.

```
In [25]: df["Expense Type 1"].value_counts()
Out[25]: N    19214
         Y    10786
         Name: Expense Type 1, dtype: int64

In [26]: fig = plt.figure(figsize=(4,4))
         sns.countplot(x = 'Expense Type 1',data = df)
         plt.show()
```

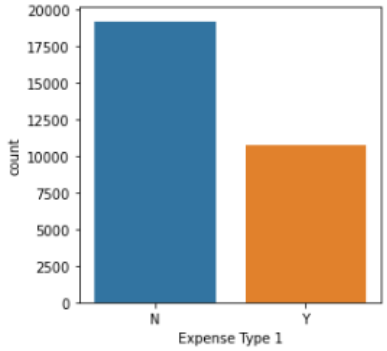

Fig: 6.9 Expense Type 1

- **Expense Type 1:** Represents a type of expense that a customer spends on (monthly) From above plot.

```
In [27]: df["Expense Type 2"].value_counts()
Out[27]: Y    20180
         N     9820
         Name: Expense Type 2, dtype: int64

In [28]: fig = plt.figure(figsize=(4,4))
         sns.countplot(x = 'Expense Type 2',data = df)
         plt.show()
```

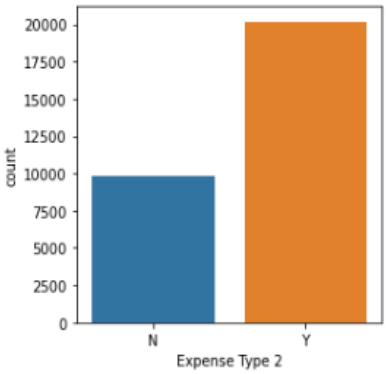

Fig: 6.10: Expense Type 2

- **Expense Type 2:** Represents a type of expense that a customer spends on (monthly).

```
In [29]: df["Has Active Credit Card"].value_counts()
Out[29]: Active         11337
         Inactive        9466
         Unpossessed     9197
         Name: Has Active Credit Card, dtype: int64

In [30]: fig = plt.figure(figsize=(4,4))
         sns.countplot(x = 'Has Active Credit Card',data = df)
         plt.show()
```

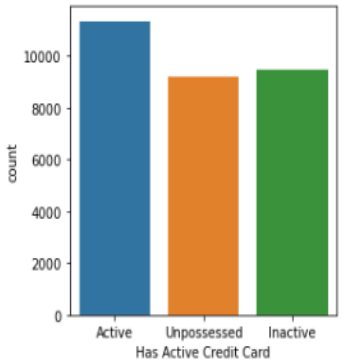

Fig 6.11: Has Active Credit Card

It has Active Credit Card it Represents if a customer has any active credit cards or not. From the above plot most of the customers have active credit card (11337). some customers have inactive credit card (9466), Further customers are having unprocessed credit card (9197).But large amount of customers are have active credit card. it is explored in this plot.

```
In [31]: df["Property Location"].value_counts()
Out[31]: Semi-Urban    10743
         Rural         10041
         Urban          9216
         Name: Property Location, dtype: int64

In [32]: fig = plt.figure(figsize=(4,4))
         sns.countplot(x = 'Property Location',data = df)
         plt.show()
```

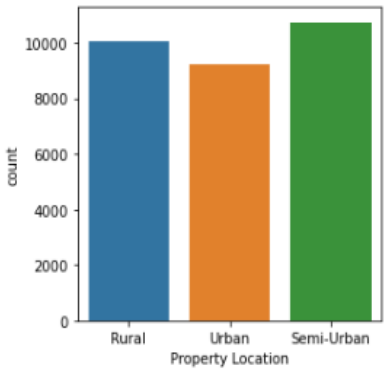

Fig 6.12: Property Location

The Property Location, it means the location of a property. In this plot the most of the customers have a property location at semi-urban side places (10743).Some customers have a property location at rural side places (10041). Few of customers are having property location at urban side places (9216). But most of customers are have property location at semi-urban side places

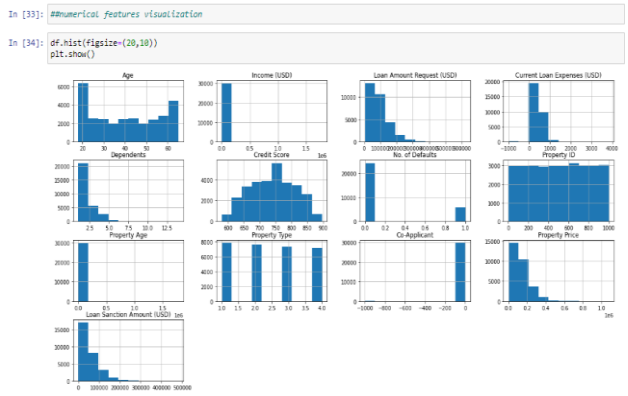

Fig: 6.13 Numerical Features Visualization

A histogram is a graphical representation of a grouped frequency distribution with continuous classes. It is an area diagram and can be defined as a set of rectangles with bases along with the intervals between class boundaries and with areas proportional to frequencies in the corresponding classes. The heights of rectangles are proportional to corresponding frequencies of similar classes and for different classes; the heights will be proportional to corresponding frequency densities. In this plot we plotted the all the numerical values that is dependents, credit score, no of defaults, property id, property age, property type, co-applicant, property price, loan sanction amount (USD).

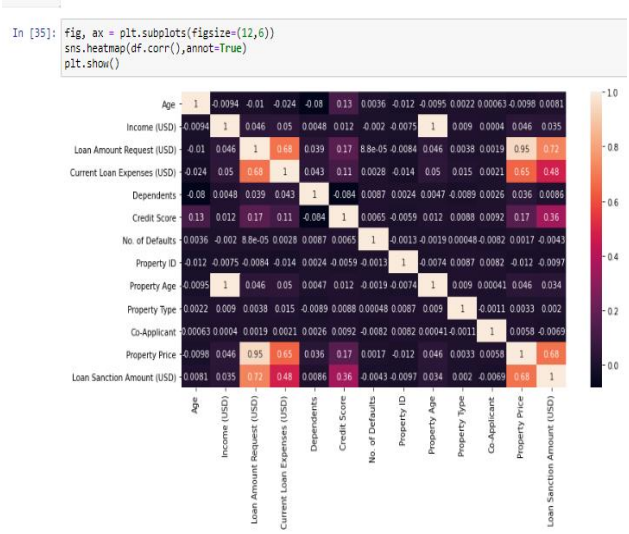

Fig: 6.14 Heat Map

Any heat map is a visual way to present the researched or predicted data of your interest. Meaning a heat map it is another form of data presentation, as are the charts, the data pies and diagrams. A heat map uses a hot-to-cold color pallet to show the high-to-low quantitative values of collected information. In this plot income (USD) and Current loan expenses (USD), and Loan amount Request(USD) and Dependents , both are correlated with a coefficient of around - 0.8 Apart from this, Credit Score and No of defaults have a correlation coefficient of around 0.6.

Serial Number	ML Algorithm Used	Accuracy Score (approx. 4 decimal places)
01	Random Forest Regressor	0.8972234930561769
02	Decision Tree Regressor	0.8938795759789925
03	Bagging Regressor	0.8913270963370292
04	Linear Regression	0.6028958929418774

Fig 5.1: Algorithm Accuracy Comparison of Loan amount Analysis And Prediction.

Random Forest Regression is known for its high accuracy in predicting continuous outcomes, such as loan amounts in a loan prediction system. Random Forest Regression is an ensemble learning method that combines multiple decision trees to make predictions.

VII. CONCLUSION

In conclusion, the loan amount analysis and prediction process using machine learning, specifically the random forest algorithm, offers a robust and reliable method for financial institutions to assess and forecast loan amounts. By leveraging historical loan data and employing and preprocessing, feature selection, and model validation techniques, the random forest model can effectively capture complex relationships within the data. This results in accurate and interpretable predictions, aiding in better decision-making and risk management. The continuous monitoring and retraining of the model ensure that it remains up-to-date with evolving data patterns, thereby maintaining its predictive power and relevance. This approach not only enhances the efficiency of loan approval processes but also contributes to improved financial stability and customer satisfaction.

VIII. FUTURE SCOPE

The future scope of loan amount analysis and prediction lies in the advancement and integration of cutting-edge technologies and methodologies. One potential avenue is the utilization of big data analytics and artificial intelligence to process vast amounts of data from diverse sources, enabling more accurate risk assessment and personalized loan offers. Additionally, incorporating machine learning algorithms such as reinforcement learning could optimize loan approval processes by dynamically adapting to changing market conditions and borrower profiles in real-time. Furthermore, the application of blockchain technology could enhance transparency and security in loan transactions, reducing the risk of fraud and improving trust between lenders and borrowers. Moreover, with the increasing popularity of peer-to-peer lending platforms and digital financial services, there is an opportunity to develop innovative algorithms and models tailored to these emerging market segments. Overall, the future of loan amount analysis and prediction holds immense potential for leveraging technology to drive efficiency, inclusivity, and sustainability in the lending ecosystem.

IX. REFERENCES

[1] Viswanatha, V, etal. (2020). Intelligent line follower robot using MSP430G2ET for industrial applications. Helix-The Scientific Explorer| Peer Reviewed Bimonthly International Journal, 10(02),23223

[2] M Meenaakumari Loan Approval Prediction Using Machine Learning Algorithms (2022)

[3] Wei Li, Shuai Ding, Yi Chen, and Shanlin Yang, Heterogeneous Ensemble for Default Prediction of Peer-to-Peer Lending in China, Key Laboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Hefei University of Technology, Hefei 2009, China

[4] Wijekoon, A., & Senevirathne, T. (2020). A Machine Learning-based Loan Approval Prediction System for Banks. In 2020 IEEE 17th International Conference on Industrial Informatics (INDIN) (pp. 739 - 744)

[5] Hasan, M. R., & Ahmed, K. (2019). Machine Learning Approach for Loan Approval Prediction. In 2019 IEEE Region 10 Symposium (TENSYMP) (pp. 994-997).

[6] Wang, S., Yu, K., Xu, Y., & Zhou, Q. (2020). Loan Application Prediction using Machine Learning. IEEE Access, 8, 34750-34761.

[7] Sathyabama Institute of Science and Technology (SIST), formerly Sathyabama University, is a private deemed university, situated at Chennai, Tamil Nadu, India. It was founded in 1987 as Sathyabama Engineering College by the late Jeppiaar, and received its university status in 2001.

[8] Mella, N. V. V. P., & Sai, R. R. LOAN APPROVAL PREDICTION

[9] Clustering Loan Applicants based on Risk Percentage using K-Means Clustering Techniques,

[10] Kumar Arun, Garg Ishan, Kaur Sanmeet, Loan Approval Prediction based on Machine Learning Approach, IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 18, Issue 3,pp. 79-81, Ver. I (May-Jun. 2016).

[11] S. Vimala, K.C. Sharmili, Prediction of Loan Risk using NB and Support Vector Machine, International Conference on Advancements in Computing Technologies (ICACT 2018), vol.4, no. 2, pp. 110-113, 2018.

[12] Loan Prediction Analysis (Classification) | Machine Learning | Python - YouTube