**REPORT ON LOAD PREDICTION MODEL  
KHUSHI LEKHWAR  
MSc, IIT KANPUR, 2024-2026**

**INTRODUCTION -**

In this project, we develop a machine learning model to predict loan approvals, addressing a crucial need in the banking and financial services industry. Financial institutions, such as banks, rely heavily on accurate assessments of loan applications to minimize risk and ensure the stability of their operations. Traditional methods of evaluating loan applications can be time-consuming, prone to human error, and biased. By leveraging machine learning techniques, we aim to streamline this process, making it more efficient and objective.

The importance of loan prediction extends beyond individual financial institutions. Accurate loan predictions can contribute to the overall health of the financial system by reducing the likelihood of defaults and non-performing assets. This, in turn, can lead to more robust economic growth and stability. Additionally, automating the loan approval process can significantly enhance the customer experience by providing quicker decisions and reducing the waiting time for applicants.

The project involves several key steps: data loading and exploration, data cleaning and preprocessing, feature engineering, data visualization, model building, and evaluation. We will be using various Python libraries, including Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn, to accomplish these tasks. Each of these libraries plays a vital role in handling, analyzing, and visualizing the data, as well as in building and evaluating the machine learning model.

This project was chosen due to its practical importance in the banking sector and its relevance in the growing field of financial analytics. By working on this project, we not only gain valuable insights into the application of machine learning in finance but also develop skills that are highly sought after in the industry. The experience gained from this project can be applied to various other domains where predictive modeling and data-driven decision-making are essential.

In summary, the loan prediction model we are developing aims to enhance the decision-making process in financial institutions, reduce risks associated with loan approvals, and improve customer satisfaction. The project showcases the power of machine learning in solving real-world problems and underscores the importance of data science in modern finance.  
  
**MODULES/LIBRARIES USED -**   
  
1. **Pandas**: Used for data manipulation and analysis, providing data structures and functions needed to manipulate structured data.

2. **NumPy**: Fundamental package for scientific computing with Python, used for handling arrays and performing numerical computations.

3. **Matplotlib**: A plotting library used for creating static, interactive, and animated visualizations in Python.

4. **Seaborn**: A statistical data visualization library based on Matplotlib, used for making complex statistical plots.

5. **Scikit-learn**: A machine learning library that includes simple and efficient tools for data mining and data analysis, used for building and evaluating machine learning models.  
  
**TOOLS USED -   
  
1. Python**: The primary programming language used for implementing the machine learning model and handling data manipulation.

**2. Jupyter Notebook**: An open-source web application used for creating and sharing documents that contain live code, equations, visualizations, and narrative text.

**3. Anaconda**: A distribution of Python and R programming languages for scientific computing, used for managing libraries and dependencies.  
  
**WORKFLOW -**1. Data Loading and Exploration

* Load the dataset
* Explore the data: check head, tail, info, shape, columns, and missing values

2. Data Cleaning

* Fill missing values for numerical columns with mean values
* Create a new feature 'TotalIncome'
* Fill missing values for categorical columns with mode values

3. Data Visualization

* Plot boxplots to check for outliers
* Plot count plots for categorical variables
* Plot a heatmap to visualize correlations

4. Feature Engineering

* Apply log transformation to numerical features to normalize them
* Drop unnecessary columns

5. Encoding Categorical Variables

* Apply label encoding to categorical features

6. Model Building and Evaluation

* Split data into training and testing sets
* Train a Logistic Regression model
* Predict on the test set
* Evaluate model performance using accuracy and precision scores

**IMPLEMENTATION CODE -**

# importing all libraries required

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

df = pd.read\_csv('C:/Users/HP/Downloads/Loan\_dataset.csv')

df.head()

df.tail()

df.info()

df.shape

df.columns

df.isnull().sum()

#Checking for Outliers:

plt.figure(figsize=(12,8))

sns.boxplot(data = df)

# Filling null values for numerical datatype

df['LoanAmount'] = df['LoanAmount'].fillna(df['LoanAmount'].mean())

df['Loan\_Amount\_Term'] = df['Loan\_Amount\_Term'].fillna(df['Loan\_Amount\_Term'].mean())

df['Credit\_History'] = df['Credit\_History'].fillna(df['Credit\_History'].mean())

df['TotalIncome'] = df['ApplicantIncome'] + df['CoapplicantIncome']

df.head()

plt.figure(figsize=(12,8))

sns.boxplot(data = df)

df['TotalIncome'] = df['TotalIncome'].fillna(df['TotalIncome'].median())

df['Gender'].mode()[0]

# Filling null values for categorical datatype

df['Gender'] = df['Gender'].fillna(df['Gender'].mode()[0])

df['Married'] = df['Married'].fillna(df['Married'].mode()[0])

df['Dependents'] = df['Dependents'].fillna(df['Dependents'].mode()[0])

df.isnull().sum()

df['Self\_Employed'] = df['Self\_Employed'].fillna(df['Self\_Employed'].mode()[0])

# Visualizing data

print('No. of people who took loan by gender')

print(df['Gender'].value\_counts())

sns.countplot(x = 'Gender', hue = 'Gender', data = df, palette = 'Set1')

print('No. of people who took loan by Marital status')

print(df['Married'].value\_counts())

sns.countplot(x = 'Married', hue = 'Married', legend = False, data = df, palette = 'Set1')

print('No. of people who took loan by Education status')

print(df['Education'].value\_counts())

sns.countplot(x = 'Education', hue = 'Education', legend = False, data = df, palette = 'Set1')

#checking correlation among various factors

df\_numeric = df.select\_dtypes(exclude=['object'])

df\_numeric

corr = df\_numeric.corr()

corr

# Visualizing correlation using heatmap

sns.heatmap(corr, annot = True, cmap = 'BuPu')

# Feature engineering: Normalizing of numerical datatype: log transformation

df['ApplicantIncomelog'] = np.log(df['ApplicantIncome'] + 1)

sns.displot(df['ApplicantIncomelog'])

df.head()

df['LoanAmountlog'] = np.log(df['LoanAmount'] + 1)

sns.displot(df['LoanAmountlog'])

df['Loan\_Amount\_Termlog'] = np.log(df['Loan\_Amount\_Term'] + 1)

sns.displot(df['Loan\_Amount\_Termlog'])

df['Total\_Incomelog'] = np.log(df['TotalIncome'] + 1)

sns.displot(df['Total\_Incomelog'])

# Dropping unnecessary columns

cols = ['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan\_Amount\_Term', 'TotalIncome']

df = df.drop(columns = cols, axis = 1)

df.head()

# Converting categorical data into numerical data: LAbel Encoding and One Hot Encoding...two ways to do it

from sklearn.preprocessing import LabelEncoder

cols = ['Gender', 'Married', 'Dependents' , 'Self\_Employed', 'Property\_Area', 'Loan\_Status' , 'Education']

le = LabelEncoder()

for col in cols:

df[col] = le.fit\_transform(df[col])

df.head()

df.dtypes

# Creating independent features and dependent feature that is loan status whether loan is approved or not

X = df.drop(columns = ['Loan\_Status', 'Loan\_ID'], axis = 1)

y = df['Loan\_Status']

# importing further necessary libraries for training testing

from sklearn.model\_selection import cross\_val\_score

from sklearn.linear\_model import LogisticRegression

# cross\_val\_score to check score of our model

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 39)

# logistic REgression model

model = LogisticRegression()

model.fit(x\_train, y\_train)

y\_pred = model.predict(x\_test)

# Accuracy: ratio of the correctly predicted values to the actual values

accuracy = accuracy\_score(y\_test, y\_pred)

accuracy\*100

precision = precision\_score(y\_test, y\_pred)

precision\*100

**APPLICATIONS –**

* **Loan Approval Systems**: Automating the loan approval process in banks and financial institutions to ensure quick and unbiased decisions.
* **Credit Risk Assessment**: Assessing the creditworthiness of loan applicants and reducing the risk of defaults.
* **Customer Segmentation**: Identifying segments of customers with high approval rates for targeted marketing and personalized offers.
* **Financial Planning**: Assisting financial advisors in providing data-driven advice to clients seeking loans.

**FUTURE ENHANCEMENTS –**

* **Feature Selection**: Implement feature selection techniques to identify and use the most significant features, improving model accuracy.
* **Advanced Algorithms**: Experiment with more advanced machine learning algorithms such as Random Forest, Gradient Boosting, or Neural Networks.
* **Hyperparameter Tuning**: Perform hyperparameter tuning to optimize the performance of the machine learning model.
* **Deployment**: Develop a web-based application to make the loan prediction model accessible to end-users, providing a user-friendly interface for inputting applicant data and receiving predictions.

**CONCLUSION -**This project demonstrates the process of building a machine learning model to predict loan approvals. The model uses historical loan application data to learn patterns that influence approval decisions. By cleaning and preprocessing the data, visualizing it for better understanding, and using logistic regression for predictions, we achieved a model with reasonable accuracy. This project highlights the importance of data-driven decision-making in the financial sector and sets the foundation for further enhancements and real-world applications.