**Jaypee Institute of Information Technology, Noida**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY



**Project Title:** Credit Card Approval Prediction Model

**Enrolment No. Name of Student**

9921103002 Rishabh Ralli

9921103009 Parth Garg

Course Name: Introduction to Big Data & Data Analytics

Course Code: 20B12CS333

Program: B. Tech. CS&E

3rd Year 5th Sem

**2023- 2024**

**TABLE OF CONTENTS**

1. Abstract
2. Introduction
   1. Problem statement
   2. Motivation
   3. Objective
   4. Contribution
3. Detailed description of the project
   1. Proposed work with tools and datasets used
   2. Workflow diagram
4. Implementation
   1. Program Code
   2. Results Analysis
5. Conclusion
6. References

**ABSTRACT**

This project introduces a Credit Card Approval system which is leveraging the power of Logistic Regression, a machine learning algorithm. The primary objective is to proficiently predict whether a given person will qualify for a credit card or not. The model is trained on a comprehensive dataset, allowing Logistic Regression to discern intricate patterns and relationships within the data, thus enabling accurate predictions. The implementation underscores the efficacy of Logistic Regression in binary classification tasks, demonstrating its practicality in detecting spam across diverse communication channels.

The project’s significance lies in the ability to accurately classify new applicants into approved and declined classes by considering the various financial parameters of the applicant. This approval system will simplify the task and help reduce the cost for analysing and classifying applicants for loan approval.

**INTRODUCTION**

**PROBLEM STATEMENT**

Traditional credit card approval processes are characterized by manual and labour-intensive tasks, involving human underwriters who manually review applicants' financial details and assess risk factors. This manual approach is time-consuming and resource-intensive.

**MOTIVATION**

The motivation behind this model stems from the inefficiencies of the existing manual credit card approval processes. The need for a more streamlined and efficient mechanism is evident, driven by the desire to enhance the speed and cost-effectiveness of credit decision-making.

**OBJECTIVES**

**Develop ml model**

Create a robust and efficient machine learning model with the primary aim of accurately classifying credit card applicants into two categories: approved and declined. This model should leverage historical data and relevant features to automate the credit card approval process

**Optimize accuracy**

Enhance classification accuracy by minimizing both false positives and false negatives, striving for a precise and reliable credit card approval prediction model.

**CONTRIBUTION**

The model contributes by providing an automated mechanism for credit card approval that significantly simplifies and accelerates the decision-making process. The contribution lies in the reduction of manual effort, time efficiency, and cost-effectiveness, offering a more scalable solution for evaluating credit risk.

**PROPOSED WORK WITH TOOLS AND DATASETS USED**

Traditional rule-based Prediction systems struggle to keep up with evolving Criteria Analysis, leading to a shift towards Machine Learning (ML).

Key advancements include:

**Data inspecting**

Data inspecting involves evaluating and understanding the characteristics of the dataset, ensuring data quality, and identifying patterns, outliers, or biases. This critical step enhances model performance by informing preprocessing decisions, mitigating biases, and ensuring the robustness of the learning process.

**Data imputation**

Data imputation in machine learning involves filling missing values within a dataset, ensuring completeness for accurate model training. It enhances the robustness of models by maintaining the integrity of the input features and improving overall predictive performance.

**Encoding categorical data**

Encoding categorical data is crucial in machine learning as it transforms non-numeric categories into numerical representations, facilitating the training of models that require numerical input and enhancing the algorithm's ability to extract meaningful patterns from the data. This process is essential for ensuring accurate predictions in various machine learning tasks.

**Feature engineering**

It involves transforming and selecting input variables to improve model performance by highlighting relevant patterns and reducing noise, contributing to more accurate predictions and better generalization.

**Criteria analysis**

Analysing involves evaluating and selecting appropriate performance metrics and evaluation criteria to assess the effectiveness of a model, ensuring it aligns with the specific goals and requirements of the task at hand. This process is crucial for making informed decisions about model performance and guiding further optimization efforts.

**About dataset**

The dataset used to train and test the ML model for credit card approval prediction is sourced from Kaggle: Credit Card Approval.

This dataset consists of the following features:

Gender, Age, Debt, Married, Bank Customer, Education Level, Ethnicity, Years Employed, Prior Default, Employed, Credit Score, Driver’s License, Citizen, Zip Code, Income, Approval Status.

**ALGORITHM**

**LOGISTIC REGRESSION**

**Introduction**

Logistic Regression is employed as the foundational algorithm for the Credit Card prediction model, serving as a binary classifier to distinguish between Approved and Declined Criteria.

**Algorithm overview**

Logistic Regression models the probability that a user belongs to a particular class (Approved or Declined) using the logistic (sigmoid) function.

Training involves optimizing a cost function to find the optimal parameters that maximize the likelihood of observed data.

Logistic Regression is computationally efficient, interpretable, and less prone to overfitting, making it suitable for this binary classification task.

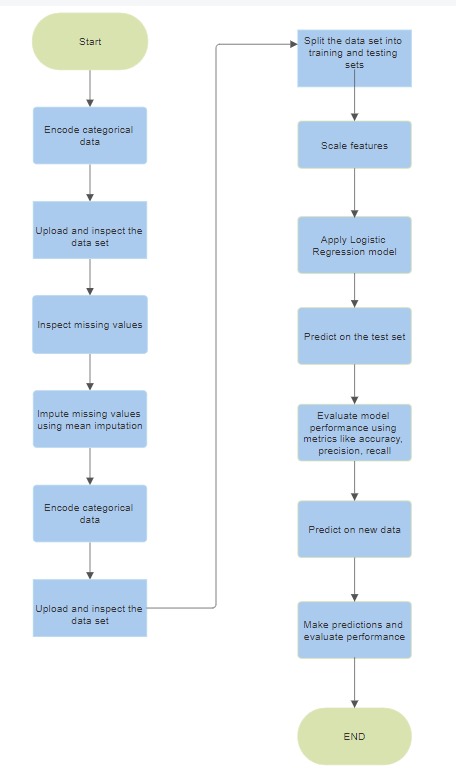
**Key features**

**Efficiency:** Logistic Regression is computationally efficient, enabling the model to handle large volumes of message data efficiently.

**Interpretability:** The model provides interpretable results, offering insights into the significance of each feature.

**Regularization:** Regularization is incorporated to prevent overfitting, contributing to the model's generalization.

**WORKFLOW DIAGRAM**



**IMPLEMENTATION**

**CODE**

**1) Including library**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

**2) Uploading and Inspecting**

applications = pd.read\_csv('cc\_approvals.data', header = None)

applications.head()

**3) Summary statics**

applications\_description = applications.describe()

print(applications\_description)

print("\n")

applications\_info = applications.info()

print(applications\_info)

print("\n")

**4) Inspect missing values in the dataset**

print(applications.isnull().values.sum())

applications = applications.replace('?', np.nan)

print('Total NaN: ' + str(applications.isnull().values.sum()))

print('NaN by column:' '\n')

print(applications.isnull().sum())

applications.tail(17)

**5) Impute the missing values with mean imputation**

applications.fillna(applications.mean(), inplace=True)

print('Total NaN: ' + str(applications.isnull().values.sum()))

applications.isnull().sum()

**6) Check if the column is of object type**

for col in applications:

    if applications[col].dtypes == 'object':

        applications = applications.fillna(applications[col].value\_counts().index[0])

print('Total missing values:' + str(applications.isnull().values.sum()))

print('Missing values in each column:')

applications.isnull().sum()

**7) Encoding the dataset**

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

for col in applications:

    if applications[col].dtypes =='object':

        le.fit(applications[col])

        applications[col]=le.transform(applications[col])

applications.head

**8) Splitting the dataset into Train and test sets**

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

applications = applications.drop([11, 13], axis=1)

print(applications.head())

applications = applications.values

X,y = applications[:,0:12] , applications[:,13]

# Split into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size=0.33,random\_state=42)

**9) Feature Scaling**

from sklearn.preprocessing import MinMaxScaler

from numpy import array\_api

scaler = MinMaxScaler(feature\_range=(0, 1))

rescaledX\_train = scaler.fit\_transform(X\_train)

rescaledX\_test = scaler.fit\_transform(X\_test)

**10) Appling LogisticRegression**

from sklearn.linear\_model import LogisticRegression

logreg = LogisticRegression()

logreg.fit(rescaledX\_train, y\_train)

**11) Predicting the Test set results**

y\_pred = logreg.predict(rescaledX\_test)

print(np.concatenate((y\_pred.reshape(len(y\_pred),1), y\_test.reshape(len(y\_test),1)),1))

**12) Predicting the Test New results**

p=scaler.fit\_transform([[1., .09482759,0.,1.,1.,0.71428571,0.11111111,0.,0.,0.,0.,1]])

z=logreg.predict(p)

**13) Making predictions and evaluating performance**

import seaborn as sns

from sklearn.metrics import confusion\_matrix,accuracy\_score

y\_pred = logreg.predict(rescaledX\_test)

print("Accuracy of logistic regression classifier: ", logreg.score(rescaledX\_test, y\_test))

# Printing the confusion matrix

print('Confusion matrix: \n ', confusion\_matrix(y\_test, y\_pred))

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

plt.figure(figsize=(8, 6))

sns.heatmap(confusion\_matrix(y\_test, y\_pred), annot=True, fmt='d', cmap='Blues', cbar=False,

            xticklabels=['Predicted Negative', 'Predicted Positive'],

            yticklabels=['True Negative', 'True Positive'])

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix')

plt.show()

**RESULT ANALYSIS**

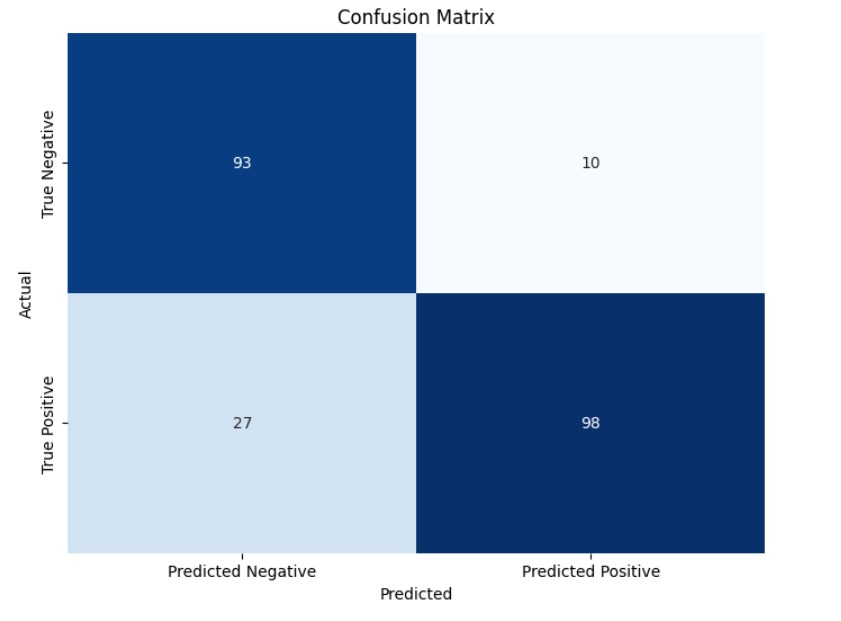
**TARGET VARIABLE**

The target variable is Approval Status.

**RESULTS**

The final model, trained on this dataset, demonstrated strong performance with 83.7% accuracy on the testing data and 87% on the training data.

**CONFUSION MATRIX**



**CONCLUSION**

This project successfully introduces a Credit Card Approval system utilizing Logistic Regression, demonstrating the effectiveness of machine learning in automating and optimizing the credit card approval process. The model's robustness is underscored by its efficient handling of diverse financial parameters, resulting in a reliable binary classification system. By achieving an 83.7% accuracy on testing data, the project showcases the potential for leveraging advanced algorithms to streamline and enhance traditional credit approval procedures. The Logistic Regression algorithm, chosen for its computational efficiency and interpretability, not only accurately predicts credit card approval but also contributes to reducing false positives and false negatives. This automation not only simplifies the approval process but also holds the promise of significantly reducing costs associated with applicant analysis and classification for loan approval. Overall, the project stands as a testament to the transformative power of machine learning in revolutionizing financial decision-making processes.

**REFERENCES**

1. GeeksforGeeks: <https://www.geeksforgeeks.org/understanding-logistic-regression/>
2. Kaggle Dataset: <https://www.kaggle.com/datasets/youssefaboelwafa/credit-card-approval>
3. Scikit-learn Documentation: <https://scikit-learn.org/stable/documentation.html>
4. Udemy Course: Machine Learning A-Z™: AI, Python https://www.udemy.com/course/machinelearning/