

**ITMD-513 Open Source Programming**

Professor: James Papadamas

Presented By: Ritu Mishra (A20469386)

Contents

[Introduction and Objectives 1](#_Toc78911103)

[Auto Loan Analysis Application 2](#_Toc78911104)

[Source Data for the application 2](#_Toc78911105)

[User Security data: 2](#_Toc78911106)

[Auto Finance data: 2](#_Toc78911107)

[Workflow Diagram 3](#_Toc78911108)

[Application functional view through screenshots 4](#_Toc78911109)

[Sign in and Error Traps 4](#_Toc78911110)

[Sign In for User 4](#_Toc78911111)

[Error message on incorrect user credentials 4](#_Toc78911112)

[Error message on blank user inputs for the user credentials 5](#_Toc78911113)

[Application Main Page and Different Analysis 6](#_Toc78911114)

[Landing Page for the Main Application screen: 6](#_Toc78911115)

[Option 1: Frequency of loan applicants by gender 7](#_Toc78911116)

[Option 2: Predictive analysis of loan approvals by credit history 8](#_Toc78911117)

[Option 3: Loan applicants by Income group 9](#_Toc78911118)

[Option 4: Loan applicants Income group by Education 10](#_Toc78911119)

[Project Outcome 11](#_Toc78911120)

[Steps to access the project 11](#_Toc78911121)

[Source Code 12](#_Toc78911122)

[File List: 12](#_Toc78911123)

[Code: 12](#_Toc78911124)

# Introduction and Objectives

Auto finance is a very big and dynamic industry in the US. The finances for new cars can get approved and rejected on many factors and thus, it becomes important for an analyst in the marketing team to understand past loans data and get some insights on what kind of market tactic should he draft for what customer base. We have prepared a web based Python application which generates some insight leveraging the auto finance data in the past and gives a view of what kind of users may get approved for auto loans as per the past data. Only the analysts who have valid user credentials will get access to the insights generated in this application.

The objectives of the application are to provide a good user interface which will be intuitive for the analyst to derive insights from auto loan data and determine what kind of user profiles are the best targets for marketing schemes. The system will be user centric and will be easy to handle but secured by user credentials to enable security. This system will use different kinds of visualizations and graph plots to generate insights and make it easy for analysis.

# Auto Loan Analysis Application

## Source Data for the application

### User Security data:

Required user credentials data for user authentication in the application is stored in SQLite database table as shown below in Figure 1:

Graphical user interface

Description automatically generated

Figure 1

### Auto Finance data:

Required auto finance data is being sourced from a CSV file which contains different data fields associated with car loans as shown in the below Figure 2:

A screenshot of a computer

Description automatically generated with low confidence

Figure 2

Data Records

## Workflow Diagram

Below Figure 3 shows the flow diagram of different building blocks in the application right from user authentication to analysis generation from the CSV file.

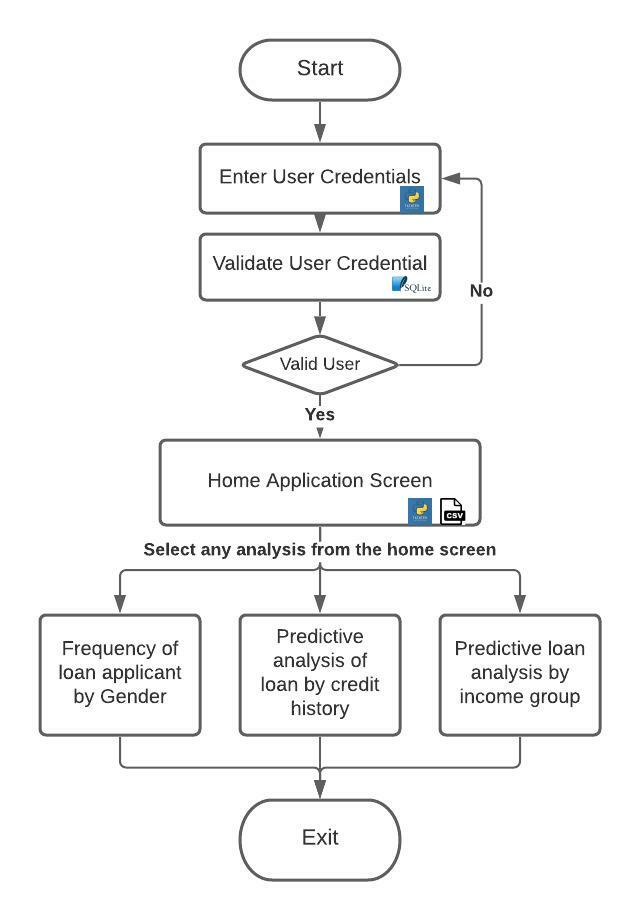


Figure 3: Workflow diagram of the application

# Application functional view through screenshots

## Sign in and Error Traps

Below Figures shows Sign-In (Login) functionalities for Application users and also demonstrates the error message in case of incorrect or blank user credential inputs in the interface.

### Sign In for User

Graphical user interface, application, website

Description automatically generated

Figure 4

### Error message on incorrect user credentials

A screenshot of a computer

Description automatically generated with medium confidence

Figure 5

### Error message on blank user inputs for the user credentials

Graphical user interface, application, website

Description automatically generated

Figure 6

## Application Main Page and Different Analysis

Below Figures shows the Auto Loan Analysis application main page and showcases different analysis performed on the data leveraging input CSV file and python scripts

### Landing Page for the Main Application screen:

Graphical user interface, text, application

Description automatically generated

Figure 7: Main Landing Page

### Option 1: Frequency of loan applicants by gender

Analysis: In this analysis we are trying to plot the number of applicants by different genders. Loans typically have discounts in the interest rate for different genders and thus this analysis is valuable

Insights: The data shows that Male gender is prominent in Auto loan applicant group for the give set of sample data. There can be special discounts planned for other genders too woo them

Below Figure 8 shows the bar graph showcasing the number of loan applicants by gender

Chart, bar chart

Description automatically generated

Figure 8

### Option 2: Predictive analysis of loan approvals by credit history

Analysis: In this analysis, we are first of all plotting the number of applicants falling under different Credit History class. We had to convert the credit history data from Y and N to 1 and 0 (0.0: N - Not available/bad, 1.0: Y - Good credit history) so that we are able to calculate **mean** (**used statistics for extra credit**) on the same and calculate the probability of getting loan approved based on credit history class.

Insights: Applicants with good credit history have ~80% probability of getting the loan while the probability of bad credit history class is less than 10%

Below Figure 9 shows visualization showcasing predictive anaylysis of loan approval by credit history

Chart, waterfall chart

Description automatically generated

Figure 9

### Option 3: Loan applicants by Income group

Analysis: In this analysis, we are creating a distribution of loan applicants by different income groups. We are using Distplot function which creates a histogram showcasing the density of values on Y axis and different groups on the X axis. The plot also gives a trend line showcasing the trend. We are also plotting a box plot to highlight the same density information in different visual

Insight: In our sample data, most of the applicants fall within the 10000 to 20000 income group. This gives the analyst the target audience group for different discounts on the interest

Below Figure 10 shows loan applicants distribution by different Income groups:

Chart, box and whisker chart

Description automatically generated

Figure 10

### Option 4: Loan applicants Income group by Education

Analysis: We had the data for different applicant income and their education so we tried to find density of applicants in different income group by their education

Insights: Density of graduates is higher in higher income groups. Analsyts can target fresh graduates for better discounts on the interest rates

Below Figure 11 shows distribution of loan applicants by different loan group and education levels:

Chart

Description automatically generated

Figure 11

# Project Outcome

The Python application developed will perform following tasks:

* Provide ability for valid users to sign in the application
* Ability to perform analysis on different auto loan data aspects
* Derive insights from the applications and download the visualization to use it for marketing strategy

# Steps to access the project

Below section shows a quick description of how to access the project, which file to execute and what login credentials can be used:

How to access the application: Please run the AutoLoanAnalysisProjectMain.py

A screenshot of a computer

Description automatically generated with medium confidence

**Admin User**

* User Name: admin
* Password: admin

**Analyst User**

* User Name: Ritu
* Password: Ritu

# Source Code

### File List:

* AutoLoanAnalysisdb.py
* AutoLoanAnalysisMainProject.py
* CarLoan.db
* Loan\_Data.csv

### Code:

AutoLoanAnalysisMainProject.py

from tkinter import Tk, Label, Button, messagebox

from tkinter import \*

from AutoLoanAnalysisdb import \*

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

import tkinter as tk

warnings.filterwarnings('ignore')

#Auto Loan Analysis App Login Page

class CarPredictionLoginPage:

    def \_\_init\_\_(self, login=Tk()):

        self.login = login

        login.protocol("WM\_DELETE\_WINDOW",self.event\_X)

        login.title("Login - Auto Loan Analysis App")

        login.geometry("450x230+450+170")

        login.configure(bg='#6E8B3D')

    # Creating widgets for the window

    #Creating layout of login form

        self.Label1=Label(login,width="300", text="Login Page", bg="#0E6655",fg="white",font=("Arial",12,"bold")).pack()

        self.username = Label(login, text="Username:")

        self.username.place(relx=0.285, rely=0.298, height=20, width=55)

        self.password = Label(login, text="Password:")

        self.password.place(relx=0.285, rely=0.468, height=20, width=55)

        # Creating Buttons for the LoginPage

        self.loginButton = Button(login, text="Login")

        self.loginButton.place(relx=0.440, rely=0.638, height=30, width=60)

        self.loginButton.configure(command=self.loginUser)

        self.loginCompleted = IntVar()

        self.exitButton = Button(login, text="Exit")  # , command=master.quit)

        self.exitButton.place(relx=0.614, rely=0.638, height=30, width=60)

        self.exitButton.configure(command=self.exitLogin)

        # Creating entry boxes

        self.usernameBox = Entry(login)

        self.usernameBox.place(relx=0.440, rely=0.298, height=20, relwidth=0.35)

        self.passwordBox = Entry(login)

        self.passwordBox.place(relx=0.440, rely=0.468, height=20, relwidth=0.35)

        self.passwordBox.configure(show="\*")

        self.passwordBox.configure(background="white")

        # Creating checkbox to view the password

        self.var = IntVar()

        self.showPassword = Checkbutton(login)

        self.showPassword.place(relx=0.285, rely=0.650, relheight=0.100, relwidth=0.125)

        self.showPassword.configure(justify='left')

        self.showPassword.configure(text='''Show''',bg='#6E8B3D')

        self.showPassword.configure(variable=self.var, command=self.cb)

    def event\_X(self):

        if messagebox.askokcancel("Exit", "Are you sure you want to exit from App"):

            exit()

    def cb(self, ):

        if self.var.get() == True:

            self.passwordBox.configure(show="")

        else:

            self.passwordBox.configure(show="\*")

    # CarLoan App Login Authentication

    def loginUser(self):

        name = self.usernameBox.get()

        password = self.passwordBox.get()

        loginCompleted = self.loginCompleted.get()

        if(name=='' or password==''):

            messagebox.showinfo("Login Warning","Please Fill the empty fields!!!")

        else:

         if (SelectDataFromUserTable(name,password)):

            messagebox.showinfo("Login page", "Login successfull!")

            # Removes the toplevel window

            self.login.destroy()

            self.loginCompleted == 1

         else:

            messagebox.showwarning("Login Failed - Acess Denied", "Username or Password incorrect!")

    def exitLogin(self):

        warningmessage = messagebox.askyesno("Exit login page", "Do you really want to exit?")

        if (warningmessage):

            exit()

    def mainloopWindow(self):

        #insertDataintoTable()

        self.login.mainloop()

loginpage = CarPredictionLoginPage()

# call the below to mainlooping the authentication window

loginpage.mainloopWindow()

#Auto Loan Analysis App Main Window

class CarLaonPredictionMainWindow:

    def \_\_init\_\_(self, main\_win=Tk()):  # This is my first change so i already initialize a Tk window inside the class

        self.main\_win = main\_win

        main\_win.title("Main Page- Auto Loan Analysis App")

        main\_win.geometry("900x500+250+130")

        main\_win.configure(bg='#838B83')

        self.lebal1 = Label(main\_win, text="Welcome to the Auto Loan Prediction Page!!..",font=("Arial",12,"bold"))

        self.lebal1.pack(padx=10, pady=10)

        self.lebal2 = Label(main\_win, text="Please Select the Below Buttons to get the Analysis",font=("Arial",10))

        self.lebal2.pack(padx=10, pady=10)

        #creating a buttons to get the prediction of the data

        self.genderbutton = Button(main\_win, text="Frequency of Loan Applicants By Gender")

        self.genderbutton.pack(side=LEFT, padx=10, pady=10)

        self.genderbutton.configure(command=self.LoanApplicantBasedOnGender)

        self.creditHistorybutton = Button(main\_win, text="Predictive Analysis of Loan By CreditHistory")

        self.creditHistorybutton.pack(side=LEFT, padx=10, pady=10)

        self.creditHistorybutton.configure(command=self.LoanAnalysisByCreditHistory)

        self.Incomebutton = Button(main\_win, text="Loan Applicants By IncomeGroup")

        self.Incomebutton.pack(side=LEFT, padx=10, pady=10)

        self.Incomebutton.configure(command=self.probabilityofApplicantIncome)

        self.Taxbutton = Button(main\_win, text="IncomeGroup By Education")

        self.Taxbutton.pack(side=LEFT, padx=10, pady=10)

        self.Taxbutton.configure(command=self.probabilityofApplicantIncomeByEducation)

        self.exitButton = Button(main\_win, text="Exit")  # , command=master.quit)

        self.exitButton.place(relx=0.500, rely=0.638, height=30, width=60)

        self.exitButton.configure(command=self.exitLogin)

    def LoanApplicantBasedOnGender(self):

        df = pd.read\_csv("Loan\_Data.csv")

        temp1 = df['Gender'].value\_counts(normalize=True)\*100

        print( 'Loan Distribution Based on gender:' )

        print (temp1)

        fig = plt.figure(figsize=(4,4))

        ax1 = fig.add\_subplot(121)

        ax1.set\_xlabel('Types of Gender',fontsize = 4, loc = "right")

        ax1.set\_ylabel('Count of Applications')

        ax1.set\_title("Frequency Table for Gender in Loan Application:")

        temp1.plot(kind='bar')

        plt.legend(loc="upper right")

        plt.show()

    def LoanAnalysisByCreditHistory(self):

        df = pd.read\_csv("Loan\_Data.csv")

        #calculating the frequency of count of application based on credit\_history

        temp1 = df['Credit\_History'].value\_counts(ascending=True)\*100

        #Loan\_status is in 'Y' and 'N' to plot easy we just map the value to 1 and 0 to calculate the mean of the column

        temp2 = df.pivot\_table(values='Loan\_Status',index=['Credit\_History'],aggfunc=lambda x: x.map({'Y':1,'N':0}).mean())

        print( 'Frequence of credit Loans for Credit History:' )

        print (temp1)

        print ('\nProbility of Loan Approval Per credit history:' )

        print (temp2)

        fig = plt.figure(figsize=(8,4))

        ax1 = fig.add\_subplot(121)

        ax1.set\_xlabel('Credit\_History of Applicants')

        ax1.set\_ylabel('Count of Applicants')

        ax1.set\_title("Applicants count of Data by Credit\_History")

        temp1.plot(kind='bar')

        plt.legend(loc="upper left")

        ax2 = fig.add\_subplot(122)

        ax2.set\_xlabel('Credit\_History')

        ax2.set\_ylabel('Probability of Loan Approval')

        ax2.set\_title("Loan Approval by credit history")

        temp2.plot(kind='bar')

        plt.legend(loc="upper left")

        plt.show()

    def probabilityofApplicantIncome(self):

        df = pd.read\_csv("Loan\_Data.csv")

        plt.figure(1)

        plt.subplot(1,2,1)

        sns.distplot(df['ApplicantIncome']);

        plt.subplot(1,2,2)

        df['ApplicantIncome'].plot.box(figsize=(8,6))

        plt.show()

    def probabilityofApplicantIncomeByEducation(self):

        df = pd.read\_csv("Loan\_Data.csv")

        df.boxplot(column='ApplicantIncome',by="Education")

        plt.suptitle(" ")

        plt.show()

    def mainloop\_window(self):

        self.main\_win.mainloop()

    def exitLogin(self):

        warningmsg = messagebox.askyesno("Exit login page", "Do you really want to exit?")

        if (warningmsg):

            exit()

# calling the main window Function

main\_win = CarLaonPredictionMainWindow()

# mainlooping the main window page

main\_win.mainloop\_window()

AutoLoanAnalysisdb.py

import  sqlite3

def createProjectTable():

    conn=sqlite3.connect("CarLoan.db")

    print("Connected to database sucessfully!!..")

    conn.execute("""

       CREATE TABLE LoanAnalysisUsers(

       ID INTEGER PRIMARY KEY AUTOINCREMENT  NOT NULL ,

       USERNAME TEXT NOT NULL,

       PASSWORD TEXT NOT NULL)

     """)

    print ("Table LoanAnalysisUsers created successfully")

def insertDataintoTable ():

      conn = sqlite3.connect('CarLoan.db')

      try:

        # cursor object

        cur = conn.cursor()

        cur.execute("INSERT INTO LoanAnalysisUsers(USERNAME,PASSWORD) VALUES ('admin', 'admin')");

        cur.execute("INSERT INTO LoanAnalysisUsers(USERNAME,PASSWORD) VALUES ('Ritu', 'Ritu')");

        cur.execute("INSERT INTO LoanAnalysisUsers(USERNAME,PASSWORD) VALUES ('Anurag', 'Anurag')");

        conn.commit()

        print ("Record is added successfully")

      except:

         print ("Error: Insert operation doesn't performed")

         conn.rollback()

      conn.close()

def SelectDataFromUserTable(name,password):

     conn = sqlite3.connect('CarLoan.db')

     #display recrod

     cursor = conn.execute('SELECT \* from LoanAnalysisUsers where USERNAME="%s" and PASSWORD="%s"'%(name,password))

     print("ID\tUSERNAME\tPASSWORD")

     row=cursor.fetchone()

     print(row)

     conn.commit()

     #print ("one record added successfully")

     return row

     conn.close()