Predicting Persona Loan Approval Using Machine Learning

Submitted by,
Sridevi Ravi,
B.SC, (CS).

1.Introduction

A loan is a sum of money that is borrowed and repaid over a period of time, typically with interest. There are various types of loans available to individuals and businesses, such as personal loans, mortgages, auto loans, student loans, business loans and many more.They are offered by banks, credit unions, and other financial institutions, and the terms of the loan, such as interest rate, repayment period, and fees, vary depending on the lender and the type of loan.

A personal loan is a type of unsecured loan that can be used for a variety of expenses such as home repairs, medical expenses, debt consolidation, and more. The loan amount, interest rate, and repayment period vary depending on the lender and the borrower's creditworthiness.To qualify for a personal loan, borrowers typically need to provide proof of income and have a good credit score.
Predicting personal loan approval using machine learning analyses a borrower's financial data and credit history to determine the likelihood of loan approval. This can help financial institutions to make more informed decisions about which loan applications to approve and which to deny.

1.1 Overview

The business requirements for a machine learning model to predict personal loan approval include the ability to accurately predict loan approval based on applicant information, Minimise the number of false positives (approved loans that default) and false negatives (rejected loans that would have been successful).Provide an explanation for the model's decision, to comply with regulations and improve transparency.

1.2 Purpose

Social Impact:- Personal loans can stimulate economic growth by providing individuals with the funds they need to make major purchases, start businesses, or invest in their education.

Business Model/Impact:- Personal loan providers may charge fees for services such as loan origination, processing, and late payments.Advertising the brand awareness and marketing to reach out to potential borrowers to generate revenue.

2.Literature Survey

A recent development of machine learning techniques and data mining has led to an interest of implementing these techniques in various fields [17]. The banking sector is no exclusion and the increasing requirements towards financial institutions to have robust risk management has led to an interest of developing current methods of risk estimation. Potentially, the implementation of machine learning techniques could lead to better quantification of the financial risks that banks are exposed to. Within the credit risk area, there has been a continuous development of the Basel accords, which provides frameworks for supervisory standards and risk management techniques as a guideline for banks to manage and quantify their risks. From Basel II, two approaches are presented for quantifying the minimum capital requirement such as the standardized approach and the internal ratings based approach (IRB) [16]. There are different risk measures banks consider in order to estimate the potential loss they may carry in future. One of these measures is the expected loss (EL) a bank would carry in case of a defaulted customer. One of the components involved in ELestimation is the probability if a certain customer will default or not. Customers in default means that they did not meet their contractual obligations and potentially might not be able to repay their loans [18]. Thus, there is an interest of acquiring a model that can predict defaulted customers. A technique that is widely used for estimating the probability of client default is Logistic Regression [19]. In this thesis, a set of machine learning methods will be investigated and studied in order to test if they can challenge the traditionally applied techniques. A prediction is a statement about what someone thinks will happen in the future. People make predictions all the time. Some are very serious and are based on scientific calculations, but many are just guesses. Prediction helps us in many things to guess what will happen after some time or after a year or after ten years. Predictive analytics is a branch of advanced analytics that uses many techniques from data mining, statistics, modeling, machine learning, and artificial intelligence to analyze current data to make predictions. “Adyan Nur Alfiyatin, Hilman Taufiq [14] and their friends work on the house price prediction. They use regression analysis and Particle Swarm Optimization (PSO) to predict house price”. One other similar work on the Mohamed El Mohadab, Belaid Bouikhalene [15] and Said Safi to predict the rank for scientific research paper using supervised learning. Kumar Arun, Garg Ishan and Kaur Sanmeet [13] work on bank loan prediction on how to bank approve a loan. They proposed a model with the help of SVM and Neural networks like machine learning algorithms. This literature review helps us carry out our work and propose a reliable bank loan prediction model.

3.Theoritical Analysis

The business requirements for a machine learning model to predict personal loan approval include the ability to accurately predict loan approval based on applicant information, Minimise the number of false positives (approved loans that default) and false negatives (rejected loans that would have been successful).Provide an explanation for the model's decision, to comply with regulations and improve transparency.

3.2 Hardware/Software Designing

The hardware required for the development of this project is:

Processor	: Inter Core TM i5-9300H
Processor speed	: 2.4GHz
RAM Size	: 8 GB DDR
System Type	: X64-based processor

Software Designing:

The software required for the development of this project is:

Deskktop GUI	: Anaconda Navigator
Operating system	: Windows 10
Front end	: HTML,CSS,JAVASCRIPT
Programming	: PYTHON
Cloud Computing Service	: IBM Cloud Services

4.Expermental Investigation Importing And Reading

The Dataset

Importing The Libraries

First step is usually importing the libraries that will be needed in the program.

Pandas: It is a python library mainly used for data manipulation.

NumPy: This python library is used for numerical analysis.

Matplotlib and Seabron: Both are the data visualization library used for plotting graph will help us for understanding the data.

Csr_matrix(): A dense matric stored in a NumPy array can be converted into a sparse matrix using the CSR representation by calling the csr_matric() function.

Train_test_split: Used for splitting data arrays into training data and for testing data.

Pickle: To serialize your machine learning algorithm and save the serialized to a file.

Reading and Datase

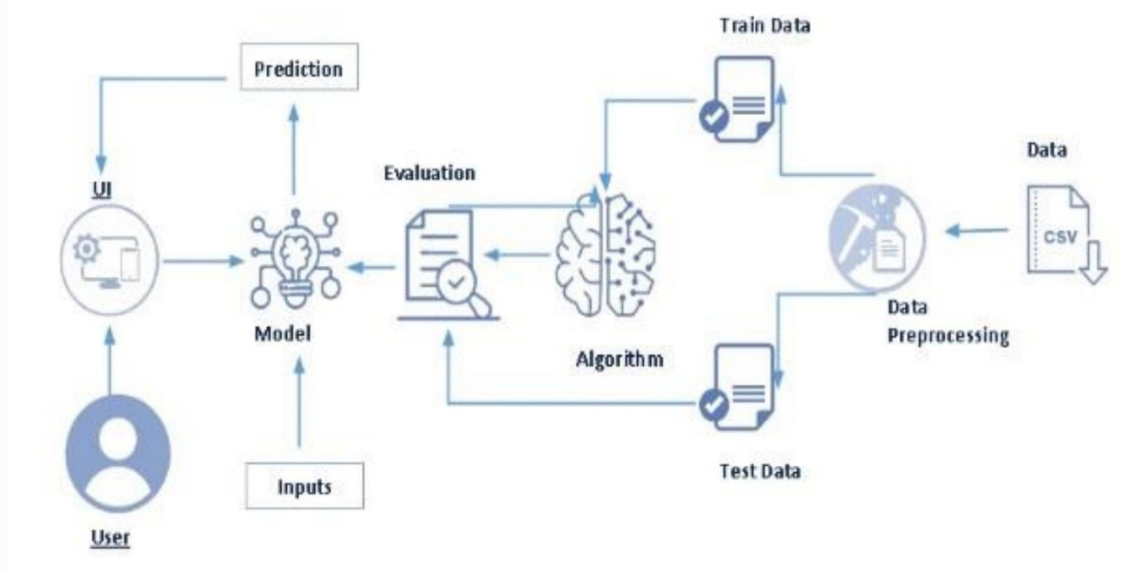
Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.
In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of the csv file.

1. Data visualization
2. Collabrative and filtering
3. Creating the Model
4. Test and save the model
5. Buil Python Code
6. Build HTML Code

7. Run the Application

We are following above sections we did and investigate it.

5.Flowchart



Project Flow:

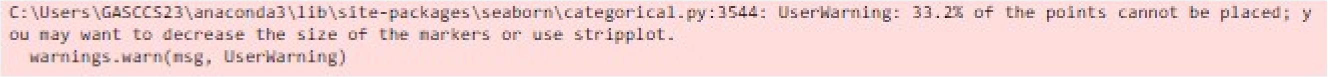
- ➢ User interacts with the UI to enter the input
- ➢ Entered input is analysed by the model which is integrated.
- ➢ Once a model analyses the uploaded inputs, the prediction is
  Showcased on the UI.

To accomplish this, we have to complete all the activities listed below,

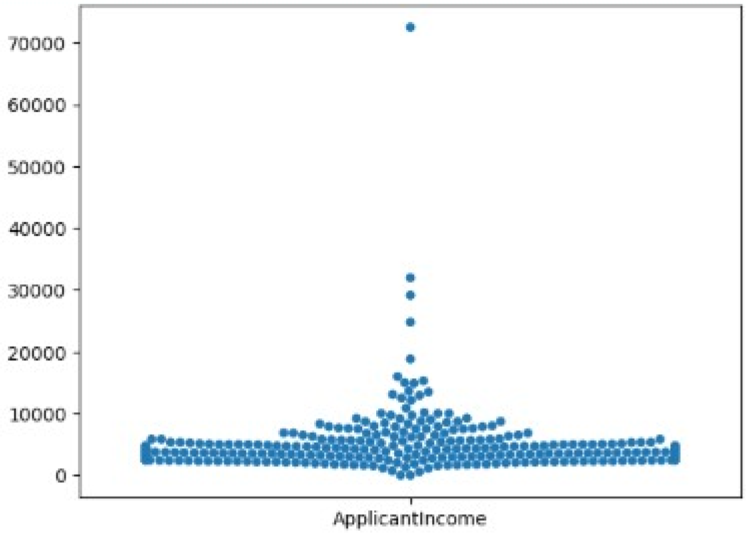
- ➢ Define problem/Problem Understanding
    - ✓ Specify the business problem
    - ✓ Business requirements
    - ✓ Literature Survey
    - ✓ Social or Business Impact.
- ➢ Data Collection & Preparation
    - ✓ Collect the dataset
    - ✓ Data Preparation.
- ➢ Exploratory Data Analysis
    - ✓ Descriptive statistical
    - ✓ Visual Analysis.
- ➢ Model Building
    - ✓ Training the model in multiple algorithms
    - ✓ Testing the model.
- ➢ Performance Testing & Hyperparameter Tuning
    - ✓ Testing model with multiple evaluation metrics
    - ✓ Comparing model accuracy before & after applying hyperparameter tuning.
- ➢ Model Deployment
    - ✓ Save the best model
    - ✓ Integrate with Web Framework.
- ➢ Project Demostration & Documentation
    - ✓ Record explanation Video for project to end solution

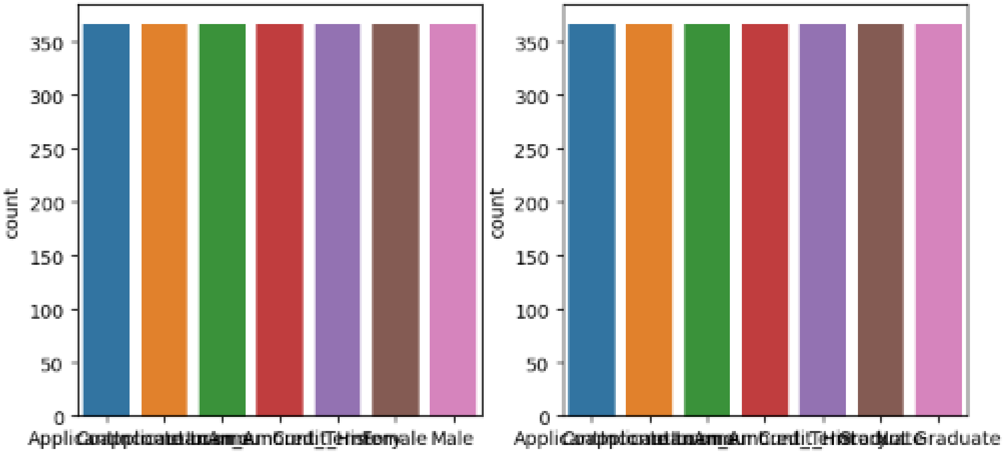
- ✓ Project Documentation-Step by step development procedure.

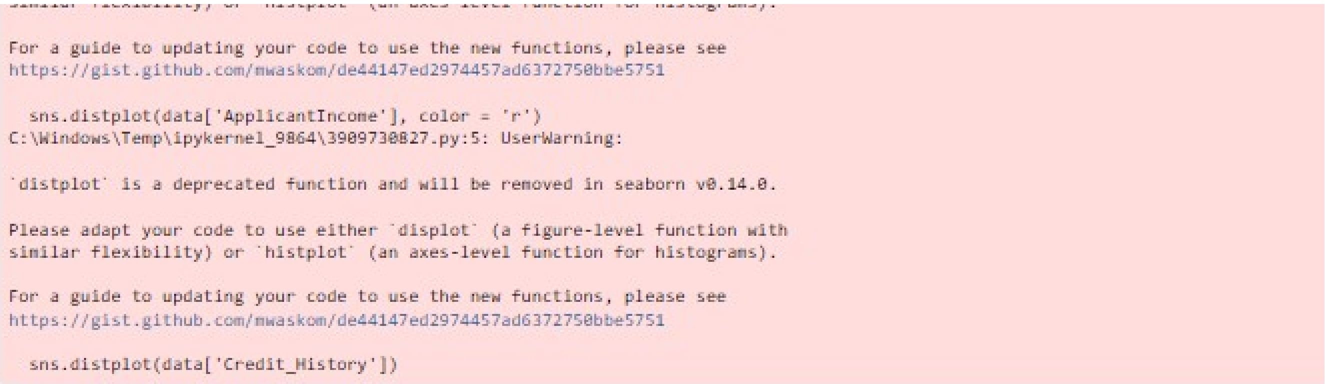
6. Result:



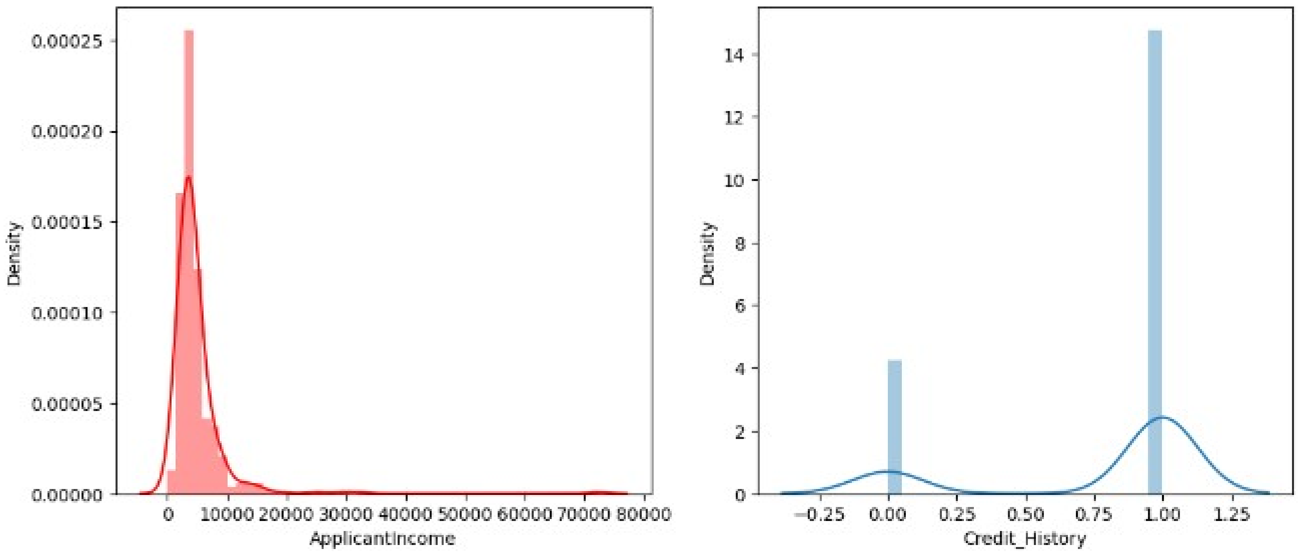
```
C:\Users\GASCCS23\anaconda3\lib\site-packages\seaborn\categorical.py:3544: UserWarning: 33.2% of the points cannot be placed; y
ou may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

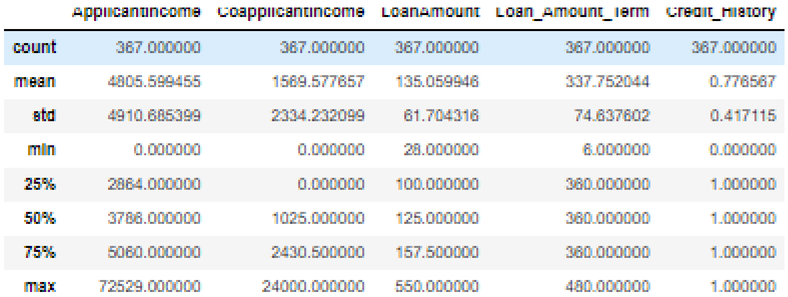






```
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['ApplicantIncome'], color = 'r')
C:\Windows\Temp\ipykernel_9864\3909730827.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['Credit_History'])
```





	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	367.000000	367.000000	367.000000	367.000000	367.000000
mean	4805.599455	1569.577657	135.059946	337.752044	0.776567
std	4910.685399	2334.232099	61.704316	74.637602	0.417115
min	0.000000	0.000000	28.000000	6.000000	0.000000
25%	2864.000000	0.000000	100.000000	360.000000	1.000000
50%	3786.000000	1025.000000	125.000000	360.000000	1.000000
75%	5060.000000	2430.500000	157.500000	360.000000	1.000000
max	72529.000000	24000.000000	550.000000	480.000000	1.000000

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 367 entries, 0 to 366
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Loan_ID            367 non-null    object
 1   Gender             367 non-null    object
 2   Married            367 non-null    object
 3   Dependents         367 non-null    object
 4   Education          367 non-null    object
 5   Self_Employed      367 non-null    object
 6   ApplicantIncome    367 non-null    int64 
 7   CoapplicantIncome  367 non-null    int64 
 8   LoanAmount         367 non-null    int64 
 9   Loan_Amount_Term   367 non-null    int64 
 10  Credit_History     367 non-null    int64 
 11  Property_Area      367 non-null    object
dtypes: int64(5), object(7)
memory usage: 34.5+ KB
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
362	False	False	False	False	False	False	False	False	False	False	False
363	False	False	False	False	False	False	False	False	False	False	False
364	False	False	False	False	False	False	False	False	False	False	False
365	False	False	False	False	False	False	False	False	False	False	False
366	False	False	False	False	False	False	False	False	False	False	False

```
In [99]: data = pd.read_csv("data set.csv")
         data
```

Out[99]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	LP001015	Male	Yes	0	Graduate	No	5720	0	110	360	1
1	LP001022	Male	Yes	1	Graduate	No	3076	1500	126	360	1
2	LP001031	Male	Yes	2	Graduate	No	5000	1800	208	360	1
3	LP001035	Male	Yes	2	Graduate	No	2340	2546	100	360	1
4	LP001051	Male	No	0	Not Graduate	No	3276	0	78	360	1
...	...	...	...	...	...	...	...	...	...	...	...
362	LP002971	Male	Yes	3+	Not Graduate	Yes	4009	1777	113	360	1
363	LP002975	Male	Yes	0	Graduate	No	4158	709	115	360	1
364	LP002980	Male	No	0	Graduate	No	3250	1993	126	360	0
365	LP002986	Male	Yes	0	Graduate	No	5000	2393	158	360	1
366	LP002989	Male	No	0	Graduate	Yes	9200	0	98	180	1

367 rows × 12 columns

7.Advantages and disadvantages

Advantages:

- The business requirements for a machine learning model to predict personal loan approval include the ability to accurately predict loan approval based on applicant information, Minimise the number of false positives (approved loans that default) and false negatives (rejected loans that would have been successful).
- Provide an explanation for the model's decision, to comply with regulations and improve transparency.

Disadvantages:

- Improper data will result in incorrect fare predictions.

8.Conclusion:

The predictive models based on Logistic Regression, Decision Tree and Random Forest, give the accuracy as 80.945%, 93.648% and 83.388% whereas the cross-validation is found to be 80.945%, 72.213% and 80.130% respectively. This shows that for the given dataset, the accuracy of model based on decision tree is highest but random forest is better at generalization even though it’s cross validation is not much higher than logistic regression.

9.Futurescope:

- **Total Income** — As discussed during bivariate analysis we will combine the Applicant Income and Co-applicant Income. If the total income is high, the chances of loan approval might also be high.

- **EMI** — EMI is the monthly amount to be paid by the applicant to repay the loan. The idea behind making this variable is that people who have high EMI's might find it difficult to pay back the loan. We can calculate the EMI by taking the ratio of the loan amount with respect to the loan amount term.

- **Balance Income** — This is the income left after the EMI has been paid. The idea behind creating this variable is that if this value is high, the chances are high that a person will repay the loan and hence increasing the chances of loan approval.

10.Bibilography:

- Vaidya and Ashlesha, Predictive and probabilistic approach using logistic regression: Application to prediction of loan approval, 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2017.
- Amin, Rafik Khairul and Yuliant Sibaroni, Implementation of decision tree using C4. 5 algorithm in decision making of loan application by debtor (Case study: Bank pasar of Yogyakarta Special Region), 2015 3rd International Conference on Information and Communication Technology (ICoICT). IEEE, 2015.
- Arora, Nisha and Pankaj Deep Kaur, A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment, Applied Soft Computing 86 (2020), 105936.

- Yang, Baoan, et al, An early warning system for loan risk assessment using artificial neural networks, Knowledge-Based Systems 14.5-6 (2001), 303-306.

Appendix:

A source code of flask:

```
From flask import Flask, render_template, request
Import numpy as np
Import pickle
App = Flask (__name__)
Model = pickle.load(open(r’rdf.pkl’,’rb’))
Scale = pickle.load(open(r’scale1.pkl’,’rb’))
@app.route(‘/’) # rendering  the html template
Def home():
Return render_template(‘home.html’)
@app.route(‘/submit’,methods = [“POST”,”GET”])# route to show the predictions in a web UI
Def submit ():
# reading the inputs given by the user
Input_feature = [int(x) for x in request.form.values()]
#input _feature = np.transpose(input_feature)
input_feature = [np.array (input_feature)]
print (input_feature)
names=[‘Gender’,’Married’,’Dependents’,’Education’,’Self_employed’,’ApplicantIncome’,’CoapplicantIncome’,’LoanAmount’,’Loan_Amount_Term’,’Credit_History’,’Property_Area’]
```

```
data = pandas.DataFrame(input_feature,columns = names)

print(data)

prediction=model.predict(data)

print (prediction)

prediction = int (prediction)

print (type(prediction))

if (prediction  == 0):

return render_template(“output.html”,result = “loan will not be approved”)

else

return render_template(“output.html”,result = “loan will  be approved”)

if __name__ ==”__main__”:

port=int(os.environ.get(‘PORT’,50000))

app.run (debug = False)
```