**Loan Status Prediction**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** |  |
|  | **INTRODUCTION** |  |
|  | * 1. General Introduction |  |
|  | * 1. Project Objectives |  |
|  | **SYSTEM** **PROPOSAL** |  |
|  | * 1. Existing System |  |
|  | * + 1. Disadvantages |  |
|  | * 1. Proposed System |  |
|  | 2.2.1 Advantages |  |
|  | * 1. Literature Survey |  |
|  | **SYSTEM DIAGRAMS** |  |
|  | * 1. Architecture Diagram |  |
|  | * 1. Flow Diagram |  |
|  | * 1. UML Diagrams |  |
|  | **IMPLEMENTATION** |  |
|  | * 1. Modules |  |
|  | * 1. Modules Description |  |
|  | **SYSTEM** **REQUIREMENTS** |  |
|  | * 1. Hardware Requirements |  |
|  | * 1. Software Requirements |  |
|  | * 1. Software Description |  |
|  | * 1. Testing of Products |  |
|  | **CONCLUSION** |  |
| **7.** | **FUTURE** **ENHANCEMENT** |  |
| **8.** | **SAMPLE** **CODING** |  |
| **9.** | **SAMPLE** **SCREENSHOT** |  |
| **10.** | **REFERENCES** |  |

**ABSTRACT**

Banking Industry always needs a more accurate predictive modeling system for many issues. Predicting credit defaulters is a difficult task for the banking industry. The loan status is one of the quality indicators of the loan. It doesn't show everything immediately, but it is a first step of the loan lending process. The loan status is used for creating a credit scoring model. Extending credit to individuals is essential for markets and society to act efficiently. Estimating the probability that an individual would default on their loan, is useful for banks to make a decision whether to approve a loan to the individual or not. In this paper, we find the accuracy of several models in R language and evaluate it to establish the finest model to forecast the finance status for an organization. We did the experiment five times on the same data set and find the experimental results that show the Tree Model for Genetic Algorithm is the best model for forecasting the finance for costumers. This system considers how automotive insurance providers incorporate machinery learning in their company, and explores how ML models can apply to insurance big data. We utilize various ML methods, such as random forest and KNN algorithm, to predict the loan status.

**CHAPTER 1**

**INTRODUCTION**

* 1. **General Introduction:**

In our banking system, banks have many products to sell but main source of income of any banks is on its credit line. So they can earn from interest of those loans which they credits. A bank’s profit or a loss depends to a large extent on loans i.e. whether the customers are paying back the loan or defaulting. By predicting the loan defaulters, the bank can reduce its Non-performing Assets. This makes the study of this phenomenon very important. Previous research in this era has shown that there are so many methods to study the problem of controlling loan default.

But as the right predictions are very important for the maximization of profits, it is essential to study the nature of the different methods and their comparison. A very important approach in predictive analytics is used to study the problem of predicting loan defaulters (i) Collection of Data, (ii) Data Cleaning and (iii) Performance Evaluation. Experimental tests found that the Naïve Bayes model has better performance than other models in terms of loan forecasting.

Loan approval is a very important process for banking organizations. The system approved or reject the loan applications. Recovery of loans is a major contributing parameter in the financial statements of a bank. It is very difficult to predict the possibility of payment of loan by the customer. Using Machine learning we predict the loan approval.

Loan Prediction is very helpful for employee of banks as well as for the applicant also. The aim of this Paper is to provide quick, immediate and easy way to choose the deserving applicants. Dream housing Finance Company deals in all loans. They have presence across all urban, semi urban and rural areas. Customer first apply for loan after that company or bank validates the customer eligibility for loan. Company or bank wants to automate the loan eligibility process (real time) based on customer details provided while filling application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and other. This project has taken the data of previous customers of various banks to whom on a set of parameters loan were approved. So the machine learning model is trained on that record to get accurate results. Our main objective of this project is to predict the safety of loan. To predict loan safety, the SVM and Naïve bayes algorithm are used. First the data is cleaned so as to avoid the missing values in the data set.

* 1. **Objectives:**

The main objective of our project is,

* To classify or to predict or to detect the loan is approved or not effectively.
* To implement the different classification algorithms for better performances.
* To enhance the overall performance for classification algorithms.

**CHAPTER 2**

**SYSTEM PROPOSAL**

* 1. **EXISTING SYSTEM:**

In existing system, Data Mining is one of the most motivating and vital area of research with the aim of extracting information from tremendous amount of accumulated data sets. In this paper a new model for classifying loan risk in banking sector by using data mining. The model has been built using data form banking sector to predict the status of loans. Three algorithms have been used to build the proposed model: j48, bayesNet and naiveBayes. By using Weka application, the model has been implemented and tested.

**2.1.1 DISADVANTAGES:**

* It doesn’t efficient for large volume of data’s
* Theoretical limits.
* Training time is high.
* The process is implemented without removing unwanted data.
  1. **PROPOSED SYSTEM:**

In this system, the loan status dataset was taken as input. The input data was taken from the dataset repository. Then, we have to implement the data pre-processing step. In this step, we have to handle the missing values for avoid wrong prediction, and to encode the label for input data. Then, we have to split the dataset into test and train. The data is splitting is based on ratio. In train, most of the data’s will be there. In test, smaller portion of the data’s will be there. Training portion is used to evaluate the model and testing portion is used to predicting the model. Then, we have to implement the classification algorithm (i.e.) machine learning. The machine learning algorithms such as KNN and Random Forest. Finally, the experimental results shows that the performance metrics such as accuracy and comparison results.

**2.2.1 ADVANTAGES:**

* It is efficient for large number of datasets.
* Time consumption is low.
* The process is implemented with removing unwanted data.
* Prediction is accurate.

**2.3 LITERATURE SURVEY:**

**2.3.1 Prediction for Loan Approval using Machine Learning Algorithm, 2021**

**Author***:* Ashwini S. Kadam1, Shraddha R. Nikam2, Ankita A. Aher3, Gayatri V. Shelke4, Amar S. Chandgude 5

**Methodology:**

In our banking system, banks have many products to sell but main source of income of any banks is on its credit line. So they can earn from interest of those loans which they credits. A bank’s profit or a loss depends to a large extent on loans i.e. whether the customers are paying back the loan or defaulting. By predicting the loan defaulters, the bank can reduce its Non-performing Assets. This makes the study of this phenomenon very important. Previous research in this era has shown that there are so many methods to study the problem of controlling loan default. But as the right predictions are very important for the maximization of profits, it is essential to study the nature of the different methods and their comparison. A very important approach in predictive analytics is used to study the problem of predicting loan defaulters (i) Collection of Data, (ii) Data Cleaning and (iii) Performance Evaluation. Experimental tests found that the Naïve Bayes model has better performance than other models in terms of loan forecasting.

**Advantage:**

* Training time is low.

**Disadvantage:**

* It is not efficient for large number of data’s.

**2.3.2 Accuracy Prediction for Loan Risk Using Machine Learning Models, 2021**

**Author**: Anchal Goyal [1], Ranpreet Kaur

**Methodology:**

Extending credit to individuals is essential for markets and society to act efficiently. Estimating the probability that an individual would default on their loan, is useful for banks to make a decision whether to approve a loan to the individual or not. In this paper, we find the accuracy of several models in R language and evaluate it to establish the finest model to forecast the finance status for an organization. We did the experiment five times on the same data set and find the experimental results that show the Tree Model for Genetic Algorithm is the best model for forecasting the finance for costumers

**Advantage*:***

* RPA can complete tasks more quickly than humans, and it's able to do so at a lower cost.

**Disadvantage*:***

* Training time is high.

**2.3.3 Prediction of loan status in commercial bank using machine learning classifier, 2021**

**Author:** G. Arutjothi; C. Senthamarai

**Methodology:**

Banking Industry always needs a more accurate predictive modeling system for many issues. Predicting credit defaulters is a difficult task for the banking industry. The loan status is one of the quality indicators of the loan. It doesn't show everything immediately, but it is a first step of the loan lending process. The loan status is used for creating a credit scoring model. The credit scoring model is used for accurate analysis of credit data to find defaulters and valid customers. The objective of this paper is to create a credit scoring model for credit data. Various machine learning techniques are used to develop the financial credit scoring model. In this paper, we propose a machine learning classifier based analysis model for credit data. We use the combination of Min-Max normalization and K-Nearest Neighbor (K-NN) classifier. The objective is implemented using the software package R tool. This proposed model provides the important information with the highest accuracy. It is used to predict the loan status in commercial banks using machine learning classifier.

**Advantage*:***

* Training time is low.
* Better performance.

**Disadvantage**:

* It creates a new instance by appropriately combining existing instances, thus making it possible to avoid the disadvantage of over fitting to a certain degree.

**2.3.4 Loan default prediction using decision trees and random forest: A comparative study, 2021**

**Author:** Mehul Madaan1, \*, Aniket Kumar1, Chirag Keshri1, Rachna Jain2and Preeti Nagrath2

**Methodology:**

With the improving banking sector in recent times and the increasing trend of taking loans, a large population applies for bank loans. But one of the major problem banking sectors face in this ever-changing economy is the increasing rate of loan defaults, and the banking authorities are finding it more difficult to correctly assess loan requests and tackle the risks of people defaulting on loans. The two most critical questions in the banking industry are (i) How risky is the borrower? and (ii) Given the borrower’s risk, should we lend him/her? In light of the given problems, this paper proposes two machine learning models to predict whether an individual should be given a loan by assessing certain attributes and therefore help the banking authorities by easing their process of selecting suitable people from a given list of candidates who applied for a loan.

**Advantage:**

* We can note that the Sensitivity for all ML models with the unbalanced data is lower than the Sensitivity for balanced data created by different resampling methods

**Disadvantage**:

* It creates a new instance by appropriately combining existing instances, thus making it possible to avoid the disadvantage of over fitting to a certain degree.

**CHAPTER 3**

**SYSTEM DIAGRAMS**

**3.1 SYSTEM ARCHITECTURE:**

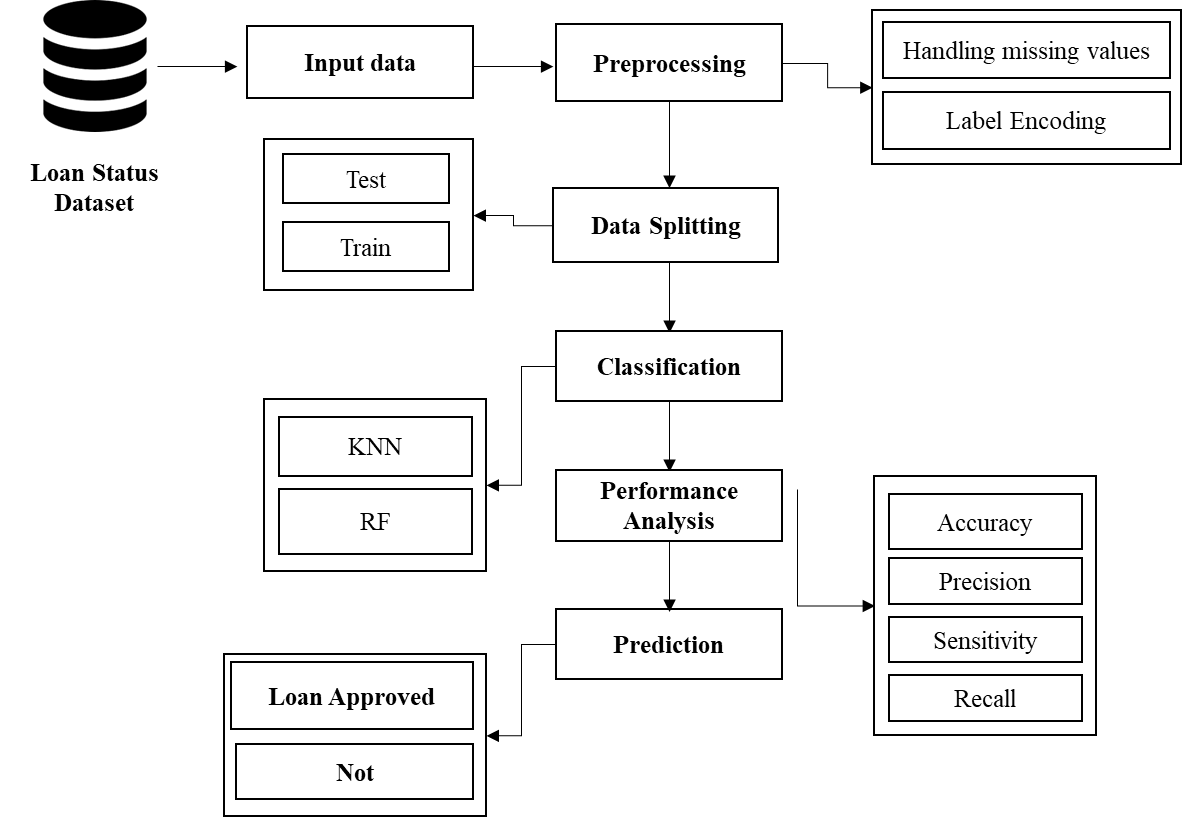
****

FIGURE 3.1: SYSTEM ARCHITECTURE

**3.2 FLOW DIAGRAM**

Input Data

Preprocessing

Data Splitting

Classification

Performance analysis

FIGURE 3.2: FLOW DIAGRAM

**3.3 UML DIAGRAMS:**

**3.3.1 USE CASE DIAGRAM:**

System

User

FIGURE 3.3.1: USE CASE DIAGRAM

**3.3.2 ACTIVITY DIAGRAM:**

Input Data

Preprocessing

Data splitting

Performance metrics

Classification

FIGURE 3.3.2: ACTIVITY DIAGRAM

**3.3.3 SEQUENCE DIAGRAM:**

**Input Data**

**Preprocessing**

**Data splitting**

**Classification**

Select data

Missing value

Test and Train

Load data

Correlation

RF and KNN

FIGURE 3.3.3: SEQUENCE DIAGRAM

**3.3.4 ER DIAGRAM:**

**Data selection**

**Preprocessing**

**Classification**

**Prediction**

FIGURE 3.3.4: ER DIAGRAM

**3.3.5 CLASS DIAGRAM:**

Select data ()

Load data ()

View data ()

**Input Data**

Test ()

**Data Splitting**

Prediction ()

Accuracy ()

Precision ()

Recall()

**Performance**

**Preprocessing**

Missing values ()

Label encode ()

RF ()

KNN ()

**Classification**

Train ()

Correlation ()

FIGURE 3.3.5: CLASS DIAGRAM

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 MODULES:**

* Data selection
* Preprocessing
* Data splitting
* Classification
* Result generation

**4.2 MODULES DESCRIPTION:**

**4.2.1: DATA SELECTION:**

* The input data was collected from dataset repository.
* In our process, loan status dataset is used.
* The data selection is the process of predicting the loan status is approved or not.
* The input dataset was taken from dataset repository such as UCI repository.
* In python, with the help of panda’s package, we can read or load our input dataset.
* Our dataset is in the format is ‘.csv’

**4.2.2: DATA PREPROCESSING:**

* Data pre-processing is the process of removing the unwanted data from the dataset.
* Pre-processing data transformation operations are used to transform the dataset into a structure suitable for machine learning.
* This step also includes cleaning the dataset by removing irrelevant or corrupted data that can affect the accuracy of the dataset, which makes it more efficient.
* Missing data removal
* Encoding Categorical data
* Missing data removal: In this process, the null values such as missing values and Nan values are replaced by 0.
* Missing and duplicate values were removed and data was cleaned of any abnormalities.
* Encoding Categorical data: That categorical data is defined as variables with a finite set of label values.
* That most machine learning algorithms require numerical input and output variables.

**4.2.3: DATA SPLITTING:**

* During the machine learning process, data are needed so that learning can take place.
* In addition to the data required for training, test data are needed to evaluate the performance of the algorithm in order to see how well it works.
* In our process, we considered 80% of the input dataset to be the training data and the remaining 20% to be the testing data.
* Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.
* One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
* Separating data into training and testing sets is an important part of evaluating data mining models.
* Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.

**4.2.5: CLASSIFICATION:**

* In our process, we have to implement the different classification algorithm such as Random forest and KNN.
* **Random forest** is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.
* **The k-nearest neighbors** algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

**4.2.6: RESULT GENERATION:**

The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like,

* **Accuracy:**

Accuracy of classifier refers to the ability of classifier. It predicts the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.

AC= (TP+TN)/ (TP+TN+FP+FN)

* **Precision**

Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives.

Precision=TP/ (TP+FP)

* **Recall**

Recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is called sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query.

Recall=TP/ (TP+FN)

**CHAPTER 5**

**SYSTEM REQUIREMENTS**

**5.1 HARDWARE REQUIREMENTS:**

* System : Pentium IV 2.4 GHz
* Hard Disk : 200 GB
* Mouse : Logitech.
* Keyboard : 110 keys enhanced
* Ram : 4GB

**5.2 SOFTWARE REQUIREMENTS:**

* O/S : Windows 7.
* Language : Python
* Front End : Anaconda Navigator – Spyder

**5.3 SOFTWARE DESCRIPTION:**

**5.3.1 Python**

Python is one of those rare languages which can claim to be both *simple* and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in. The official introduction to Python is Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. I will discuss most of these features in more detail in the next section.

## **5.3.2 Features of Python**

### **Simple**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

### **Easy to Learn**

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

### **Free and Open Source**

Python is an example of a FLOSS (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

### **High-level Language**

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

### **Portable**

Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC!

You can even use a platform like [Kivy](http://kivy.org) to create games for your computer and for iPhone, iPad, and Android.

### **Interpreted**

This requires a bit of explanation.

A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it.

Python, on the other hand, does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

### **Object Oriented**

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

### **Extensible**

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use it from your Python program.

### **Embeddable**

You can embed Python within your C/C++ programs to give scripting capabilities for your program's users.

### **Extensive Libraries**

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the Batteries Included philosophy of Python.

Besides the standard library, there are various other high-quality libraries which you can find at the Python Package Index.

**5.4 TESTING PRODUCTS:**

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. . A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

**5.4.1 UNIT TESTING:**

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as ‘module testing’.

The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

**5.4.2 INTEGRATION TESTING:**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

i) Top-down integration testing. ii) Bottom-up integration testing.

**5.4.3 TESTING TECHNIQUES/STRATEGIES:**

* **WHITE BOX TESTING:**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we

Derived test cases that guarantee that all independent paths within a module have been exercised at least once.

* **BLACK BOX TESTING:**

1. Black box testing is done to find incorrect or missing function
2. Interface error
3. Errors in external database access
4. Performance errors.
5. Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’. It tests the external behaviour of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**5.4.4 SOFTWARE TESTING STRATEGIES**

**VALIDATION TESTING:**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many,

But a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer

**USER ACCEPTANCE TESTING:**

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

**OUTPUT TESTING**:

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

**CHAPTER 6**

**CONCLUSION**

We conclude that, the auto insurance claim dataset was taken as input. The input dataset was mentioned in our research paper. We are implemented the classification algorithms (i.e) machine learning algorithms. Then, machine learning algorithms such as random forest and KNN. Finally, the result shows that the accuracy for above mentioned algorithm and estimated the performances metrics such as accuracy for both algorithms and comparison graph.

**CHAPTER 7**

**FUTURE ENHANCEMENT**

Future work may be done in the next directions: Using hybrid classifiers to improve comparison and performance. Furthermore, feature selection approaches may be used to enhance model results and gain a deeper understanding of the important features. It will also be worthwhile to conduct this research for another insurance branch, whether to predict claim occurrences or to predict fraud because these kinds of data always are very heavily unbalanced.

**CHAPTER 8**

**SAMPLE CODING**

#======================= IMPORT PACKAGES =============================

import pandas as pd

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

import warnings

warnings.filterwarnings('ignore')

import matplotlib.pyplot as plt

from sklearn import preprocessing

#===================== DATA SELECTION ==============================

#=== READ A DATASET ====

data\_frame=pd.read\_csv("Dataset.csv")

print("----------------------------------")

print(" 1.Data Selection ")

print("----------------------------------")

print()

print(data\_frame.head(20))

#===================== DATA PREPROCESSING ==============================

#=== CHECK MISSING VALUES ===

print("=====================================================")

print(" 2.Preprocessing ")

print("=====================================================")

print()

print("--------------------------------------------")

print(" Before Checking missing values ")

print("--------------------------------------------")

print()

print(data\_frame.isnull().sum())

print()

print("--------------------------------------------")

print(" After Checking missing values ")

print("--------------------------------------------")

print()

data\_frame=data\_frame.fillna(0)

print(data\_frame.isnull().sum())

#=== LABEL ENCODING ===

label\_encoder = preprocessing.LabelEncoder()

print("---------------------------------")

print(" Before label encoding ")

print("---------------------------------")

print()

print(data\_frame['Loan\_ID'].head(10))

data\_frame['Loan\_ID']=label\_encoder.fit\_transform(data\_frame['Loan\_ID'])

data\_frame['Gender']=label\_encoder.fit\_transform(data\_frame['Gender'].astype(str))

data\_frame['Married']=label\_encoder.fit\_transform(data\_frame['Married'].astype(str))

data\_frame['Education']=label\_encoder.fit\_transform(data\_frame['Education'].astype(str))

data\_frame['Dependents']=label\_encoder.fit\_transform(data\_frame['Dependents'].astype(str))

data\_frame['Self\_Employed']=label\_encoder.fit\_transform(data\_frame['Self\_Employed'].astype(str))

data\_frame['Property\_Area']=label\_encoder.fit\_transform(data\_frame['Property\_Area'].astype(str))

data\_frame['Loan\_Status']=label\_encoder.fit\_transform(data\_frame['Loan\_Status'])

print("-------------------------------------------")

print(" After label Encoding ")

print("------------------------------------------")

print()

print(data\_frame['Loan\_ID'].head(20))

#============================= DATA SPILLTING =========================

X = data\_frame.drop("Loan\_Status",axis=1)

Y = data\_frame["Loan\_Status"]

print("----------------------------------------")

print("DATA SPLITTING")

print("------------------------------------")

print()

x\_train,x\_test,y\_train,y\_test=train\_test\_split(X,Y,test\_size=0.3,random\_state=1)

print()

print("Total Number Of data = ", len(X))

print()

print("Total Number Of Test data = ", len(x\_test))

print()

print("Total Number Of Train data = ", len(x\_train))

print()

#============================= CLASSIFICATION =========================

# === RANDOM FOREST =====

from sklearn.ensemble import RandomForestClassifier

regressor = RandomForestClassifier(n\_estimators = 10)

# fit the regressor with x and y data

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

Y\_pred = regressor.predict(x\_train)

from sklearn import metrics

Accuracy\_rf=metrics.accuracy\_score(y\_train,Y\_pred)\*100

print("----------------------------------------")

print("RANDOM FORES --> RF")

print("------------------------------------")

print()

print("1. Accuracy =",Accuracy\_rf )

print()

print(metrics.classification\_report(y\_train,Y\_pred))

# === K NEAREST NEIGHBOUR =====

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()

# fit the regressor with x and y data

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

Y\_pred\_knn = knn.predict(x\_train)

from sklearn import metrics

Accuracy\_knn=metrics.accuracy\_score(y\_train,Y\_pred\_knn)\*100

print("----------------------------------------")

print("K NEAREST NEIGHBOUR --> KNN")

print("------------------------------------")

print()

print("1. Accuracy =",Accuracy\_knn )

print()

print(metrics.classification\_report(y\_train,Y\_pred\_knn))

# ============== PREDICTION =====================

print("----------------------------------------")

print("PREDICTION ")

print("------------------------------------")

print()

for i in range(0,5):

if Y\_pred[i]==0:

print("-------------------------")

print([i],"Loan is NOT-APPROVED")

print("-------------------------")

else:

print("-------------------------")

print([i],"Loan is APPROVED")

print("-------------------------")

# ===== COMPARISON =====

vals=[Accuracy\_knn,Accuracy\_rf]

inds=range(len(vals))

labels=["KNN","RF"]

fig,ax = plt.subplots()

rects = ax.bar(inds, vals)

ax.set\_xticks([ind for ind in inds])

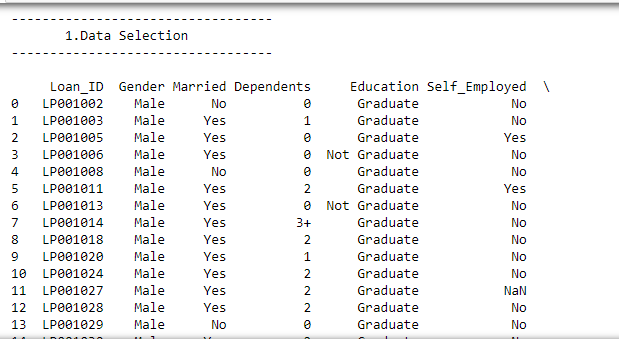
ax.set\_xticklabels(labels)

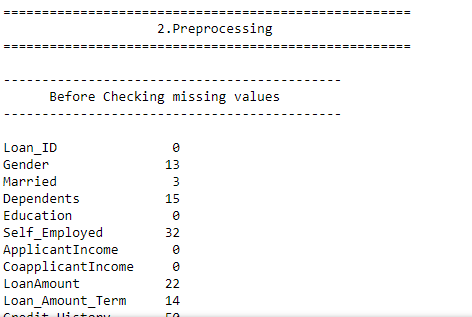
plt.savefig("Performance")

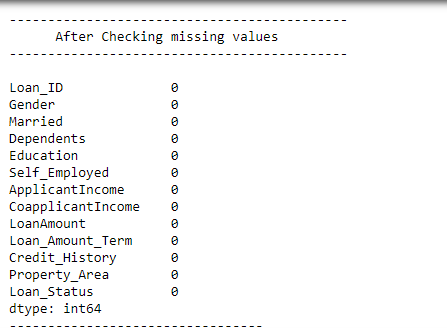
plt.show()

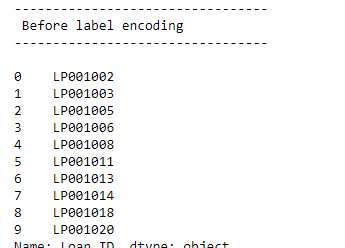
**CHAPTER 9**

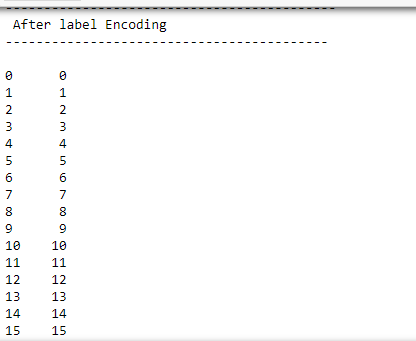
**SAMPLE SCREENSHOTS**

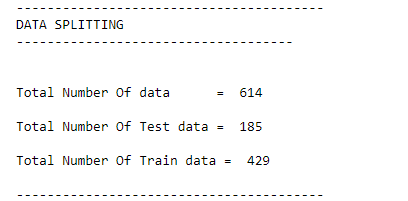


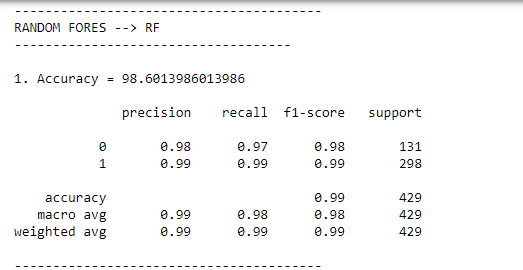


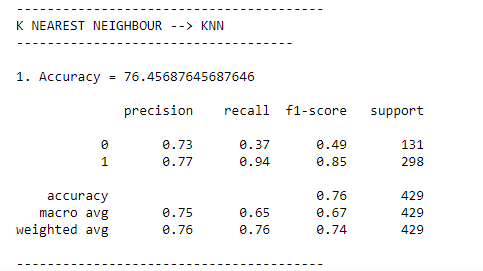


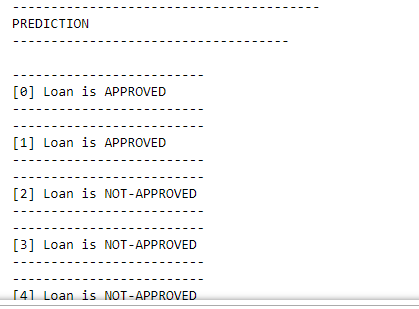












**CHAPTER 10**

**REFERENCES**

[1] Ogawa, Ms Sumiko, et al. Financial Interconnectedness and Financial Sector Reforms in the Caribbean. No. 13-175. International Monetary Fund, 2013.

[2] Strahan, Philip E. "Borrower risk and the price and nonprice terms of bank loans." FRB of New York Staff Report 90 (1999).

[3] Tomar, Divya, and Sonali Agarwal. "A survey on Data Mining approaches for Healthcare." International Journal of Bio-Science and Bio-Technology 5.5 (2013): 241-266.

[4] Pulakkazhy, Sreekumar, and R. V. S. Balan. "Data mining in banking and its applications-a review." Journal of computer science 9.10 (2013): 1252.

[5] Tomar, Divya, and Sonali Agarwal. "A survey on Data Mining approaches for Healthcare." International Journal of Bio-Science and Bio-Technology 5.5 (2013): 241-266.

[6] Sharma, Poonam, and Gudla Balakrishna. "PrefixSpan: Mining Sequential Patterns by PrefixProjected Pattern." International Journal of Computer Science and Engineering Survey 2.4 (2011): 111.

[7] Chitra, K., and B. Subashini. "Data Mining Techniques and its Applications in Banking Sector." International Journal of Emerging Technology and Advanced Engineering 3.8 (2013): 219-226.

[8] Zurada, Jozef, and Martin Zurada. "How Secure Are “Good Loans”: Validating Loan-Granting Decisions And Predicting Default Rates On Consumer Loans."Review of Business Information Systems (RBIS) 6.3 (2011): 65-84. [

9] İkizler, Nazlı, and H. Altay Guvenir. "Mining interesting rules in bank loans data." Proceedings of the Tenth Turkish Symposium on Artificial Intelligence and Neural Networks. 2001.

[10] Abhijit A. Sawant and P. M. Chawan, "Comparison of Data Mining Techniques used for Financial Data Analysis," International Journal of Emerging Technology and Advanced Engineering, june 2013. [11] Sudhakar M and Dr. C. V. Krishna Reddy, "CREDIT EVALUATION MODEL OF LOAN PROPOSALS FOR BANKS USING DATA MINING," International Journal of Latest Research in Science and Technology, pp. 126-131, july 2014.

[12] r.R.Mahammad Shafi, A Tool for Enhancing Business Process in Banking Sector, 3rd ed.: International Journal of Scientific & Engineering Research, 2012.

[13] Islam, Md Samsul, Lin Zhou, and Fei Li. "Application of artificial intelligence (artificial neural network) to assess credit risk: a predictive model for credit card scoring." MSc, School of Management Blekinge Institute of Technology(2009).

[14] Vikas Jayasree and Rethnamoney Vijayalakshmi Siva Balan, A REVIEW ON DATA MINING IN BANKING SECTOR.: American Journal of Applied Sciences, 2013.

[15] Abhijit A. Sawant and P. M. Chawan, Comparison of Data Mining Techniques used for Financial Data Analysis.: International Journal of Emerging Technology and Advanced Engineering, june 2013.

[17] Moin, Kazi Imran, and Dr Qazi Baseer Ahmed. "Use of data mining in banking."International Journal of Engineering Research and Applications Vol 2 (2012): 738-742.

[18] Chu, Wesley, and Tsau Young Lin. Foundations and advances in data mining. Vol. 180. Springer Science & Business Media, 2005.