FINANCIAL RISK MANAGEMENT

1. INTRODUCTION

1.1 OVERVIEW

Financial risk management is nothing but calculating a person's financial risk and making their
financial status is more manageable by using financial instruments. . Financial risks can be foreign exchange risk, business risk, legal risk, etc. Financial risk management can be either qualitative or quantitative .Based on these, it helps people to manage their costly exposures to risk

1.2 PURPOSE

The main purpose of this Financial risk management is ,to be
able to identify one's financial status and know the risk one might face regarding his finance which is based on various things like gender, age , the type of housing, profession , savings, checking accounts, etc.
All these details, when considered give the probability of the risk a person might face regarding his or her financial status.

2. LITERATURE SURVEY

1.Risk is inherent in every business and every organization has to manage it according to its size and nature of operation because without it no organization can survive in long run(Ajay Shukla(2014))

2.All of these approaches are built on financial market data, on the basis of which they lead to the estimation of appropriate quantiles of the estimated loss probability distribution of a financial institution, conditional on a crash event on the financial market (Huang X(2012)

3. explicitly geared towards estimation of the interrelationships among all institutions, is based on network models, and has been proposed in(Getmansky M(2012))

4.Different features have different bias which may be stronger when the probability of multiple failures are to be estimated, as it occurs in systemic risk(Hirsch JE(2005))

5.Advanced information technologies can help in identification, measurement and monitoring of financial risk. With the advent of sophisticated communication and database technology(Arun Bansal(2011))

2.1 EXISTING PROBLEM

People these days face a lot of financial crises. These can be either in the form of healthcare costs, debts or loans from banks or individuals, college or academic expenses, renting a house , or high costs of living . Many don't know how to manage all these finance related problems and hence face financial risks be it in business, marketing, education, etc. Learning to manage your money is like overcoming a big hurdle in your life. The younger generations also face these risks in forms of student loans, or taking other loans when just beginning their career. People don't realize and don't come to a conclusion about their financial status and hence fall into the risk.

2.2 PROPOSED SOLUTION

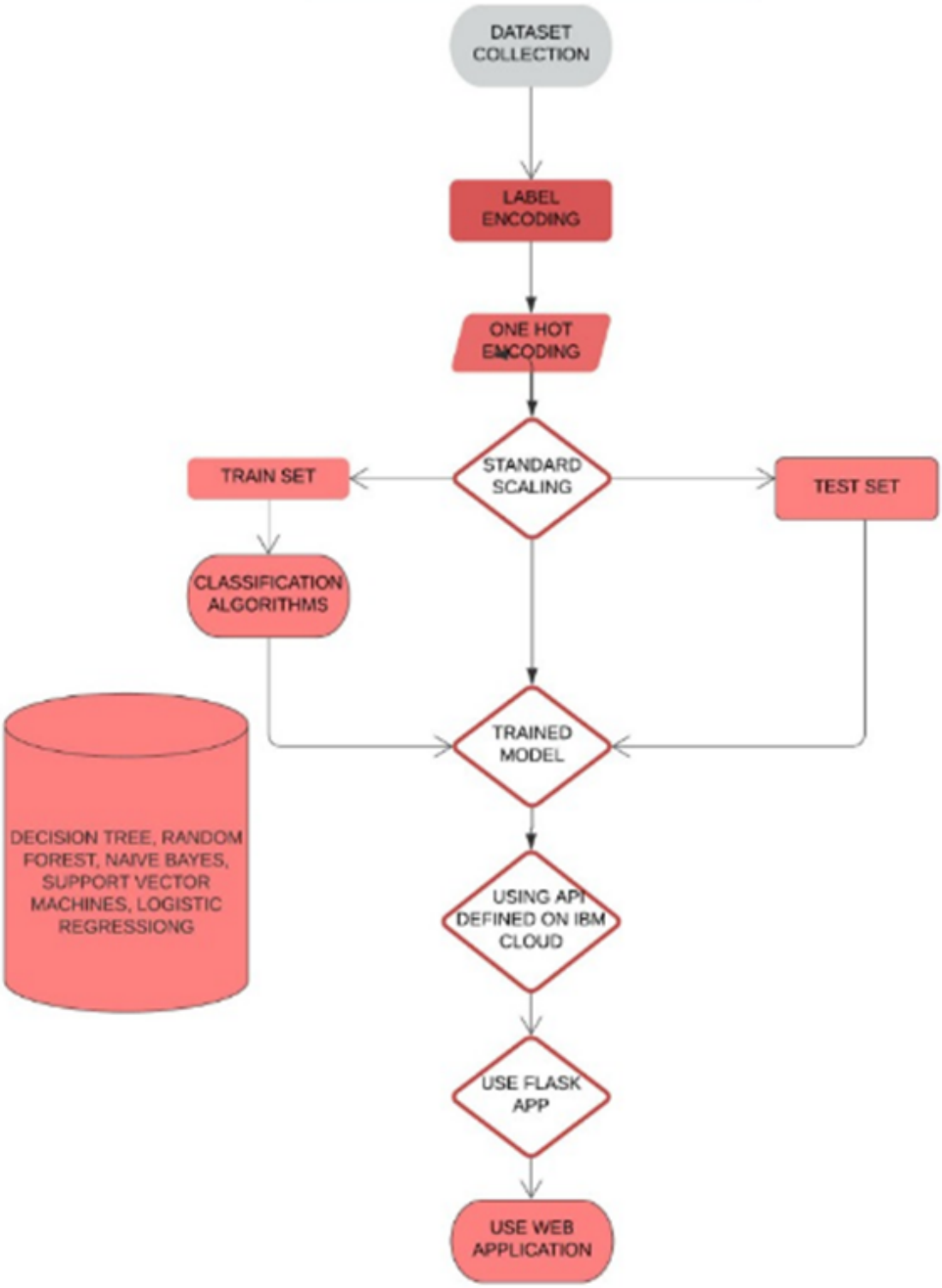
By applying Machine Learning algorithms to the financial risk management dataset, which takes the input values of age, gender, job, housing, credit amount, etc., the risk is calculated and hence concluded whether the risk is good or bad for that particular details of a person. This is done by using various algorithms such as Decision Tree Classifier, Naïve Bayes, Logistic Regression, Support Vector Machine. Out of these, SVM was proved to be the best to train the model, having an accuracy of 69.5%.

3. THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM

The following is the block diagram which summarizes the entire procedure involved in the Model Building of the Financial Risk Management. It basically shows the entire step-wise procedure from the beginning that is collection of the dataset to the building of the web page and running the model.

Block diagram for Financial Risk Management



3.2 HARDWARE / SOFTWARE DESIGNING

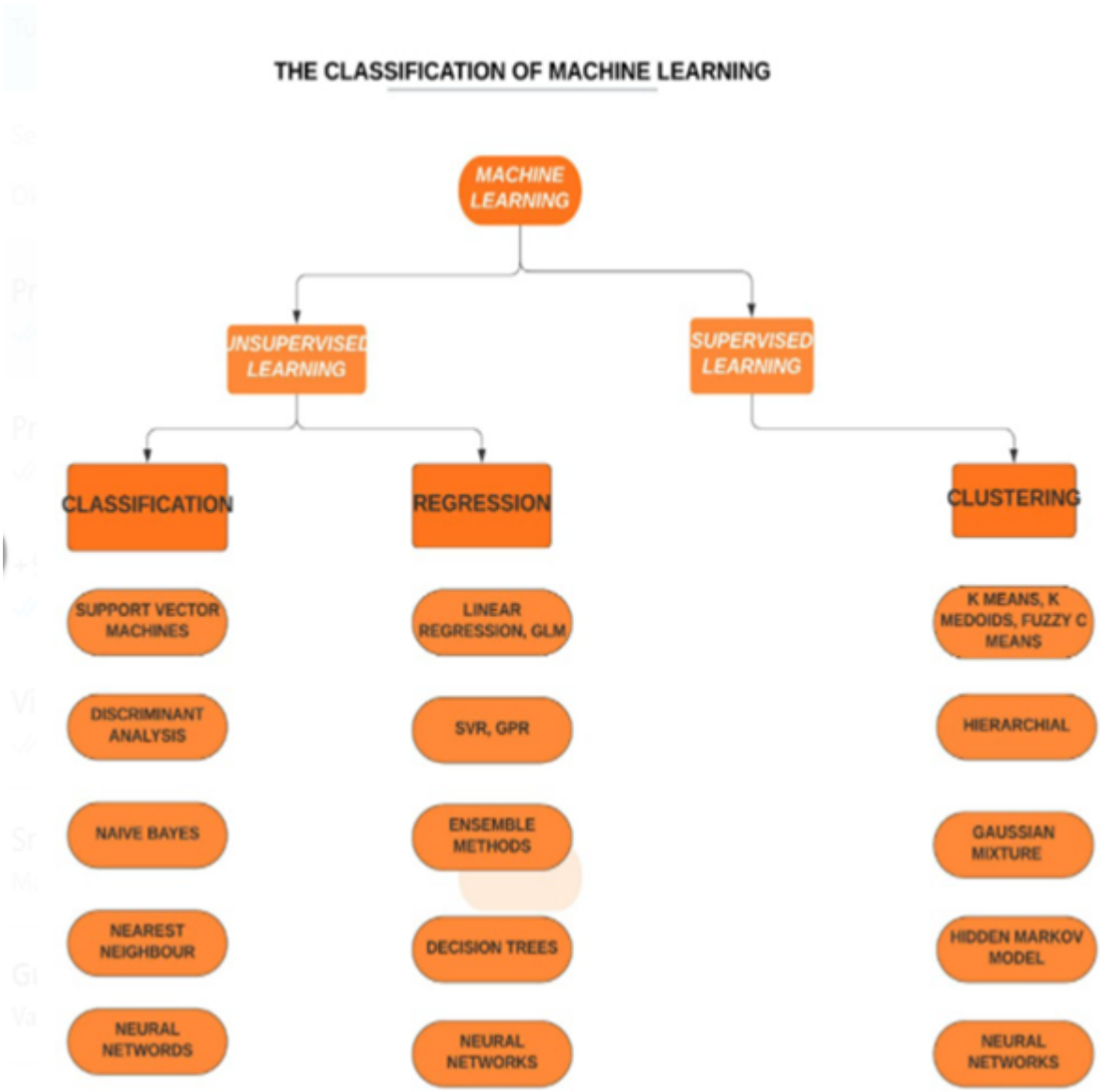
● Hardware

1. Laptop with 4GB RAM
2. 64-bit operating system
3. i7 4. Hard disk

● Software

1. Python 3.6
2. Anaconda environment
3. Jupyter Notebook (You can also use google colab, if your laptop has low computational power)
4. Spyder IDE

The machine learning algorithm used in this model of Financial Risk Management involves the use of Supervised Learning. In supervised learning, the model that is trained here is under the Classification and Regression model. This model uses Classification because the output has to get segregated into different classes and the inputs which are age, gender, etc. when entered give the desired output. Classification in machine learning is in which the program reads and learns the dataset given to convert it into new classifications or classes. Given is an example which uses a Classification model. Considering an example of the data which is about Heart Disease detection. It is a pure binary classification model which will have only two classes i.e. either the person has a heart disease or not. Similarly, our Financial Risk Management model works the same where it is trained with the dataset to understand the required inputs and then the classifier is trained and the one with the best accuracy is considered to give the precise output of a person having his or her financial risk or not.



4. EXPERIMENTAL INVESTIGATIONS

● Data Collection The data for the given topic is collected from the dataset. The dataset contains various categories depending upon the topic. Out of which, some are to be chosen as inputs and the final one as an output i.e. the risk, whether it is good or bad.

● Data Pre-processing

1. Importing the libraries The first and the foremost step for starting the code is by importing the libraries. The two libraries used here in the first step are pandas to read the dataset and convert it into a DataFrame and the second is NumPy to convert the dataset to an array

```
import pandas as pd
import numpy as np
```

2. Importing the dataset The given dataset is imported and then read by using pandas read_csv() (which reads the dataset). The path of the downloaded dataset is given along with the following syntax and then the dataset can be read.

```
dataset = pd.read_csv("datasets_9109_12699_german_credit_data.csv")
```

```
dataset
```

	Unnamed: 0	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	
0	0	67	male	2	own	NaN	little	1169	6	
1	1	22	female	2	own	little	moderate	5951	48	
2	2	49	male	1	own	little	NaN	2096	12	e
3	3	45	male	2	free	little	little	7882	42	furniture/e
4	4	53	male	2	free	little	little	4870	24	
...	...	...	...	...	...	...	...	...	...	
995	995	31	female	1	own	little	NaN	1736	12	furniture/e
996	996	40	male	3	own	little	little	3857	30	
997	997	38	male	2	own	little	NaN	804	12	
998	998	23	male	2	free	little	little	1845	45	
999	999	27	male	2	own	moderate	moderate	4576	45	

1000 rows × 11 columns

3.Taking care of the missing values The dataset is checked if it has any missing values in any of it's classes which are indicated by NaN (Not a Number).

```
In [8]: dataset.isnull().any()

Out[8]: Unnamed: 0          False
        Age                 False
        Sex                 False
        Job                 False
        Housing             False
        Saving accounts     False
        Checking account    False
        Credit amount       False
        Duration            False
        Risk                False
        dtype: bool

In [9]: dataset.drop(["Unnamed: 0"],axis=1,inplace=True)

In [10]: dataset["Duration"].unique()

Out[10]: array([ 6, 48, 12, 42, 24, 36, 30, 15,  9, 10,  7, 60, 18, 45, 11, 27,  8,
                54, 20, 14, 33, 21, 16,  4, 47, 13, 22, 39, 28,  5, 26, 72, 40],
               dtype=int64)

In [11]: dataset["Sex"].unique()

Out[11]: array(['male', 'female'], dtype=object)

In [12]: dataset["Housing"].unique()

Out[12]: array(['own', 'free', 'rent'], dtype=object)

In [13]: dataset["Saving accounts"].unique()
```

The places where there are True indicate that there are NaN values present in those particular columns. Then these missing values are filled using mean mode or median accordingly.

Then all the columns are checked for the missing values. All Falses show that there are no missing values left in the dataset.

4. Label Encoding Label Encoding refers to the conversion of textual data into numerical data. All the columns having textual data like gender, housing, risk, etc are converted into numerical data.

```
In [16]: from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
dataset["Sex"]=le.fit_transform(dataset["Sex"])
dataset["Housing"]=le.fit_transform(dataset["Housing"])
dataset["Saving accounts"]=le.fit_transform(dataset["Saving accounts"])
dataset["Checking account"]=le.fit_transform(dataset["Checking account"])
dataset["Risk"]=le.fit_transform(dataset["Risk"])
```

```
In [17]: dataset
```

	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Risk
0	67	1	2	1	0	0	1169	6	1
1	22	0	2	1	0	1	5951	48	0
2	49	1	1	1	0	0	2096	12	1
3	45	1	2	0	0	0	7882	42	1
4	53	1	2	0	0	0	4870	24	0
...	...	...	...	...	...	...	...	...	...
995	31	0	1	1	0	0	1736	12	1
996	40	1	3	1	0	0	3857	30	1
997	38	1	2	1	0	0	804	12	1
998	23	1	2	0	0	0	1845	45	0
999	27	1	2	1	1	1	4576	45	1

1000 rows × 9 columns

5. Taking input and output from the dataset From the dataset required columns are chosen as input which affect the output that is the financial risk. Then the selected inputs and output is converted in the form of arrays using '.values'.

```
x=dataset.iloc[:,0:8].values
y=dataset.iloc[:,8:9].values
```

```
x
```

```
array([[  67,    1,    2, ...,    0, 1169,    6],
       [  22,    0,    2, ...,    1, 5951,   48],
       [  49,    1,    1, ...,    0, 2096,   12],
       ...,
       [  38,    1,    2, ...,    0,  804,   12],
       [  23,    1,    2, ...,    0, 1845,   45],
       [  27,    1,    2, ...,    1, 4576,   45]], dtype=int64)
```

6. Testing and Training the model Splitting the dataset into train and test. 20% of the dataset is kept for testing and 80% is kept for training

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.2,random_state=0)
```

7. Standard Scaling Standard scaling is thus done to the executed data.

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
```

● Model Building

1. Building and training the model The model has to be built using the different types of Classifiers like Decision Tree Classifier, Logistic Regression, Random Forest Classifier, Support Vector Machines, etc. Among these the one with the highest accuracy is taken as the most precise classifier and hence the inputs are given under these. One of the classifier used here is given below

```
In [34]: from sklearn.tree import DecisionTreeClassifier
         dtc=DecisionTreeClassifier(random_state=5,criterion="entropy")
         dtc.fit(x_train,y_train)

Out[34]: DecisionTreeClassifier(criterion='entropy', random_state=5)

In [35]: dtcpred=dtc.predict(x_test)
         dtcpred
```

The accuracy of this decision tree classifier is found

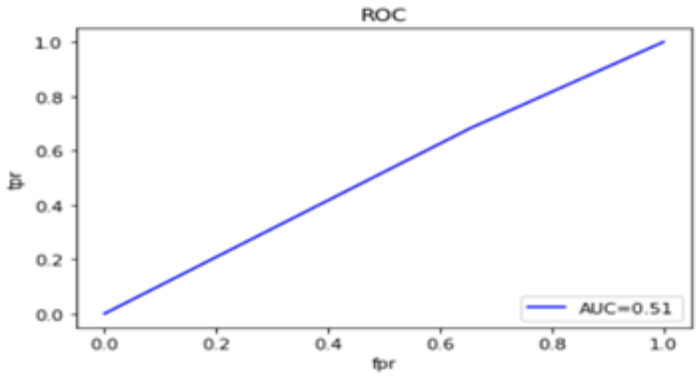
```
In [37]: from sklearn.metrics import accuracy_score
         accuracy=accuracy_score(y_test,dtcpred)
         accuracy

Out[37]: 0.585
```

The ROC curve is found and the AUC is also checked using other libraries.

```
In [41]: import matplotlib.pyplot as plt
         plt.plot(fpr,tpr,"blue",label="AUC=%.2f"%roc_auc)
         plt.legend(loc="lower right")
         plt.title("ROC")
         plt.xlabel("fpr")
         plt.ylabel("tpr")

Out[41]: Text(0, 0.5, 'tpr')
```



Since the accuracy of this classifier isn't satisfactory other classifiers are used and the one with best accuracy or AUC is chosen.

Using Support Vector Machines (SVM)

```
[79]: from sklearn.svm import SVC
      svm=SVC(kernel="rbf",random_state=7)
      svm.fit(x_train,y_train)

      C:\Users\Prathul\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pa
      ssed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
        return f(*args, **kwargs)

Out[79]: SVC(random_state=7)

[80]: import pickle
      pickle.dump(svm,open("svmrisk.pkl",'wb'))

[81]: svmpred=svm.predict(x_test)

[82]: svmpred
```

Then the accuracy of SVM is obtained and the dataset is predicted using random values from the dataset.

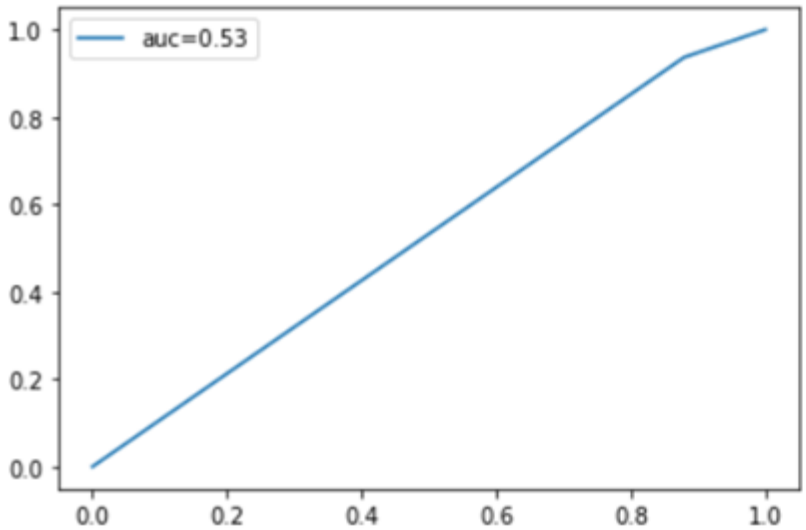
```
In [83]: svmacc=accuracy_score(y_test,svmpred)

In [84]: svmacc
Out[84]: 0.7
```

The ROC curve is also obtained using the respective libraries and AUC is also checked.

```
import matplotlib.pyplot as plt
plt.plot(svmfpr,svmtpr,label="auc=%0.2f"%svmroc_auc)
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1dfb79a2be0>
```

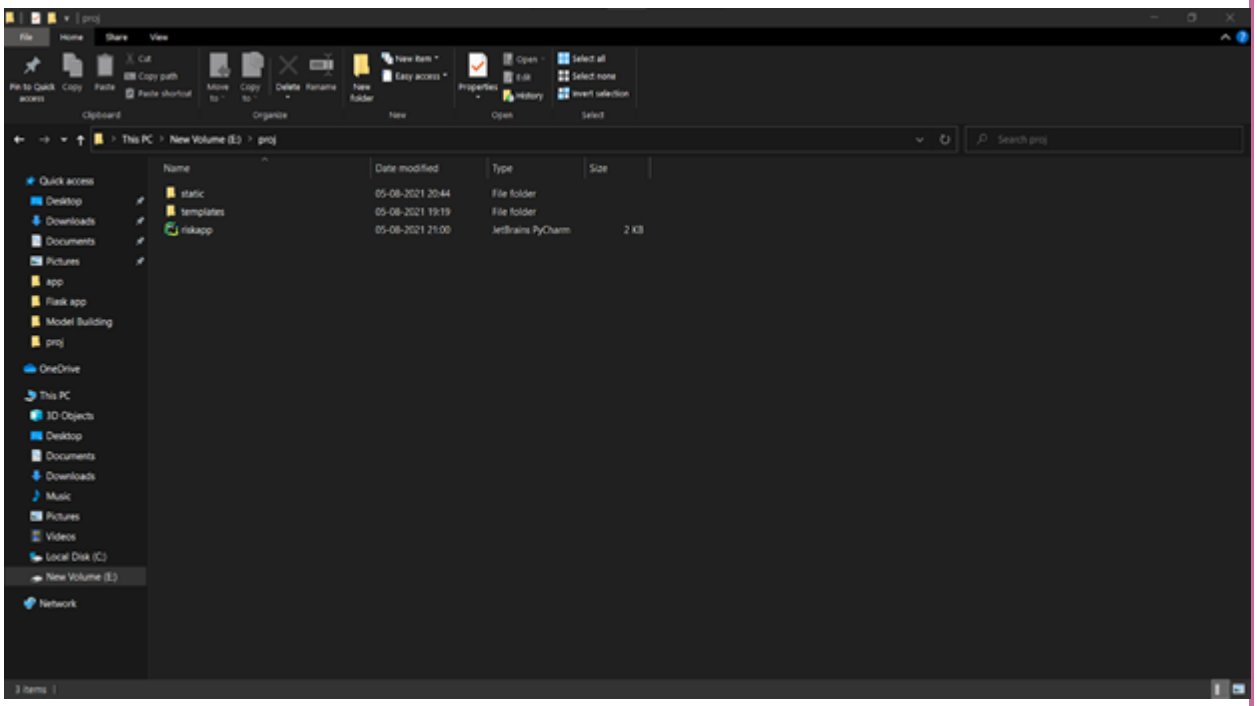


After saving the model, we pickle the file by importing the respective library into the precise Classifier with a high accuracy.

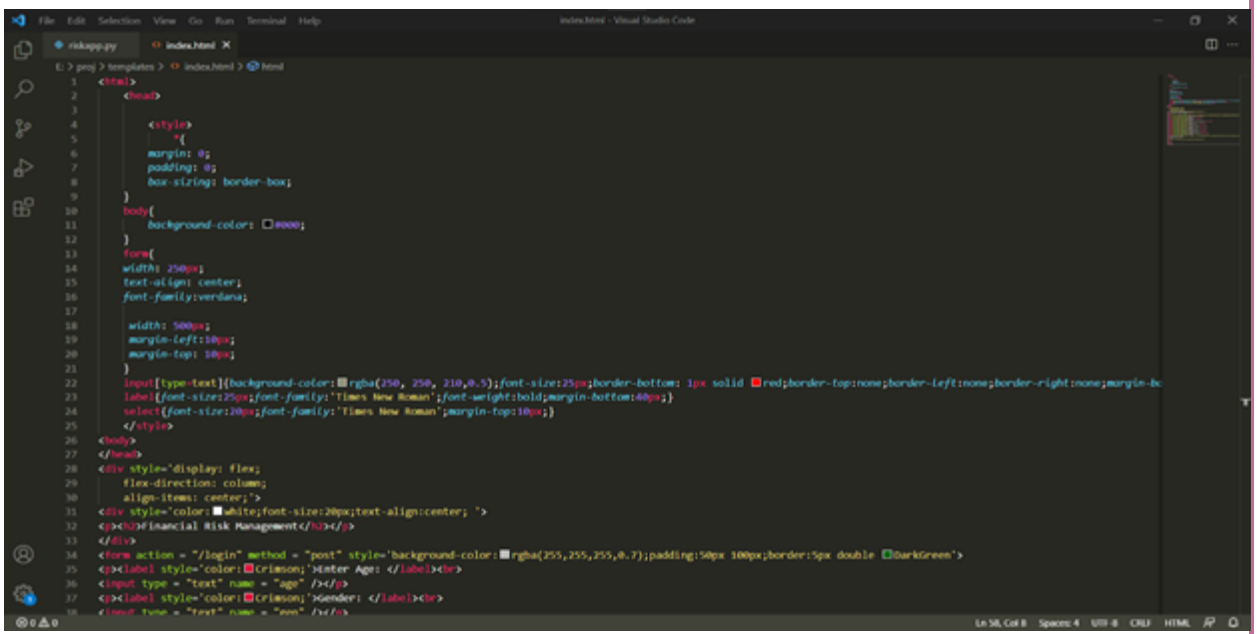
```
[33]: 1 import pickle
      2 pickle.dump(svm,open("svmrisk.pkl",'wb'))
```

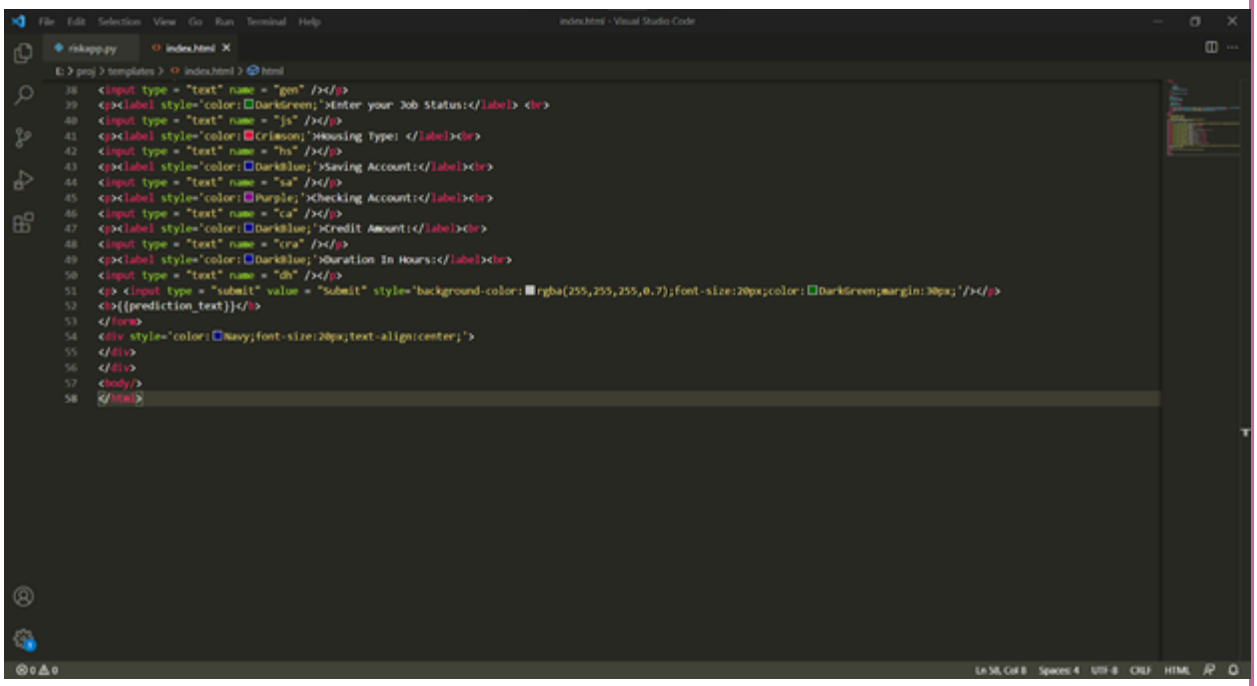
INTEGRATING A WEBSITE WITH THE MACHINE LEARNING MODEL:(by using flask app)

Create a folder and then create templates folder and save the HTML file in it as index.html
In the main folder we should have our app.py and pkl file

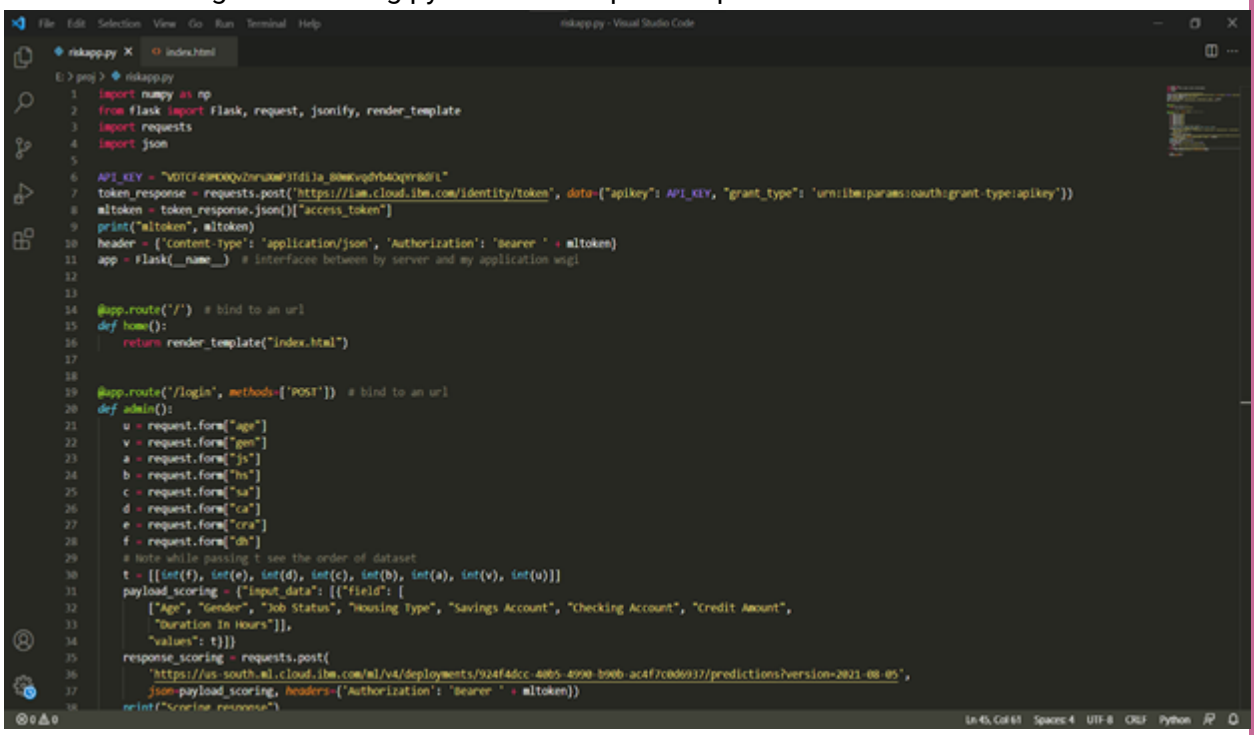


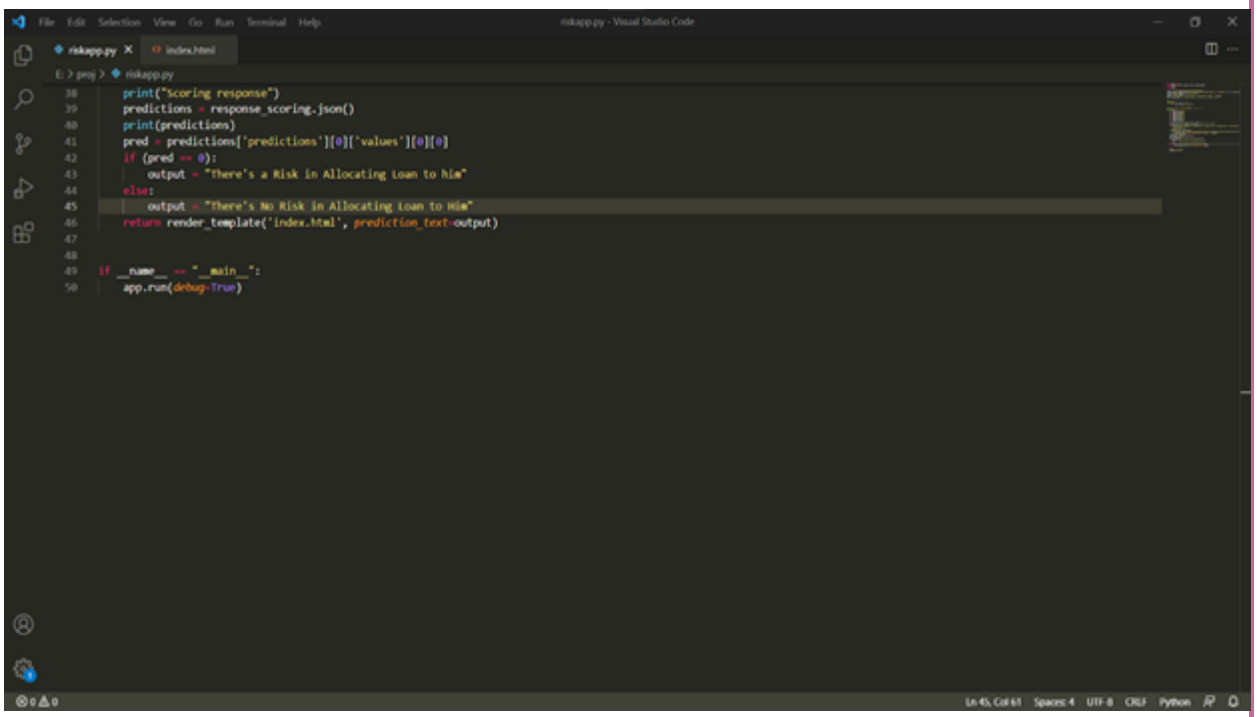
In the spyder app we are applying CSS to our HTML file, along with the HTML code for our UI as shown:



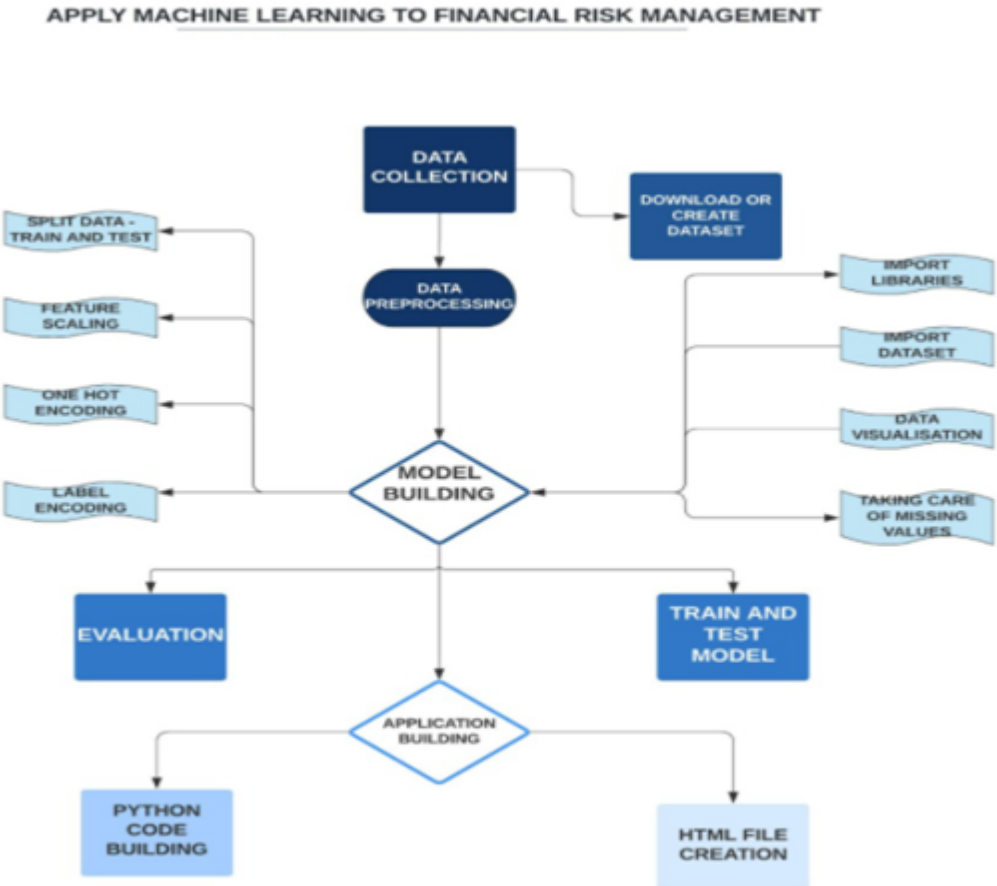


And we are using the following python code to print response:




riskapp.py - Visual Studio Code
print("Scoring response")
predictions = response_scoring.json()
print(predictions)
pred = predictions['predictions'][0]['values'][0][0]
if (pred == 0):
output = "There's a Risk in Allocating Loan to him"
else:
output = "There's No Risk in Allocating Loan to Him"
return render_template('index.html', prediction_text=output)
if __name__ == "__main__":
app.run(debug=True)

5. FLOWCHART


APPLY MACHINE LEARNING TO FINANCIAL RISK MANAGEMENT
DATA COLLECTION
DOWNLOAD OR CREATE DATASET
DATA PREPROCESSING
SPLIT DATA - TRAIN AND TEST
FEATURE SCALING
ONE HOT ENCODING
LABEL ENCODING
IMPORT LIBRARIES
IMPORT DATASET
DATA VISUALISATION
TAKING CARE OF MISSING VALUES
MODEL BUILDING
EVALUATION
TRAIN AND TEST MODEL
APPLICATION BUILDING
PYTHON CODE BUILDING
HTML FILE CREATION

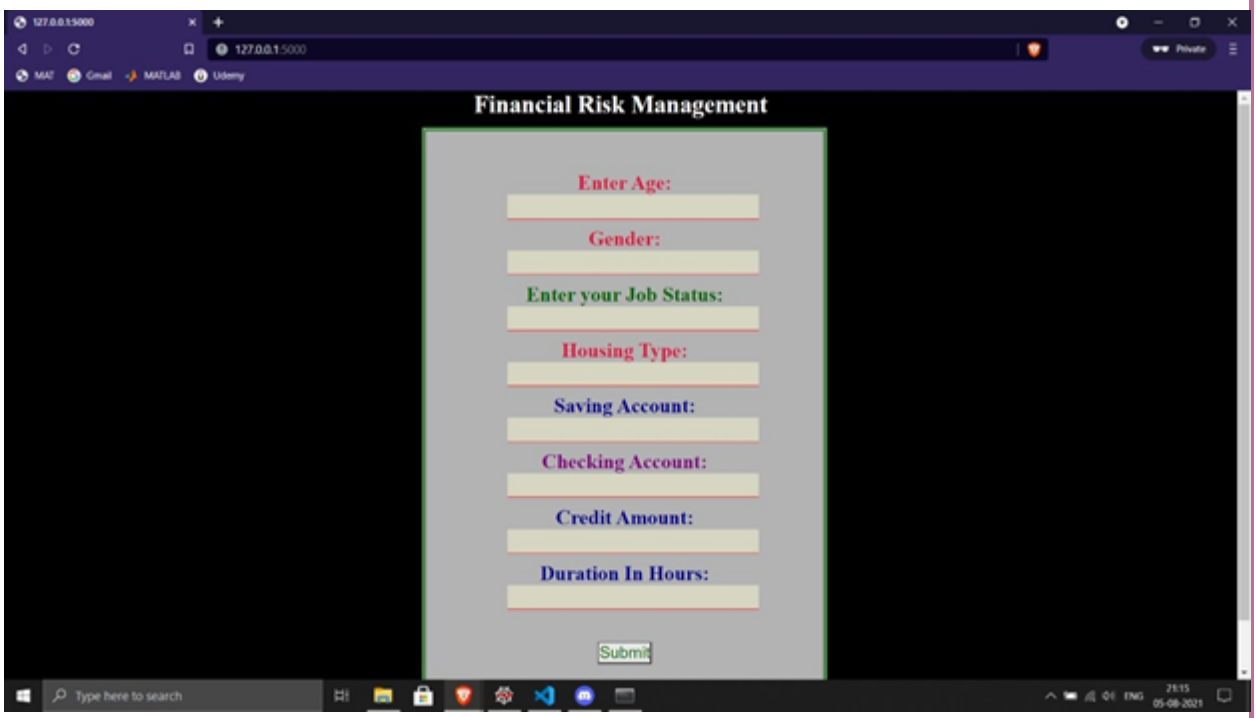
6. RESULT

This project collects the data of around 1000 people and gives out to losses or risk and to protect the value of its assets.

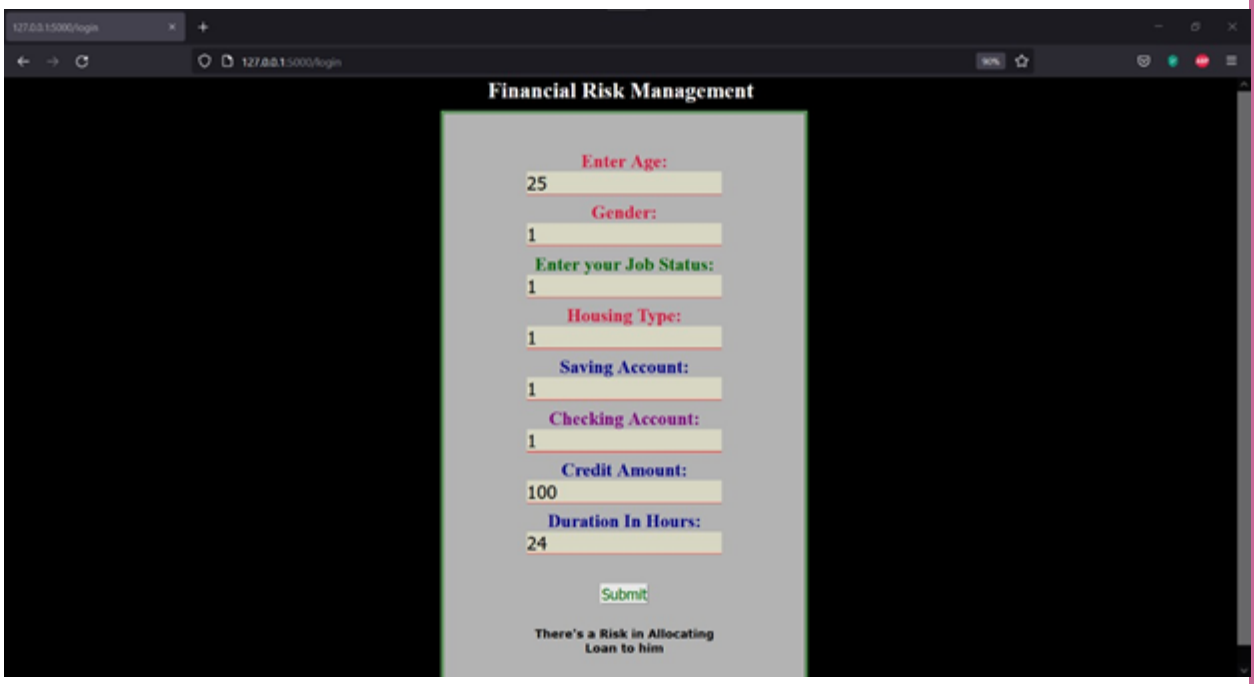
Machine learning is applied to calculate the risk the company might have to face due to any particular person.

For this, the project uses various technical entities such as the anaconda prompt, jupyter notebook, spyder.

Here we show you one output that we derived on the basis of the details entered by the customer:



The output of this information was calculated and given as:



The output is given as 1, which means the risk is good. Once the basic empirical results have been obtained, decisions have to be made about the appropriate risk-management techniques to use for managing the risks.

7. ADVANTAGES

1. This model helps people to know their financial status and also helps them know if they are at a risk or not
2. It take in age, gender , housing etc to and helps customers to know if they are financially stable or not
3. It secures the financial position of the company by helping them know which customers may pose a risk to the company.

DISADVANTAGES:

1. This model only takes age, gender, job and housing etc as an input. This may serve as a disadvantage as detailed information is not being inserted
2. The age has to be changed into 1 for male and 0 for female. Customers might not know this and hence it may cause difficulties.

8. APPLICATIONS

These days everything is being managed on a digital scale. From education to shopping to jobs, the internet has opened vast opportunities for everybody. A lot of business transactions are also made online. This creates a need to manage the finances technically on a very large scale in a short period of time. Hence the management of finances using machine learning is the most convenient way to do so. These methods are used in banks to calculate what risk a customer pay poses to the bank's financial management. It used in large, small and multi business companies to manage their finances easily and safely. They are used in almost every central and private sector. Financial management for any start up is necessary. It is necessary. Companies must always keep a track of where all the money is being spent and that is where machine learning comes into picture. Financial management through machine learning opens many windows of opportunities to the world and helps small start ups flourish.

9. CONCLUSIONS

To conclude, our model correctly and precisely predicts the risk the company may have to face in investing in any particular member. The model decides if the risk will be good or bad for the company.

When we type in the inputs of age , gender(0=female,1=male),job, housing, saving account, checking account , credit amount and duration is hours it will calculate and give us the output as good or bad.

10. FUTURE SCOPE

Artificial Intelligence is a game-changer for risk management in finance as it provides banks and credit unions with tools and AI solutions to identify potential risks and fraud. The financial crisis of the previous decade gave financial services firms a lot of problems with credit-challenged consumers. Before the digital revolution in financial services industry customer intelligence was based on some relatively simple heuristics, the customer value data was gained through focus groups and surveys of consumer behavior the results of which didn't always correspond to reality.

Today new technologies give businesses access to really tremendous amounts of data about consumers' behavior and needs.

Risk management in banks should use cognitive technologies to gain competitive advantage and use risk to power their organizations' performance.

Artificial intelligence and risk management perfectly align when there is a need for handling and evaluating unstructured data. It is estimated that risk managers of financial institutions will focus on analytics and stopping losses in a proactive manner based on AI findings, rather than spending time in managing the risks inherent in the operational processes.

AI solutions are able to fuel financial institutions with trusted and timely data for building competence around their customer intelligence and successful implementation of their strategies.

11. BIBLIOGRAPHY

[1] https://en.wikipedia.org/wiki/Financial_risk_management

[2] https://www.edureka.co/blog/classification-in-machine-learning/

[3] https://archer-soft.com/blog/how-ai-changing-risk-management

APPENDIX

A. CODE

Flask code:

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import requests
import json

API_KEY = "VDTCF49MO0QvZnruXmP3TdiJa_80mKvqdYb4OqYr8dFL"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
print("mltoken", mltoken)
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
app = Flask(__name__)  # interfacee between by server and my application wsgi


@app.route('/')  # bind to an url
def home():
    return render_template("index.html")


@app.route('/login', methods=['POST'])  # bind to an url
def admin():
    u = request.form["age"]
    v = request.form["gen"]
    a = request.form["js"]
    b = request.form["hs"]
    c = request.form["sa"]
    d = request.form["ca"]
    e = request.form["cra"]
    f = request.form["dh"]
    # Note while passing t see the order of dataset
    t = [[int(f), int(e), int(d), int(c), int(b), int(a), int(v), int(u)]]
    payload_scoring = {"input_data": [{"field": [
```

```
        ["Age", "Gender", "Job Status", "Housing Type", "Savings Account", "Checking Account",
"Credit Amount",
         "Duration In Hours"]],
        "values": t}]}
    response_scoring = requests.post(

'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/924f4dcc-40b5-4990-b90b-ac4f7c0
d6937/predictions?version=2021-08-05',
        json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    predictions = response_scoring.json()
    print(predictions)
    pred = predictions['predictions'][0]['values'][0][0]
    if (pred == 0):
        output = "There's a Risk in Allocating Loan to him"
    else:
        output = "There's No Risk in Allocating Loan to Him"
    return render_template('index.html', prediction_text=output)


if __name__ == "__main__":
    app.run(debug=True)
```

HTML Code:

```
<html>
   <head>

      <style>
         *{
      margin: 0;
      padding: 0;
      box-sizing: border-box;
   }
   body{
      background-color: #000;
   }
   form{
   width: 250px;
   text-align: center;
   font-family:verdana;

    width: 500px;
    margin-left:10px;
    margin-top: 10px;
   }
   input[type=text]{background-color:rgba(250, 250, 210,0.5);font-size:25px;border-bottom:
1px solid red;border-top:none;border-left:none;border-right:none;margin-bottom:10px}
   label{font-size:25px;font-family:'Times New Roman';font-weight:bold;margin-bottom:40px;}
   select{font-size:20px;font-family:'Times New Roman';margin-top:10px;}
   </style>
<body>
</head>
<div style='display: flex;
   flex-direction: column;
   align-items: center;'>
<div style='color:white;font-size:20px;text-align:center; '>
<p><h2>Financial Risk Management</h2></p>
</div>
```

```
<form action = "/login" method = "post"
style='background-color:rgba(255,255,255,0.7);padding:50px 100px;border:5px double
DarkGreen'>
<p><label style='color:Crimson;'>Enter Age: </label><br>
<input type = "text" name = "age" /></p>
<p><label style='color:Crimson;'>Gender: </label><br>
<input type = "text" name = "gen" /></p>
<p><label style='color:DarkGreen;'>Enter your Job Status:</label> <br>
<input type = "text" name = "js" /></p>
<p><label style='color:Crimson;'>Housing Type: </label><br>
<input type = "text" name = "hs" /></p>
<p><label style='color:DarkBlue;'>Saving Account:</label><br>
<input type = "text" name = "sa" /></p>
<p><label style='color:Purple;'>Checking Account:</label><br>
<input type = "text" name = "ca" /></p>
<p><label style='color:DarkBlue;'>Credit Amount:</label><br>
<input type = "text" name = "cra" /></p>
<p><label style='color:DarkBlue;'>Duration In Hours:</label><br>
<input type = "text" name = "dh" /></p>
<p> <input type = "submit" value = "Submit"
style='background-color:rgba(255,255,255,0.7);font-size:20px;color:DarkGreen;margin:30px;'/
></p>
<b>{{prediction_text}}</b>
</form>
<div style='color:Navy;font-size:20px;text-align:center;'>
</div>
</div>
<body/>
</html>
```