**Applying Machine Learning To Financial Risk Management**

**Developed by:**Aitha Sai Priya-(18UK1AO501),Gudikandula BharathChandra-(18UK1A0574),Sravani Karne- (18UK1A0553),Dongala AnjiReddy-(18UK1A0514)

# 1.INTRODUCTION

**1.1 Overview**

Timely loan repayment along with determining the level of customer reliability is one of the major elements of credit risk assessment. Based on the customer's characteristics credit history analysis and scoring is done. Credit scoring is one of the methods widely used for estimation of the risks associated in granting a loan, or rather the probability of its non-repayment. On the basis of the calculation according to data provided in the loan applications the customer score is obtained which determines the risk levels associated with it. Regardless of how it is calculated and what characteristics taken into account, it eliminates the human factors, adds objectivity to the process which in turn speeds up the process makes it smooth and reduce the risk. Banks usually sanction the loan on the basis of qualitative and quantitative analysis.  Based on statistical methods credit scoring helps to predict the probability of a certain events occurring in the future. Credit Scoring models can be classified according to different criteria. Thus, we can talk about a scoring of individuals or companies or credit card, cash or mortgage scoring. The goal of the scoring models for most of the parts is to determine the risk of debt default.

**1.2 Purpose**

# Many lenders and financial institutions use statistical credit scoring analysis to determine the creditworthiness of a person or small scale business. Creditworthiness is how a lender will determines the failure of the debt on debt obligation and whether the person is suitable for new credit or not. Lenders uses credit scoring for the analysis of the risks associated with the sanction of the loan or credit, or to decide whether to extend or deny the credit. Credit scoring determines the person’s ability to borrow money for mortgages and for different .

**2.LITERATURE SURVEY**

**2.1 Existing Problem:**

## [Financial Risks and Management](https://www.sciencedirect.com/science/article/pii/B9780081022955100654) :

Ekaterina Svetlova, Karl-Heinz Thielmann

Traditional Banking methods have limitations where they cannot analyse large volumes of data and are entirely dependent on credit scores and limited values. These greatly reduce the capability of institutions to reduce risk.

The number of errors that take place on a daily basis in traditional systems is a cause for concern. The cost and manpower required to run this system are immense and it takes a large amount of time for any changes to be implemented in the system.

These concepts are explored in great detail in this particular literature.

**2.2 Proposed Solutions:**

Palgrave Studies in Digital Business & Enabling Technologies

Chapter 3: Machine Learning and AI for Risk Management- Saqib Aziz and Michael Dowling

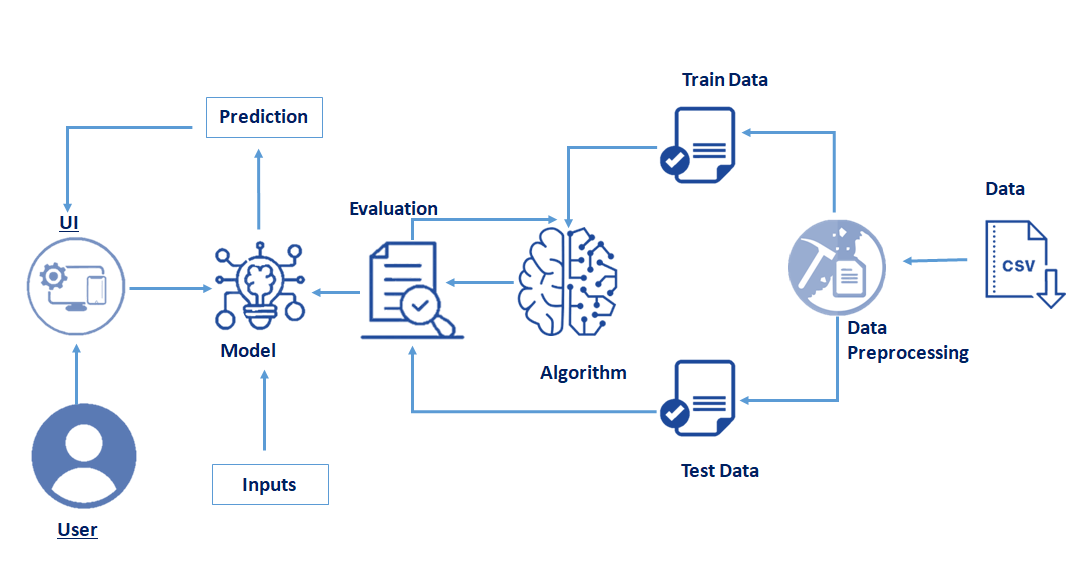
This particular chapter explores the possible uses of AI and Machine Learning in well established banking systems, it explores the possible use of ML to free man power and reduce the various risks prevalent in the banking system.

Machine Learning algorithms give investment managers, consumers and entire banking corporations future insights into the how the market will change much earlier than traditional banking models.

Using machine learning techniques, banks and financial institutions can significantly lower the risk levels by analysing a massive volume of data sources. Unlike the traditional methods which are usually limited to essential information such as credit score, ML can analyze significant volumes of personal information to reduce their risk.

**3.THEORETICAL ANALYSIS**

**3.1 Project Architecture**

****

**Figure1: Project Architecture**

**3.2 Hardware/Software designing**

* Initially, we have used Jupyter Notebook as a tool for data retrieving and preprocessing of the dataset and later on moved towards the model building by applying various classification algorithms.
* After performing all the algorithms that comes under classification, we have chosen the one that has given the highest accuracy and implemented it on a web application using the flask framework for server-side scripting.
* With the help of flask a user can interact with the UI (User Interface) to enter the input values that can be analyzed by the model based on which prediction is showcased on the UI.

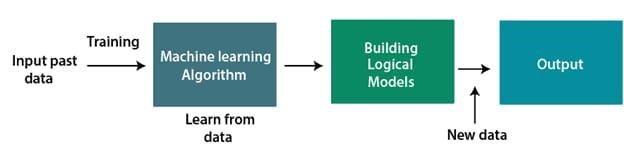
Software Requirements:

* Anaconda Environment
* Flask
* Python 3.9
* And other python libraries like NumPy, pandas.

Hardware Requirements:

* Operating system
* Processing
* RAM
* Operating system specifications
* Disk space

**3.3 Block Diagram**



**Figure 2:Block Diagram**

**4. EXPERIMENTAL ANALYSIS**

**4.1 PROJECT FLOW**

* User interacts with the UI (User Interface) to enter the input values
* Entered input values are analyzed by the model which is integrated
* Once model analyses the input the prediction is showcased on the UI
* To accomplish this, we have to complete all the activities and tasks listed below

**Milestone-1:Data Collection.**

* ML depends heavily on data, without data, a machine can't learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.
* You can collect datasets from different open sources like kaggle.com, data.gov; UCI machine learning repository etc. The dataset used for this project was obtained from Kaggle.

**Milestone-2:Data Preprocessing.**

* Import the Libraries.
* Importing the dataset.
* Checking for Null Values.
* Data Visualization.
* Taking care of Missing Data.
* Splitting Data into Train and Test.
* Feature Scaling.

**Milestone-3:Model Building**

The model building process involves setting up ways of collecting data, understanding and paying attention to what is important in the data to answer the questions you are asking, finding a statistical, mathematical or a simulation model to gain understanding and make predictions. Model Building Includes:

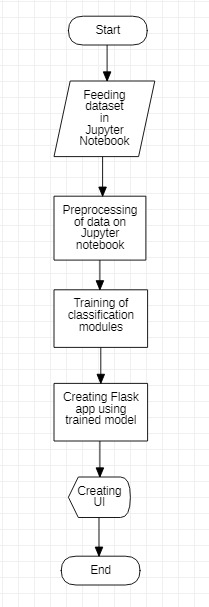
* Import the model building Libraries
* Initializing the model
* Training and testing the model
* Evaluation of Model
* Save the Model

**Milestone-4:Application Building**

* Create an HTML file
* Create an App.py file

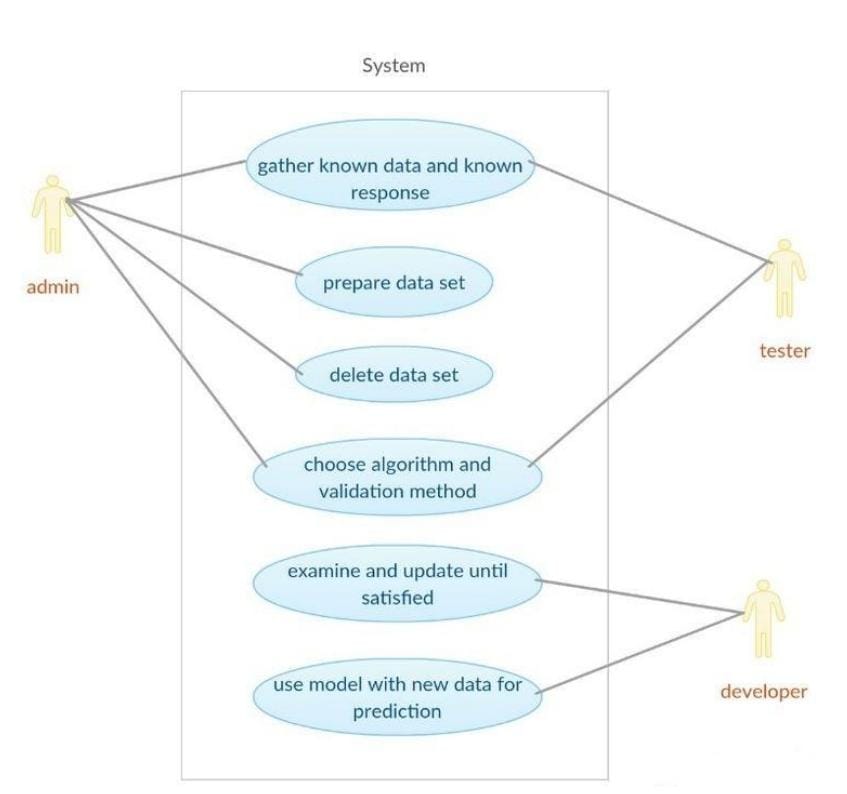
**5.DESIGN**

**5.1 FLOW CHART:**

****

**Figure 3:Flowchart**

**5.2 USE CASE:**



**Figure 4: Use case Diagram**

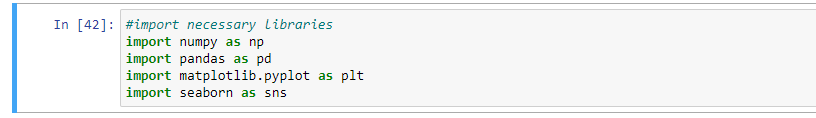
**6.CODE SNIPPETS**

**6.1 MODEL CODE:**

### Necessary Libraries:

It is important to import all the necessary libraries such as pandas, numpy, matplotlib.

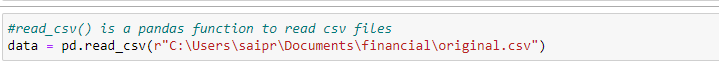
* **Numpy**- It is an open-source numerical Python library. It contains a multi-dimensional array and matrix data structures. It can be used to perform mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.
* **Pandas**- It is a fast, powerful, flexible and easy to use opensource data analysis and manipulation tool, built on top of the Python programming language.
* **Seaborn**- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
* **Matplotlib**- Visualization with python. It is a comprehensive library for creating static, animated, and interactive visualizations in Python



**Figure 5: ipynb code describing importing libraries**

**Importing The Dataset:**

* You might have your data in .csv files, .excel files
* Let’s load a .csv data file into pandas using **read\_csv() function**. We will need to locate the directory of the CSV file at first (it’s more efficient to keep the dataset in the same directory as your program).



**Figure 6:** **ipynb code describing reading the dataset**

* If your dataset is in some other location ,Then

Data=pd.read\_csv(r”File\_location”)

**Note**: r stands for "raw" and will cause backslashes in the string to be interpreted as actual backslashes rather than special characters.

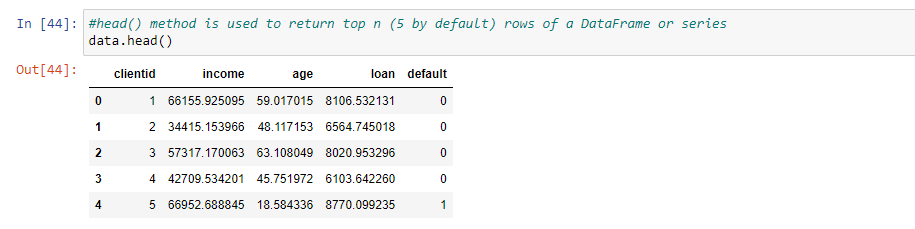
* If the dataset in same directory of your program, you can directly read it, without giving raw as r.
* Our Dataset contains following Columns

1. Age
2. Income
3. Loan

The output column to be predicted is Risk .Based on the input variables we predict the financial risk.

### Analyse The Data:

* head() method is used to return top n (5 by default) rows of a DataFrame or series.

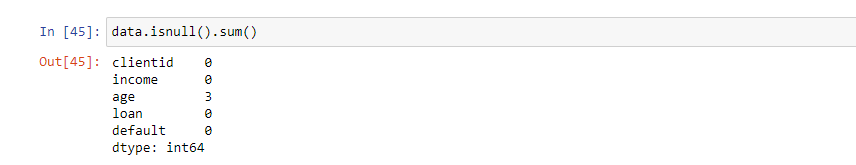
****

**Checking For Null Values Or Taking Care Of Missing Data:**

Sometimes you may find some data are missing in the dataset. We need to be equipped to handle the problem when we come across them. One of the most common ideas to handle the problem is

to take a mean of all the values of the same column and have it to replace the missing data.

We will be using isnull().sum() method to see which column has missing values.isnull().sum() will give you the total count of null values present in each column

****

**Figure 7: ipynb code checking for number of null values in dataset**

### Handling Missing Values:

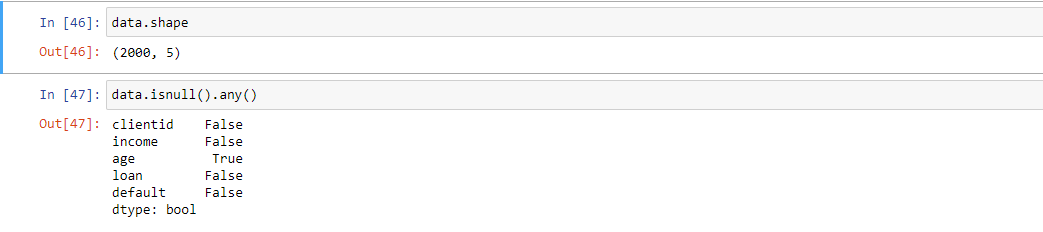
After loading it is important to check the complete information of data as it can indication many of the hidden infomation such as null values in a column or a row

Check whether any null values are there or not. if it is present then following can be done,

a.Imputing data using Imputation method in sklearn

b.Filling NaN values with mean, median and mode using **fillna()**method.

.



**Figure 8: ipynb code checking for any null values in dataset**

**6.png**

### Data Visualization:

* Data visualization is where a given data set is presented in a graphical format. It helps the detection of patterns, trends and correlations that might go undetected in text-based data.
* Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. In fact, even the most sophisticated machine learning models will perform poorly on data that wasn’t visualized and understood properly.
* To visualize the dataset we need libraries called Matplotlib and Seaborn.
* The Matplotlib library is a Python 2D plotting library which allows you to generate plots, scatter plots, histograms, bar charts etc.

Let’s visualize our data using Matplotlib and searborn library.

Before diving into the code, let's look at some of the basic properties we will be using when plotting.

**xlabel:** Set the label for the x-axis.

**ylabel:** Set the label for the y-axis.

title: Set a title for the axes.

              Legend: Place a legend on the axes.

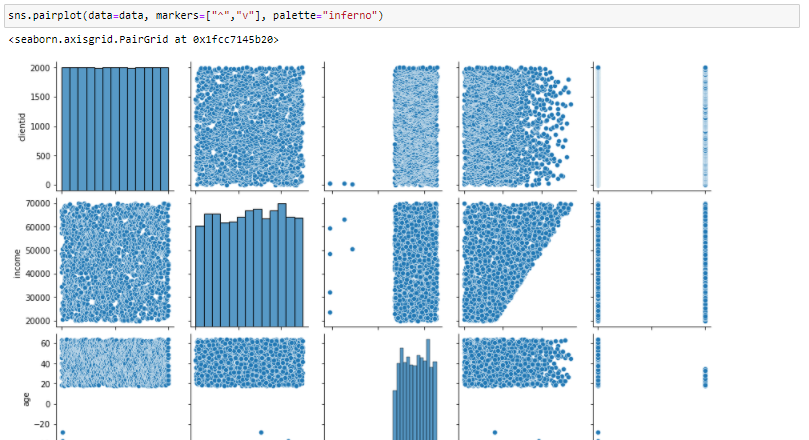
1. data.corr() gives the correlation between the columns



**Figure 9: ipynb code describing correlation graph**

2.Pair Plot: Plot pairwise relationships in a dataset.

* By default, this function will create a grid of Axes such that each numeric variable in data will by shared across the y-axes across a single row and the x-axes across a single column. The diagonal plots are treated differently: a univariate distribution plot is drawn to show the marginal distribution of the data in each colum.
* We implement this using the below code Pair plot usually gives pair wise relationships of the columns in the dataset

****

**Figure 10: ipynb code describing pair plot.**

**Drop The Columns:**

dropping is one of the primary operations when it comes to data analysis. Very often we see that a particular attribute in the data frame is not at all useful for us while working on a specific analysis(example: name, ID etc), rather having it may lead to problems and unnecessary change in the prediction.

To drop it the syntax looks like. Here drop is a function; we need to pass the column names inside the function. And the axis=1 represents the Column wise operation

Now dropping a client id column from data.

**9.png**

**Figure 11: ipynb code for deleting or droping the unwanted columns**

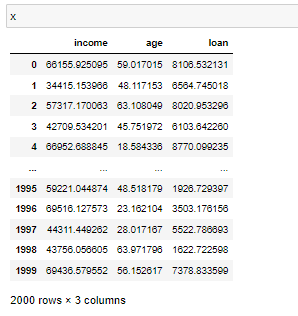
### Splitting The Dataset Into Dependent And Independent Variable:

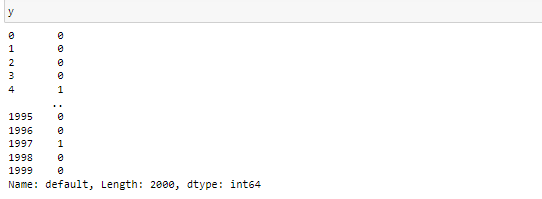
* In machine learning, the concept of dependent variable (y) and independent variables(x) is important to understand. Here, Dependent variable is nothing but output in the dataset and the independent variable is all inputs in the dataset.
* With this in mind, we need to split our dataset into the matrix of independent variables and the vector or dependent variable. Mathematically, Vector is defined as a matrix that has just one column.

### 10.png

After splitting we see the data as below

X and Y





**Figure 12: ipynb code for Splitting The Dataset Into Dependent And Independent Variable.**

**Import the Imblearn library:**

Imbalanced-learn (imported as imblearn ) is an open source, MIT-licensed library relying on scikit-learn (imported as sklearn ) and provides tools when dealing with classification with imbalanced classes.

Imbalanced-Learn is a Python module that helps in balancing the datasets which are highly skewed or biased towards some classes. Thus, it helps in resampling the classes which are otherwise oversampled or undesampled. If there is a greater imbalance ratio, the output is biased to the class which has a higher number of examples.



**Figure 13: ipynb code for balance the Imbalanced dataset**

### Splitting The Data Into Train And Test:

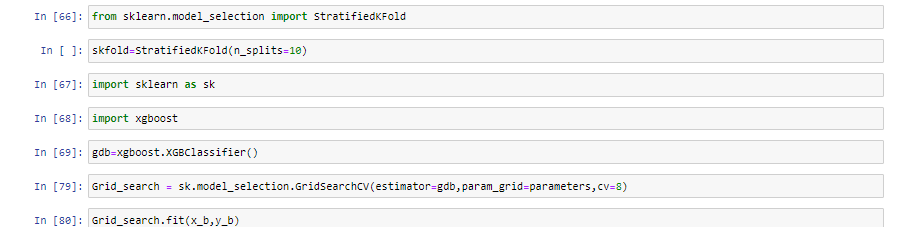
* The train-test split is a technique for evaluating the performance of a machine learning algorithm.
* **Train Dataset**: Used to fit the machine learning model.
* **Test Dataset**: Used to evaluate the fit machine learning model.
* In general you can allocate 80% of the dataset to training set and the remaining 20% to test set.We will create 4 sets— X\_train (training part of the matrix of features), X\_test (test part of the matrix of features), Y\_train (training part of the dependent variables associated with the X train sets, and therefore also the same indices), Y\_test (test part of the dependent variables associated with the X test sets, and therefore also the same indices.
* There are a few other parameters that we need to understand before we use the class:
* **test\_size** — this parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset
* **train\_size**— you have to specify this parameter only if you’re not specifying the test\_size. This is the same as test\_size, but instead you tell the class what percent of the dataset you want to split as the training set.
* **random\_state** — here you pass an integer, which will act as the seed for the random number generator during the split. Or, you can also pass an instance of the Random\_state class, which will become the number generator. If you don’t pass anything, the Random\_state instance used by np.random will be used instead.
* Now split our dataset into train set and test using train\_test\_split class from scikit learn library.



**Figure 14: ipynb code for splitting data into train and test**

**Stratified K-Folds cross-validator :**

Provides train/test indices to split data in train/test sets.This cross-validation object is a variation of KFold that returns stratified folds. The folds are made by preserving the percentage of samples for each class.



**Figure 15: ipynb code for train the model**

### XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks.

XGBoost provides a wrapper class to allow models to be treated like classifiers or regressors in the scikit-learn framework.This means we can use the full scikit-learn library with XGBoost models.The XGBoost model for classification is called **XGBClassifier**. We can create and and fit it to our training dataset. Models are fit using the scikit-learn API and the **model.fit()** function.

Grid searching is a method to find the best possible combination of hyper-parameters at which the model achieves the highest accuracy. Before applying Grid Searching on any algorithm, Data is used to divided into training and validation set, a validation set is used to validate the models. A model with all possible combinations of hyperparameters is tested on the validation set to choose the best combination.

### Model Evaluation

After training the model, the model should be tested by using the test data which has been separated while splitting the data for checking the functionality of the model.

**Regression Evaluation Metrics:**

These model evaluation techniques are used to find out the accuracy of models built in classification type of machine learning models.

* Accuracy\_score
* Confusion matrix

1. Accuracy\_score

* It is the ratio of number of correct predictions to the total number of input samples.

2.  Confusion Matrix

* It is a matrix representation of the results of any binary testing

For testing the model we use the below method,



**Figure 16: ipynb code for testing the accuracy.**

**Save The Model**

After building the model we have to save the model.

**Pickle**in **Python** is primarily used in serializing and deserializing a **Python** object structure. In other words, it's the process of converting a **Python** object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network. wb indicates write method and rd indicates read method.

This is done by the below code



**Figure 17: ipynb code for saving the model**

### 6.2 HTML CODE AND PYTHON CODE

### Application Building:

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server side script

**Build HTML Code:**

In this HTML page, we will create the front end part of the web page. In this page we will accept input from the user and Predict the values.

In our project we have 3 HTML files,

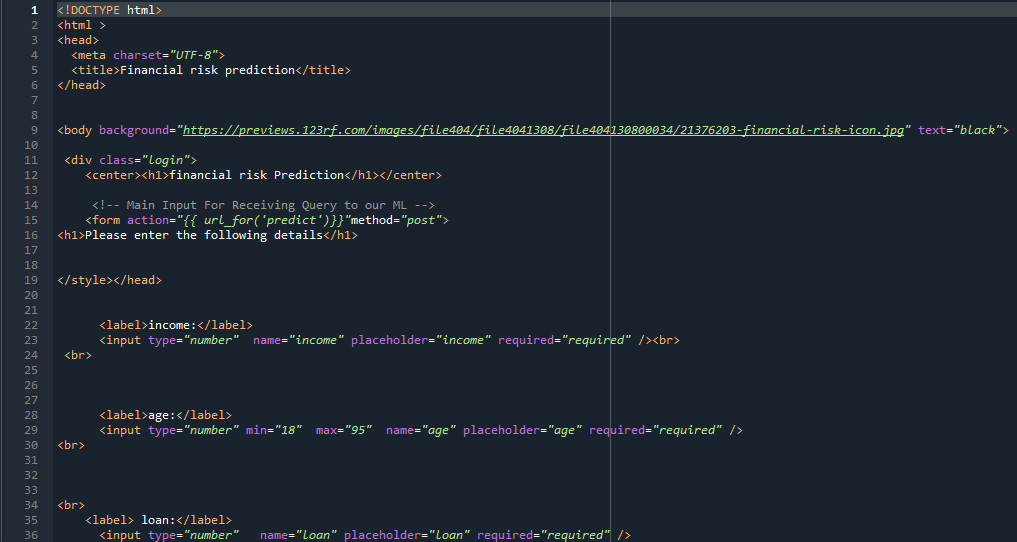
they are:

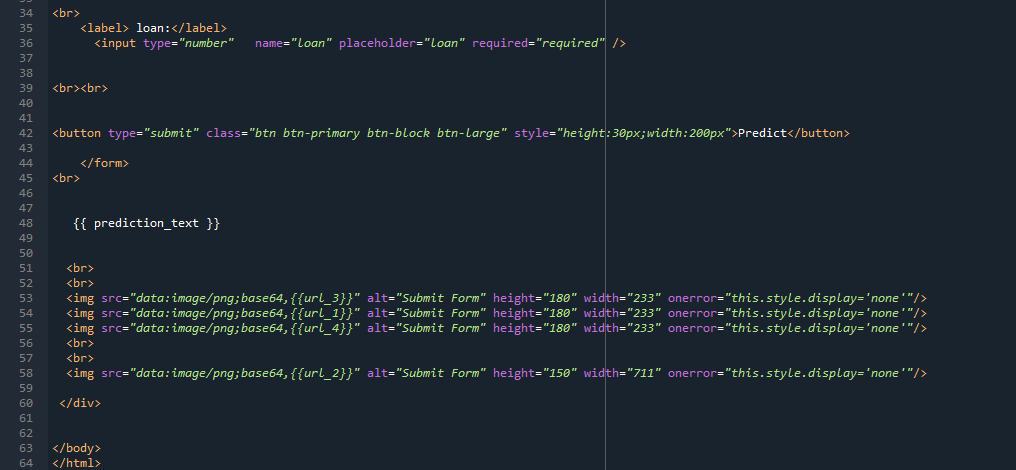
1. index.html

2. predgood.html

3. predbad.html

**index.html**



****

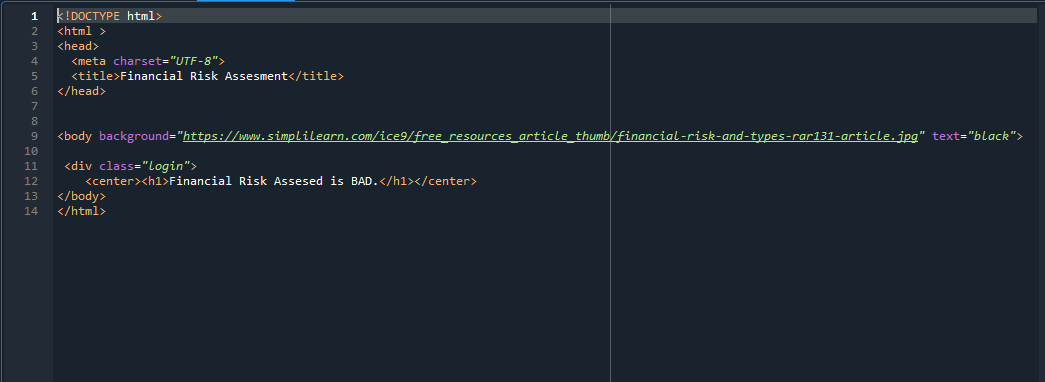
**Figure 18: index.html page code for index page of the web application**

**predgood.html**

****

**Figure 19: predgood.html is the page which displays that no financial risk.**

**Predbad.html**

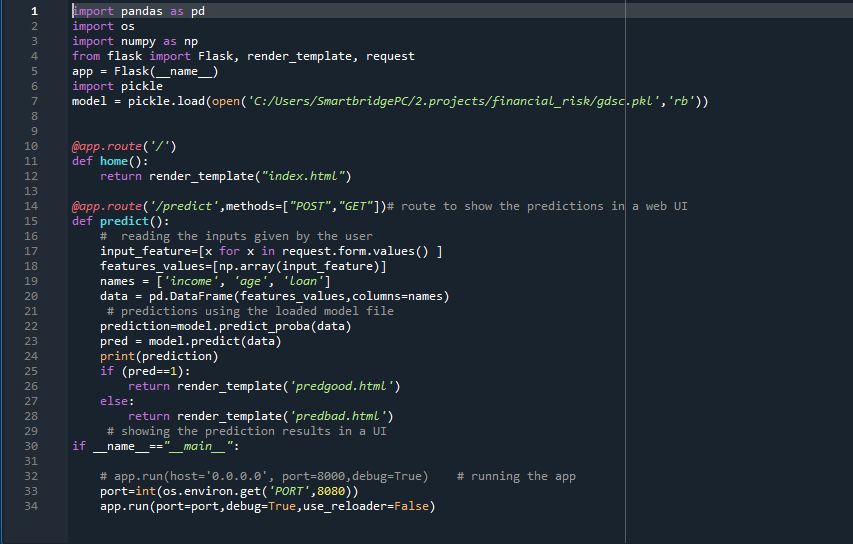
****

**Figure 20: predbad.html is the page which displays that there is a financial risk**

**App.py**

App.py flask file which is a web framework written in python for server-side scripting. Let’s see the step by step procedure for building the backend application.

In order to develop web api with respect to our model, we basically use the Flask framework which is written in python.

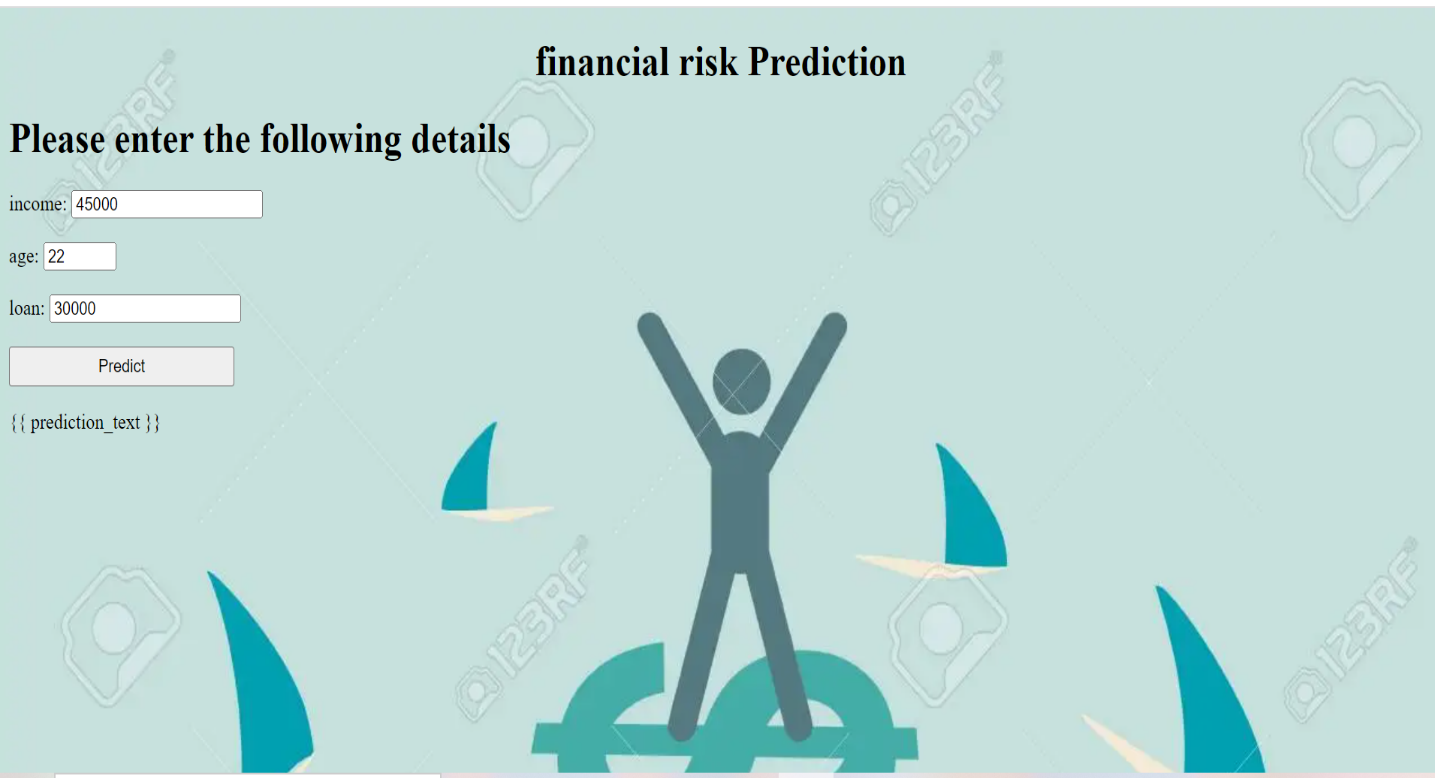
****

**Figure 21: python code used for rendering all the HTML pages**

### Run The App

* Open anaconda prompt  from the start menu
* Navigate to the folder where your python script is.
* Now type “python app.py” command
* Navigate to the localhost where you can view your web page.

**7.UI OUTPUT**

****

**Figure 22: index page (which takes inputs from user)**

****

**Figure 23: output page (displays that there is no financial risk)**



**Figure 24: output page(displays that there is a financial risk)**

**8.ADVANTAGES**

* Scalability: The ML model can easily be scaled up to meet the requirement.
* It is quite cost effective as it requires very little capital and can take decisions much faster.
* The ML model can easily be changed or modified to accommodate changes in the market.
* The automatization of credit risk testing ultimately reduces losses for banks.

**9.DISADVANTAGES**

* When using an algorithm, it can be difficult to trust the decision as there is less transparency.
* Even with the best algorithm, there are still many human elements to take into consideration.
* The effectiveness of an algorithm can reduce with changes in economic factors and will constantly need to be updated and reviewed.

**10.APPILICATIONS**

Our model can help financial institutions determine the credit worthiness of potential customers. By analyzing past spending behaviour and patterns, a system could identify how much credit should be extended to a given customer.

The technology would be especially useful in the case of new customers or those who lack a long credit history, i.e. millennials. Automating credit and risk scoring processes on a mass scale can help banks enhance their credit and risk scoring models across the board.

**11.FUTURE SCOPE**

For accepting or rejecting a loan, a credit scoring model is a tool which is used for decision-making. A credit scoring model is the classification statistical model which is based on the borrower's information, it allows one to distinguish between “good” or “bad” loans. It is just one of the factors used for evaluating a credit application of the borrowers.

Although credit scoring methods are linked applications in banking and finance, they can be widely used in a large variety of other data analytics problems, such as:

Factors which can contribute to a consumer’s choice?

What are some important factors which can generate the biggest impact to a consumer’s choice?

With a further boost in each of the impact factors what profit can be associated?

Will customer be willing to adopt a new service?

Such questions can all be answered within the same classification statistical Model

**12.CONCLUSION**

The main objective of this project was to build machine learning algorithms that would be able to identify potential defaulters and therefore reduce company loss. The best model possible would be the one that could minimize false negatives, identifying all defaulters among the client base, while also minimizing false positives, preventing clients to be wrongly classified as defaulters. Meeting these requirements can be quite difficult as there is a trade-off between precision and recall, meaning that increasing the value of one of these metrics often decreases the value of the other. Considering the importance of minimizing company loss, we decided to give more emphasis on reducing false positives, searching for the best hyperparameters that could increase the recall rate.

**13.BIBLIOGRAPHY**

* Exploring the potential of machine learning: How machine learning can support financial risk management

- Ohlsson, Caroline

Uppsala University, Disciplinary Domain of Humanities and Social Sciences, Faculty of Social Sciences, Department of Business Studies.

* Interaction between financial risk measures and machine learning methods.

- Jun-ya Gotoh, Akiko Takeda & Rei Yamamoto

# Machine Learning in Banking Risk Management

by [Martin Leo](https://sciprofiles.com/profile/535550), [Suneel Sharma](https://sciprofiles.com/profile/855200) and [K. Maddulety](https://sciprofiles.com/profile/652449)

SP Jain School of Global Management, Sydney 2127, Australia

* Machine Learning: A Revolution in Risk Management and Compliance?

Bart van Liebergen – Associate Policy Advisor, Institute of International Finance