# Prediction of Modernized Loan Approval System Based on Machine Learning Approach

**TRAINING REPORT**

**At**

**Sasi Institute of Technology and Engineering**

**(AUTONOMOUS)**

Submitted in partial fulfillment of the requirements for the award of Bachelor

of Engineering Degree in Computer Science and Engineering

(Data Science) BY

NAME: Kanikireddy Pavan Teja Reddy

REGNO: 22K61A4429

NAME: Yadamakanthi N.V.Nithin Reddy

REGNO: 22K61A4458

NAME: Kudipudi Komali Srinuja

REGNO: 22K61A4438

NAME: Namburi Pranav Gupta

REGNO: 22K61A4442

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)**

** SASI INSTITUTE OF TECHNOLOGY AND ENGINEERING**

**TADEPALLIGUDEM-534101,WEST GODAVARI,ANDHRAPRADESH**

# SASI INSTITUTE OF TECHNOLOGY

# AND ENGINEERING

**Accredited with Grade “A+” by NAAC**

**TADEPALLIGUDEM-534101,WEST GODAVARI,ANDHRAPRADESH**

### www.sasi.ac.in

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(Data Science)**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the Bonafide work of **KANIKIREDDY PAVAN TEJA REDDY(Reg.no.22K61A4429),YADAMAKANTHI N.V.NITHIN REDDY (Reg.no. 22K61A4458), KUDIPUDI KOMALI SRINUJA (Reg.no.22K61A4438) And NAMBURI PRANAV GUPTA(22K61A4442)** and who carried out the project entitled **“Prediction of**

**Modernized Loan Approval System Based on Machine Learning Approach”** under our supervision from September 2024 to November 2024.

**Internal Guide**

**PRASANTH KUMAR**

**Head of the Department**

**Dr.K.N.S Prasad Ph.D.**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to Board of Management of **SASI** **INSTITUTE OF TECHNOLOGY AND ENGINEERING** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey thanks to **K.Prasanth Kumar,and Dr.K.N.S.Prasad Ph.D., Head of the Department**, Department of Computer Science Engineering and Data Science for providing us the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **K.Prasanth Kumar,** for his valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project.

I wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **COMPUTER SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

## ABSTRACT

Technology has boosted the existence of human kind the quality of life they live. Every day we are planning to create something new and different. We have a solution for every other problem we have machines to support our lives and make us somewhat complete in the banking sector candidate gets proofs/ backup before approval of the loan amount. The application approved or not approved depends upon the historical data of the candidate by the system. Every day lots of people applying for the loan in the banking sector but Bank would have limited funds. In this case, the right prediction would be very beneficial using some classes-function algorithm. An example the logistic regression, random forest classifier, support vector machine classifier, etc. A Bank's profit and loss depend on the amount of the loans that is whether the Client or customer is paying back the loan. Recovery of loans is the most important for the banking sector. The improvement process plays an important role in the banking sector. The historical data of candidates was used to build a machine learning model using different classification algorithms. The main objective of this paper is to predict whether a new applicant granted the loan or not using machine learning models trained on the historical data set

**INTRODUCTION**

**Aim:** To determine the loan approval system using machine learning algorithms.

**Synopsis:**

Loan approval is a very important process for banking organizations. The systems 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. In recent years many researchers worked on loan approval prediction systems. Machine Learning (ML) techniques are very useful in predicting outcomes for large amount of data. In this paper different machine learning algorithms are applied to predict the loan approval of customers..In this paper, various machine learning algorithms that have been used in past are discussed and their accuracy is evaluated. The main focus of this paper is to determine whether the loan given to a particular person or an organization shall be approved or not.

**SYSTEM ANALYSIS**

## 2.1 EXISTING SYSTEM

The enhancement in the banking sector lots of people are applying for bank loans but the bank has its limited assets which it has to grant to limited people only, so finding out to whom the loan can be granted which will be a safer option for the bank is a typical process. In existing process, they are use RF algorithm in loan approval system. But the efficiency and accuracy was pretty low. Already banks are provide online transaction system, online bank account opening system, etc,. But there is no loan approval system in the banking sector. Then now we create a new system for loan approval. So now we move on to the proposed system.

## 2.2 PROPOSED SYSTEM

The proposed model focuses on predicting the credibility of customers for loan repayment by analyzing their details. The input to the model is the customer details collected. On the output from the classifier, decision on whether to approve or reject the customer request can be made. Using different data analytics tools loan prediction and there severity can be forecasted. In this process it is required to train the data using different algorithms and then compare user data with trained data to predict the nature of loan. The training data set is now supplied to machine learning model; on the basis of this data set the model is trained. Every new applicant details filled at the time of application form acts as a test data set. After the operation of testing, model predict whether the new applicant is a fit case for approval of the loan or not based upon the inference it conclude on the basis of the training data sets. By providing real time input on the web app. In our project, Logistic Regression gives high accuracy level compared with other algorithms. Finally, we are predicting the result via data visualization and display the predicted output using web app using flask.

**REQUIREMENT SPECIFICATIONS**

## 3.1 INTRODUCTION

Prediction of modernized loan approval system based on machine learning approach is a loan approval system from where we can know whether the loan will pass or not. In this system, we take some data from the user like his monthly income, marriage status, loan amount, loan duration, etc. Then the bank will decide according to its parameters whether the client will get the loan or not. So there is a classification system, in this system, a training set is employed to make the model and the classifier may classify the data items into their appropriate class. A test dataset is created that trains the data and gives the appropriate result that, is the client potential and can repay the loan. Prediction of a modernized loan approval system is incredibly helpful for banks and also the clients. This system checks the candidate on his priority basis. Customer can submit his application directly to the bank so the bank will do the whole process, no third party or stockholder will interfere in it. And finally, the bank will decide that the candidate is deserving or not on its priority basis. The only object of this research paper is that the deserving candidate gets straight forward and quick results.

#### 3.2 Machine learning

**Introduction:**

Machine learning (ML) is the [scientific study](https://en.wikipedia.org/wiki/Branches_of_science) of [algorithms](https://en.wikipedia.org/wiki/Algorithm) and [statistical models](https://en.wikipedia.org/wiki/Statistical_model) that [computer systems](https://en.wikipedia.org/wiki/Computer_systems) use to perform a specific task without using explicit instructions, relying on patterns and [inference](https://en.wikipedia.org/wiki/Inference) instead. It is seen as a subset of [artificial intelligence.](https://en.wikipedia.org/wiki/Artificial_intelligence) Machine learning algorithms build a [mathematical model](https://en.wikipedia.org/wiki/Mathematical_model) based on sample data, known as "[training data"](https://en.wikipedia.org/wiki/Training_data), in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as [email filtering](https://en.wikipedia.org/wiki/Email_filtering) and [computer vision,](https://en.wikipedia.org/wiki/Computer_vision) where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to [computational statistics,](https://en.wikipedia.org/wiki/Computational_statistics) which focuses on making predictions using computers. The study of [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization) delivers methods, theory and application domains to the field of machine learning. [Data mining](https://en.wikipedia.org/wiki/Data_mining) is a field of study within machine learning, and focuses on [exploratory data analysis](https://en.wikipedia.org/wiki/Exploratory_data_analysis) through learning. In its application across business problems, machine learning is also referred to as [predictive analytics.](https://en.wikipedia.org/wiki/Predictive_analytics)

**Machine learning tasks:**

Machine learning tasks are classified into several broad categories. In [supervised learning,](https://en.wikipedia.org/wiki/Supervised_learning) the algorithm builds a [mathematical model](https://en.wikipedia.org/wiki/Mathematical_model) from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the [training data](https://en.wikipedia.org/wiki/Training_data) for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. Semi algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.

[Classification](https://en.wikipedia.org/wiki/Statistical_classification) algorithms and [regression](https://en.wikipedia.org/wiki/Regression_analysis) algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a [limited set](https://en.wikipedia.org/wiki/Discrete_number) of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "[spam"](https://en.wikipedia.org/wiki/Email_spam) or "not spam", represented by the [Boolean](https://en.wikipedia.org/wiki/Boolean_data_type) values true and false. [Regression](https://en.wikipedia.org/wiki/Regression_analysis) algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object.

In [unsupervised learning,](https://en.wikipedia.org/wiki/Unsupervised_learning) the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in [feature learning.](https://en.wikipedia.org/wiki/Feature_learning) [Dimensionality reduction](https://en.wikipedia.org/wiki/Dimensionality_reduction) is the process of reducing the number of "[features"](https://en.wikipedia.org/wiki/Feature_(machine_learning)), or inputs, in a set of data.

[Active learning](https://en.wikipedia.org/wiki/Active_learning_(machine_learning)) algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. [Reinforcement learning](https://en.wikipedia.org/wiki/Reinforcement_learning) algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used in [autonomous vehicles](https://en.wikipedia.org/wiki/Autonomous_vehicle) or in learning to play a game against a human opponent. Other specialized algorithms in machine learning include [topic modeling,](https://en.wikipedia.org/wiki/Topic_modeling) where the computer program is given a set of [natural language](https://en.wikipedia.org/wiki/Natural_language) documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservableprobability density functio[n](https://en.wikipedia.org/wiki/Probability_density_function) indensity estimatio[n](https://en.wikipedia.org/wiki/Density_estimation) problems. [Meta learning](https://en.wikipedia.org/wiki/Meta_learning_(computer_science)) algorithms learn their owninductive bia[s](https://en.wikipedia.org/wiki/Inductive_bias) based on previous experience. In [developmental robotics,](https://en.wikipedia.org/wiki/Developmental_robotics) [robot learning](https://en.wikipedia.org/wiki/Robot_learning) algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

**Types of learning algorithms**:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

**Supervised learning:**

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as [training data,](https://en.wikipedia.org/wiki/Training_data) and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an [array](https://en.wikipedia.org/wiki/Array_data_structure) or vector, sometimes called a feature vector, and the training data is represented by a [matrix.](https://en.wikipedia.org/wiki/Matrix_(mathematics)) Through iterative optimization of an [objective function,](https://en.wikipedia.org/wiki/Loss_function) supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data.

Supervised learning algorithms include [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression.](https://en.wikipedia.org/wiki/Regression_analysis) Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range.

**Unsupervised learning**:

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of [density estimation](https://en.wikipedia.org/wiki/Density_estimation) in [statistics,](https://en.wikipedia.org/wiki/Statistics) though unsupervised learning encompasses other domains involving summarizing and explaining data features.

**Semi-supervised learning:**

Semi-supervised learning falls between [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning) (without any labeled training data) and [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

##### 4.1 Design and Implementation Constraints

###### 4.5.1 Constraints in Analysis

♦ Constraints as Informal Text

♦ Constraints as Operational Restrictions

♦ Constraints Integrated in Existing Model Concepts

♦ Constraints as a Separate Concept

♦ Constraints Implied by the Model Structure

###### 4.1.2 Constraints in Design

♦ Determination of the Involved Classes

♦ Determination of the Involved Objects

♦ Determination of the Involved Actions

♦ Determination of the Require Clauses

♦ Global actions and Constraint Realization

###### 4.1.3 Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

##### 4.2 Other Nonfunctional Requirements

###### 4.2.1 Performance Requirements

The application at this side controls and communicates with the following three main general components.

⮚ embedded browser in charge of the navigation and accessing to the web service;

⮚ Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

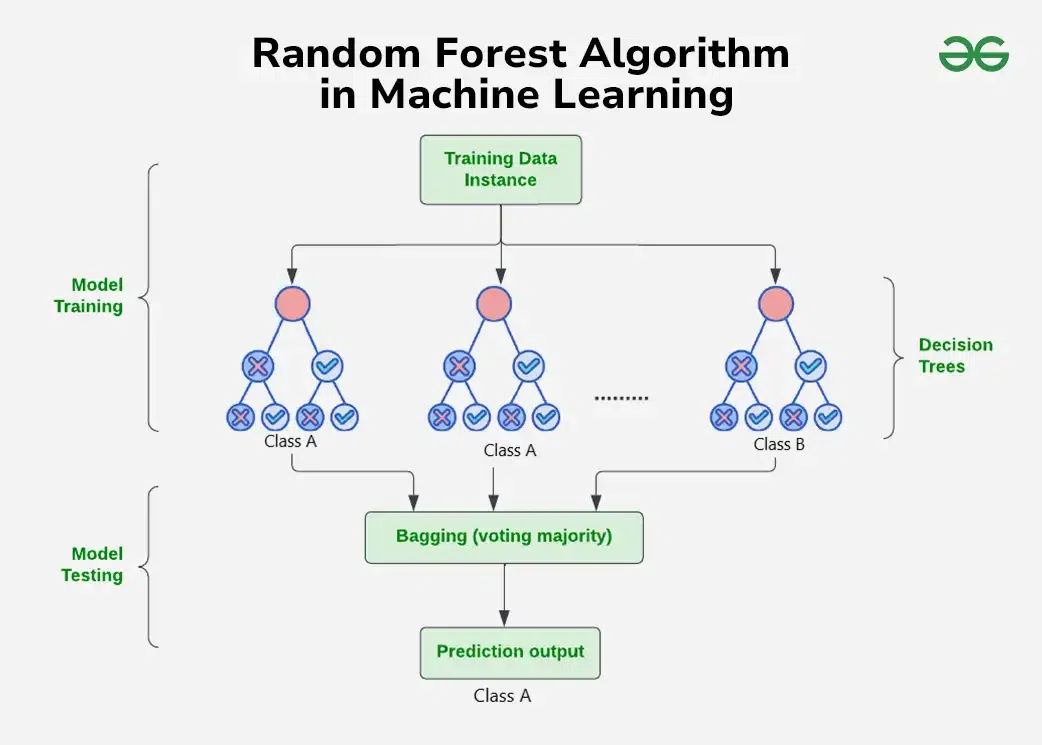
Random Forest Allgorithm

A Random Forest is a collection of decision trees that work together to make predictions. In this article, we'll explain how the Random Forest algorithm works and how to use it.

**Understanding Intuition for Random Forest Algorithm**

Random Forest algorithm is a powerful tree learning technique in Machine Learning to make predictions and **then we do voting of all the tress to make prediction**. They are widely used for classification and regression task.

* It is a type of classifier that uses many decision trees to make predictions.
* It takes different random parts of the dataset to train each tree and then it combines the results by averaging them. This approach helps improve the accuracy of predictions. Random Forest is based on [ensemble learning](https://www.geeksforgeeks.org/a-comprehensive-guide-to-ensemble-learning/).



**Key Features of Random Forest**

* **Handles Missing Data**: Automatically handles missing values during training, eliminating the need for manual imputation.
* Algorithm ranks **features based on their importance in making predictions** offering valuable insights for feature selection and interpretability.
* **Scales Well with Large and Complex Data** without significant performance degradation.
* Algorithm is versatile and can be applied to both classification tasks (e.g., predicting categories) and regression tasks (e.g., predicting continuous values).

**How Random Forest Algorithm Works?**

The random Forest algorithm works in several steps:

* Random Forest builds **multiple decision trees using random samples of the data**. **Each tree is trained on a different subset of the data which makes each tree unique**.
* When creating each tree the **algorithm randomly selects a subset of features or variables to split the data rather than using all available features at a time. This adds diversity to the trees.**
* Each decision tree in the forest **makes a prediction based on the data it was trained on. When making final prediction random forest combines the results from all the trees.**
  + For classification tasks the final prediction is decided by a majority vote. This means that the category predicted by most trees is the final prediction.
  + For regression tasks the final prediction is the average of the predictions from all the trees.
* The **randomness in data samples and feature selection helps to prevent the model from overfitting making the predictions more accurate and reliable.**

**Assumptions of Random Forest**

* **Each tree makes its own decisions: Every tree in the forest makes its own predictions without relying on others.**
* **Random parts of the data are used: Each tree is built using random samples and features to reduce mistakes.**
* **Enough data is needed: Sufficient data ensures the trees are different and learn unique patterns and variety.**
* **Different predictions improve accuracy: Combining the predictions from different trees leads to a more accurate final results.**

**Logistic Regression in Machine Learning**

In our previous discussion, we explored the fundamentals of machine learning and walked through a hands-on implementation of **Linear Regression**. Now, let’s take a step forward and dive into one of the first and most widely used classification algorithms — **Logistic Regression**

**What is Logistic Regression?**

**Logistic regression** is a **supervised machine learning algorithm**used for **classification tasks** where the goal is to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyze the relationship between two data factors. The article explores the fundamentals of logistic regression, it’s types and implementations.

Logistic regression is used for binary [classification](https://www.geeksforgeeks.org/getting-started-with-classification/) where we use [sigmoid function](https://www.geeksforgeeks.org/derivative-of-the-sigmoid-function/), that takes input as independent variables and produces a probability value between 0 and 1.

For example, we have two classes Class 0 and Class 1 if the value of the logistic function for an input is greater than 0.5 (threshold value) then it belongs to Class 1 otherwise it belongs to Class 0. It’s referred to as regression because it is the extension of[linear regression](https://www.geeksforgeeks.org/ml-linear-regression/) but is mainly used for classification problems.

**Key Points:**

* Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.
* It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
* In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).

**Types of Logistic Regression**

On the basis of the categories, Logistic Regression can be classified into three types:

1. **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
2. **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as “cat”, “dogs”, or “sheep”
3. **Ordinal:**In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as “low”, “Medium”, or “High”

**Assumptions of Logistic Regression**

We will explore the assumptions of logistic regression as understanding these assumptions is important to ensure that we are using appropriate application of the model. The assumption include:

1. Independent observations: Each observation is independent of the other. meaning there is no correlation between any input variables.
2. Binary dependent variables: It takes the assumption that the dependent variable must be binary or dichotomous, meaning it can take only two values. For more than two categories SoftMax functions are used.
3. Linearity relationship between independent variables and log odds: The relationship between the independent variables and the log odds of the dependent variable should be linear.
4. No outliers: There should be no outliers in the dataset.
5. Large sample size: The sample size is sufficiently large

###### Safety Requirements

1. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
2. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
3. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
4. Systems with different requirements for safety levels must be separated.
5. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

**Architecture Diagram:**



Dataset

Preprocessing

User Input

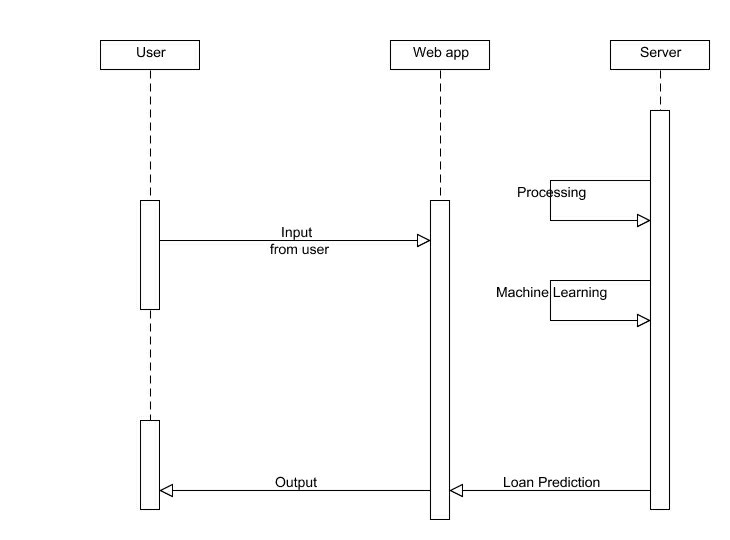
Web

app



**Sequence Diagram:**

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



**5.3 Use Case Diagram:**

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

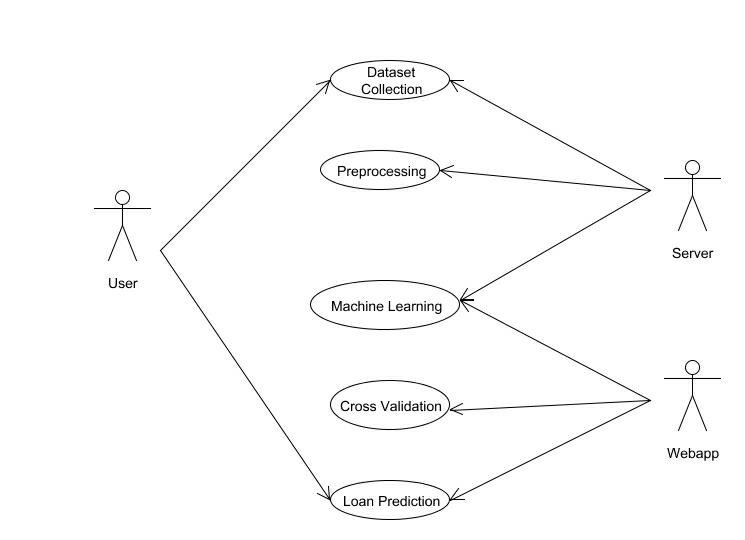
###### 5.3.1. USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

**Use case:** A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

**Actor:** An actor is a person, organization or external system that plays a role in one or more interaction with the system.

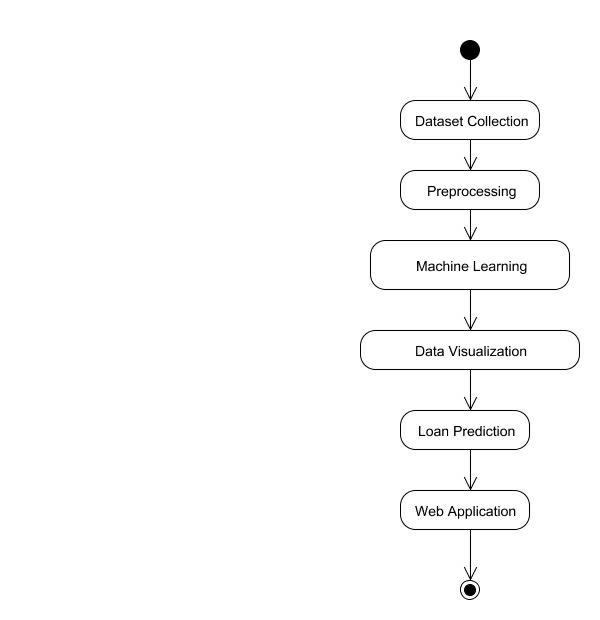


**5.4 Activity Diagram:**

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

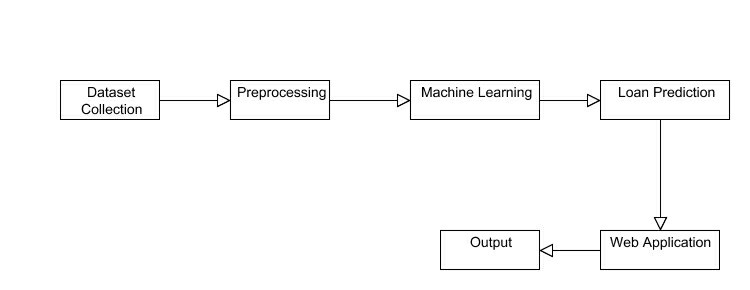
The most important shape types:

* Rounded rectangles represent activities.
* Diamonds represent decisions.
* Bars represent the start or end of concurrent activities.
* A black circle represents the start of the workflow.
* An encircled circle represents the end of the workflow.



**5.5 Collaboration Diagram:**

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.



##### 6.1 MODULES

⮚ **Dataset collection**

⮚ **Machine Learning Algorithm**

###### ⮚ Prediction

**6.2 MODULE EXPLANATION:**

**6.2.1 Dataset collection:**

Dataset is collected from the kaggle.com. That dataset have some value like gender, marital status, self-employed or not, monthly income, etc,. Dataset has the information, whether the previous loan is approved or not depends up on the customer information. That data well be preprocessed and proceed to the next step.

**Machine learning Algorithm:**

In this stage, the collected data will be given to the machine algorithm for training process. We use multiple algorithms to get high accuracy range of prediction. A preprocessed dataset are processed in different machine learning algorithms. Each algorithm gives some accuracy level. Each one is undergoes for the comparison.

###### ✔ Logistic Regression ✔ Random Forest

**Prediction:**

Preprocessed data are trained and input given by the user goes to the trained dataset. The Logistic Regression trained model is used to predict and determine whether the loan given to a particular person shall be approved or not.

Once the design aspect of the system is finalizes the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

##### 7.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format

Exception and error handling

###### 7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each

subsequent letters of the new words in uppercase and the rest of letters in lowercase.

###### 7.2.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

⮚ Proper default values for the variables.

⮚ Proper validation of values in the field.

⮚ Proper documentation of flag values.

###### 7.2.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

###### 7.2.4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

⮚ X – User has performed illegal operation.

⮚ ! – Information to the user.

##### 7.3 TEST PROCEDURE SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

⮚ Static analysis is used to investigate the structural properties of the Source code.

⮚ Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

##### TEST DATA AND OUTPUT

###### UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

###### 7.4.2 FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

⮚ Performance Test

⮚ Stress Test

⮚ Structure Test

###### 7.4.3 PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

###### STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

###### STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

⮚ Exercise all logical decisions on their true or false sides.

⮚ Execute all loops at their boundaries and within their operational bounds.

⮚ Exercise internal data structures to assure their validity.

⮚ Checking attributes for their correctness.

⮚ Handling end of file condition, I/O errors, buffer problems and textual errors in output information

###### INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

##### TESTING TECHNIQUES / TESTING STRATERGIES

###### TESTING

Testing is a 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 as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

**SOFTWARE TESTING STRATEGIES:**

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

⮚ Testing begins at the module level and works “outward” toward the integration of the entire computer based system.

⮚ Different testing techniques are appropriate at different points in time.

⮚ Th e developer of the software and an independent test group conducts testing.

⮚ Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

**INTEGRATION TESTING:**

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing.

There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

**PROGRAM TESTING:**

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

**SECURITY TESTING:**

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

7.5.2.4 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software testvalidation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

* The function or performance characteristics confirm to specifications and are accepted.
* A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

7.5.2.5 USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

* Input screen design.
* Output screen design.

##### REFERENCES

1] Amruta S. Aphale and R. Prof. Dr. Sandeep. R Shinde, “Predict Loan Approval in Banking

System Machine Learning Approach for Cooperative Banks Loan Approval”, International

Journal of Engineering Trends and Applications (IJETA), vol. 9, issue 8, 2020)

1. Loan Prediction Using Ensemble Technique, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 3, March 2016

1. Exploratory data analysis https://en.wikipedia.org/wiki/Exploratory\_data\_analysis

1. Pandas Library https://pandas.pydata.org/pandas-docs/stable/

1. MeanDecreaseAccuracy https://dinsdalelab.sdsu.edu/metag.stats/code/randomforest.hml