Following ML Algorithms are going to be used in Loan prediction system:

1) Logistic regression.

2) Decision tree.

3) Random Forest.

Logistic Regression

I) Logistic regression is a classification algorithm which uses logistic function or sigmoid function which takes value in the range [0, 1].

II) Sigmoid function: f(x) = 1/(1+e^(-x))

III) Projecting extreme values i.e. f: [–inf,+inf] → [0,1].

IV) Logistic regression work much like Linear regression taking Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y=f(x)).

V) Equation of logistic regression:

$$y = e^{(b0 + b1*x)} / (1 + e^{(b0 + b1*x)}) = 1/(1 + e^{(-(b0 + b1*x))})$$

VI) Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in input data has an associated b coefficient (a constant real value).

VII) The coefficients (Beta values b) of the logistic regression algorithm must be estimated from your training data. This is done using maximum-likelihood estimation, which uses maxima-minima technique to find value of parameters such that error rates are as low as possible.

e.g.: Consider we want to create a model which predict loan to be pass or not on basis of Income.

We will find probability of Loan pass (1) and rejected (0) depending on Income.

i.e. p(x) =p(L=1|Income=value) = e^(b0 + b1*x) / (1 + e^(b0 + b1*x)) (loan pass when certain value limit in Income identified).

After simplifying this equation we get ln(p(x)/(1-p(x))) = b0+b1*x. After applying maximum likelihood on given function (since, p(x;b0,b1) = L(b0,b1;x)) using our train set we will find value of b0 and b1 and we will apply these values with value of x to find corresponding value of y in test set. We will consider Loan pass if f(x) >= 0.5 and rejected if f(x) < 0.5.

y = e^(b0 + b1*X) / (1 + e^(b0 + b1*X))

y = e^(-100 + 0.6*150) / (1 + e^(-100 + 0.6*X)) (consider, b0=-100 and b1=0.6)

f(x) = y = 0.0000453978687

since, f(x) < 0.5 Loan is Rejected.

Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

This algorithm split the node into n-nodes depending on best cost of split (lowest cost). Different type of cost function can be used, for classification decision tree we used Gini function given by: **G = sum(pk(1-pk))**

Here, pk is proportion of same class inputs present in a particular group. A perfect class purity occurs when a group contains all inputs from the same class, in which case pk is either 1 or 0 and G = 0, where as a node having a 50–50 split of classes in a group has the worst purity, so for a binary classification it will have pk = 0.5 and G = 0.5. We're going to use classification tree only. We set some condition to stop and yield output i.e. Leaf Node which reduces the complexity of tree and avoid overfitting.

Random Forest

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

A random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision.