` **Classification model:**

**Project Details:**

**a. Loan Status Prediction:**

i. This is a Property Loan Status Prediction dataset, we have the data of applicants who had previously applied for the loan based on the property and applicant-based metrics.

ii. The bank will decide whether to give a loan to the applicant based on some factors such as Applicant Income, Loan Amount, previous Credit History, Co-applicant Income, etc.

iii. The objective is to build a Machine Learning Model to predict the loan to be approved or to be rejected for an applicant.

iv. You are free to use any classification model such Logistic Regression, Decision Tree, RF or XGBoost.

v. Write the performance and key insights on the data in a word doc.

Objective:

Build a Machine Learning Model to predict the loan to be approved or to be rejected for an applicant. All the classification modules will be tried to predict the loan to be approved.

**Artificial Intelligence:**

* Artificial –means made by humans/a non-natural things
* Intelligence - ability to think/understand/decision

We wanted our computers/machine able to think or adopt decision making. Human having more experience can able to make a good decision, while machine doesn’t required experience but required data. Artificial Intelligence is developing a system that mimics humans to solve problems based on the data and rules provided to them.

Examples: Siri, customer service via chatbots, Expert Systems, Machine Translation like Google Translate, Intelligent humanoid robots such as Sophia and so on.

* AI is a broader family consisting of ML and DL as its components.

**Machine Learning (ML):**

* ML is the subset of Artificial Intelligence.
* ML model that can generalize pattern from the data and make prediction or decision without being explicitly programmed
* The goal is to learn from data on certain tasks to maximize the performance on that task.
* Examples: Google’s search algorithms, Banking fraud analysis, Stock price forecast

**Two main type of Machine Learning**

**Supervised learning**

**(Learning from Labelled Data)**

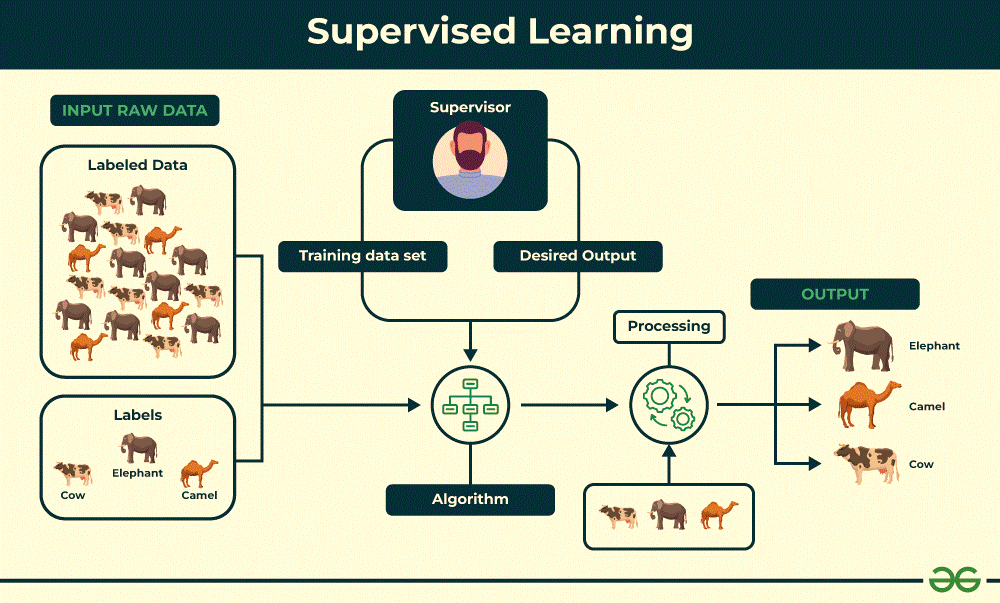
We train the machine using data that is well-labelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that the supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labeled data.

**Unsupervised learning**

**(Learning from Labelled Data)**

This means that the data does not have any pre-existing labels or categories. The goal of unsupervised learning is to discover patterns and relationships in the data without any explicit guidance.

**Supervised learning (Learning from Labelled Data)**



Supervisor

**Supervised learning is classified into two categories of algorithms:**

**Regression**

**Classification**

**Output variable/Target variable/Dependent variable is a categorical value, such as predicting disease like diabetes or no diabetes.**

**Classification Algorithms:**

**Logistic Regression: A binary classification algorithm that models the probability of an outcome.**

**k-Nearest Neighbors (k-NN): Used for multiclass classification by finding the nearest neighbors based on input features.**

**Decision Trees: Can handle both binary and multiclass problems by creating a tree-like structure.**

**Support Vector Machines (SVM): Suitable for binary or multiclass classification tasks.**

**Naive Bayes Classifier: Based on Bayes’ theorem, it’s commonly used for text classification.**

**Random Forest: An ensemble method combining multiple decision trees.**

**Gradient Boosting: Builds an ensemble of weak learners to improve accuracy.**

**Cost-sensitive Logistic Regression, Decision Trees, and Support Vector Machines: These address imbalanced classification problems.**

**Output variable/Target variable/Dependent variable is a real or continuous value, such as predicting “salary” or “weight”.**

**Regression helps economists and financial analysts in things ranging from asset valuation to making predictions.**

**Regression Algorithms:**

**Linear Regression: One of the simplest and widely used statistical models. It aims to find the best-fitting linear relationship between variables1.**

**Polynomial Regression: An extension of linear regression that fits a polynomial function to the data.**

**Stepwise Regression: Automates variable selection.**

**Ridge Regression: Analyzes multicollinearity.**

**Lasso Regression: Performs variable selection.**

**Elastic Net Regularization: Combines ridge and lasso techniques.**

**Bayesian Linear Regression: Based on Bayes’ theorem.**

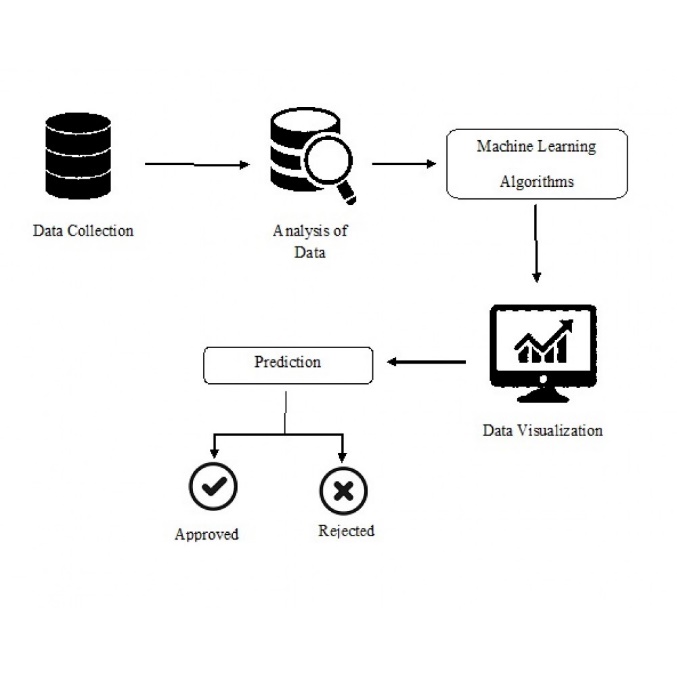
**Decision Tree Regression: Uses a flowchart-like structure.**

**Random Forest Regression: An ensemble of decision trees.**

**Support Vector Regression: A type of support vector machine for regression tasks.**

**​**

In this article we will be discussing about classification models. Classification model will be helpful to predict whether the loan to be approved or not, because the target variable is categorical type.



Reading Data:

Available data: Loan Status training data and Loan Status test csv data.

Pandas library: Used to read the data.

installation: pip install pandas

first step we need to import pandas library using command (import pandas as pd)

Both training data set and test data sets were imported using pandas library.

Data Preprocessing:

Identified the shape of test and trained data using .shape command.

It showed df\_shape: train (614, 13) test (367, 12).

614 rows and 13 columns in training data and 367 rows and 12 columns in test data.

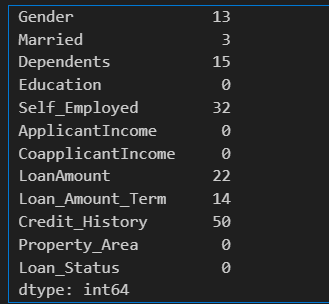
To analysis the data, data information was checked (train\_data.info ())

The below points have been noted:

1. Datatype = object in the Loan\_ID, Gender, Married, Dependents, Education, Self\_Employed, Property\_Area, Loan\_Status columns.
2. Datatype = int64 in ApplicantIncome
3. Datatype = float64 in CoapplicantIncome, LoanAmount, Loan\_Amount\_Term, Credit\_History
4. In training data Loan ID contain 614 unique value (train\_data.Loan\_ID.nunique()) which is not going to helpful in predicting the target value, hence Loan ID column was dropped.

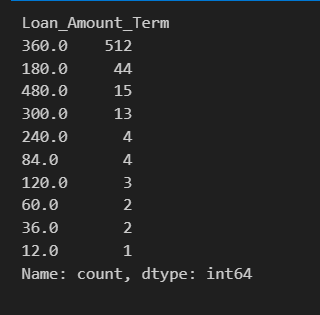
Handling of Null values:

To check the null values: train\_data.isnull().sum()



There are very few missing values in Gender, Married, Dependents, Credit\_History, and Self\_Employed features so we can fill them using the mode of the features.

For numerical variables: imputation using mean or median Fill the missing values in Loan\_Amount\_Term. Value count of the Loan amount term variable



the value of 360 is repeating the most. So we will replace the missing values in this variable using the mode of this variable.

LoanAmount variable: we can use mean or median to impute the missing values. We will use the median to fill the null values as earlier we saw that the loan amount has outliers . So the mean will not be the proper approach as it is highly affected by the presence of outliers.

The same will be followed for test\_data.

Converting text into numerical data:

As all the categorical values are binary so we can use Label Encoder for all such columns and the values will change into int datatype.

“from sklearn.preprocessing import LabelEncoder”

LabelEncoder used to convert the text data into numerical data.

**Model Training and Evaluation**

From Training data, Dependent variable (Loan\_Status) removed and assigned to X.

Dependent Variable (Loan\_Status) assented to y.

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

Training data was splitted into X\_train, X\_test, y\_train, y\_test using train\_test\_split

**Classification Models:**

The below list of models was trained:

KNeighborsClassifier - from sklearn.neighbors import KNeighborsClassifier

RandomForestClassifier - from sklearn.ensemble import RandomForestClassifier

SVC - from sklearn.svm import SVC

LogisticRegression - from sklearn.linear\_model import LogisticRegression

GradientBoostingClassifier - from sklearn.ensemble import GradientBoostingClassifier

DecisionTreeClassifier - from sklearn.tree import DecisionTreeClassifier

GaussianNB - from sklearn.naive\_bayes import GaussianNB

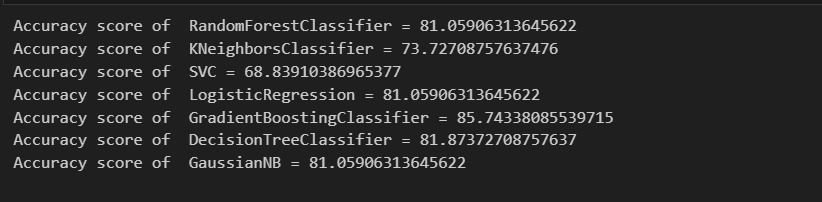
Same thing has been followed for test data

Evaluation:

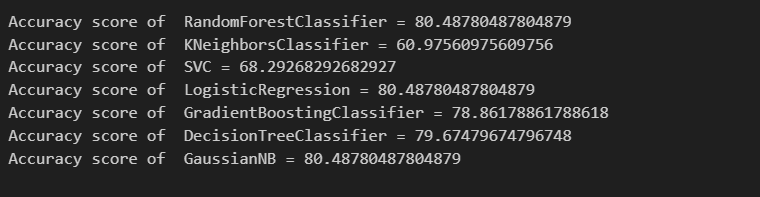
The below library used to evaluate the accuracy of the data.

“from sklearn.metrics import accuracy\_score”

**Accuracy Score in Training Data:**



**Accuracy Score in Test Data:**

****

**Conclusion:**

|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy Score in Training Data:** | **Accuracy Score in Test Data:** |
| KNeighborsClassifier | 81 | 80 |
| RandomForestClassifier | 74 | 61 |
| SVC | 69 | 68 |
| LogisticRegression | 81 | 80 |
| GradientBoostingClassifier | 86 | 79 |
| DecisionTreeClassifier | 82 | 80 |
| GaussianNB | 81 | 80 |

DecisionTreeClassifier accuracy score was noted as 82% training data but only 80% in test data.

While the Accuracy score in training was noted as 81% and test data was noted as 80% in three models.

1. KNeighborsClassifier
2. LogisticRegression
3. GaussianNB

Hence these models were performed well and considered as Good Models but considering Accuracy score 81% is not enough to make a good prediction.