Machine Learning - Overview

DR DV Ramana
Academic Advisor
And Program Manager

Agenda - Theory

- Exploratory Data Analysis
- Covariance & Correlation
- Random forest
- Gradient boosting
- Machines, Model Stacking
  - CAT Boost
  - XG Boost

Agenda -LAB

- Implement
  - Random Forest
  - SVM
  - Logistic regression
  - f1 score for all three algorithms.

Exploratory Data Analysis

Exploratory Data Analysis (EDA), also known as Data Exploration, is a step in the Data Analysis Process, where a number of techniques are used to better understand the dataset being used.

Understanding the dataset' can refer to a number of things including but not limited to…

- Extracting important variables and leaving behind useless variables
- Identifying outliers, missing values, or human error
- Understanding the relationship(s), or lack of, between variables
- Ultimately, maximizing your insights of a dataset and minimizing potential error that may occur later in the process

Exploratory Data Analysis does two main things

- It helps clean up a dataset.
- It gives you a better understanding of the variables and the relationships between them

Exploratory Data Analysis

Exploratory Data Analysis(EDA) - An exhaustive look at existing data from current and historical surveys conducted by a company

Exploratory Data Analysis(EDA) - To determine whether a predictive model is a viable analytical tool for a particular business problem, and if so, which type of modeling is most appropriate

Exploratory Data Analysis(EDA) - May reveal aspects of your company's performance that others may not have seen

Exploratory Data Analysis

Exploratory Data Analysis(EDA) - Why do it

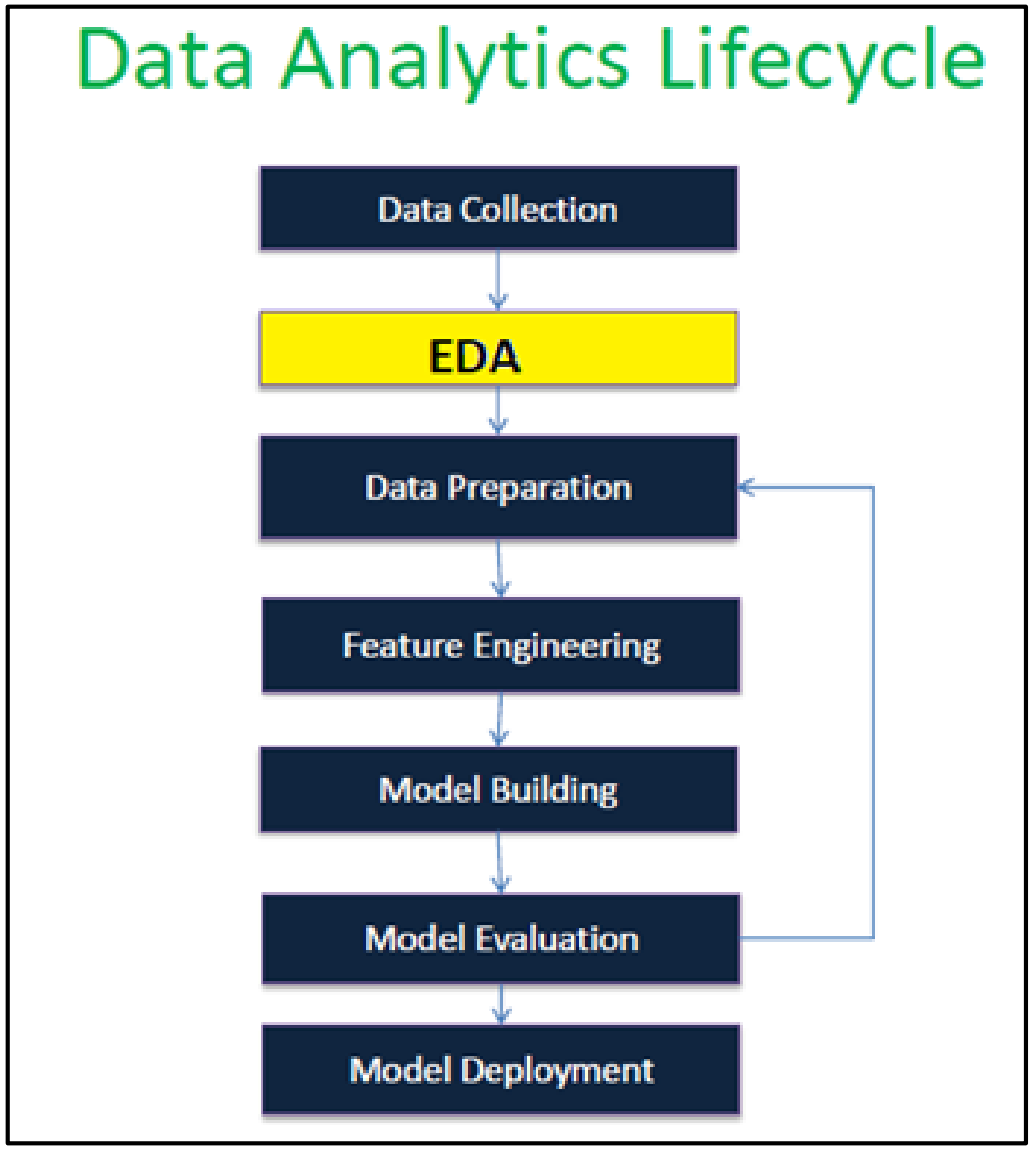
Exploratory Data Analysis(EDA) - Thorough examination meant to uncover the underlying structure of a data set and is important for a company because it exposes trends, patterns, and relationships that are not readily apparent

Exploratory Data Analysis(EDA) - To look at it carefully and methodically through an analytical lens

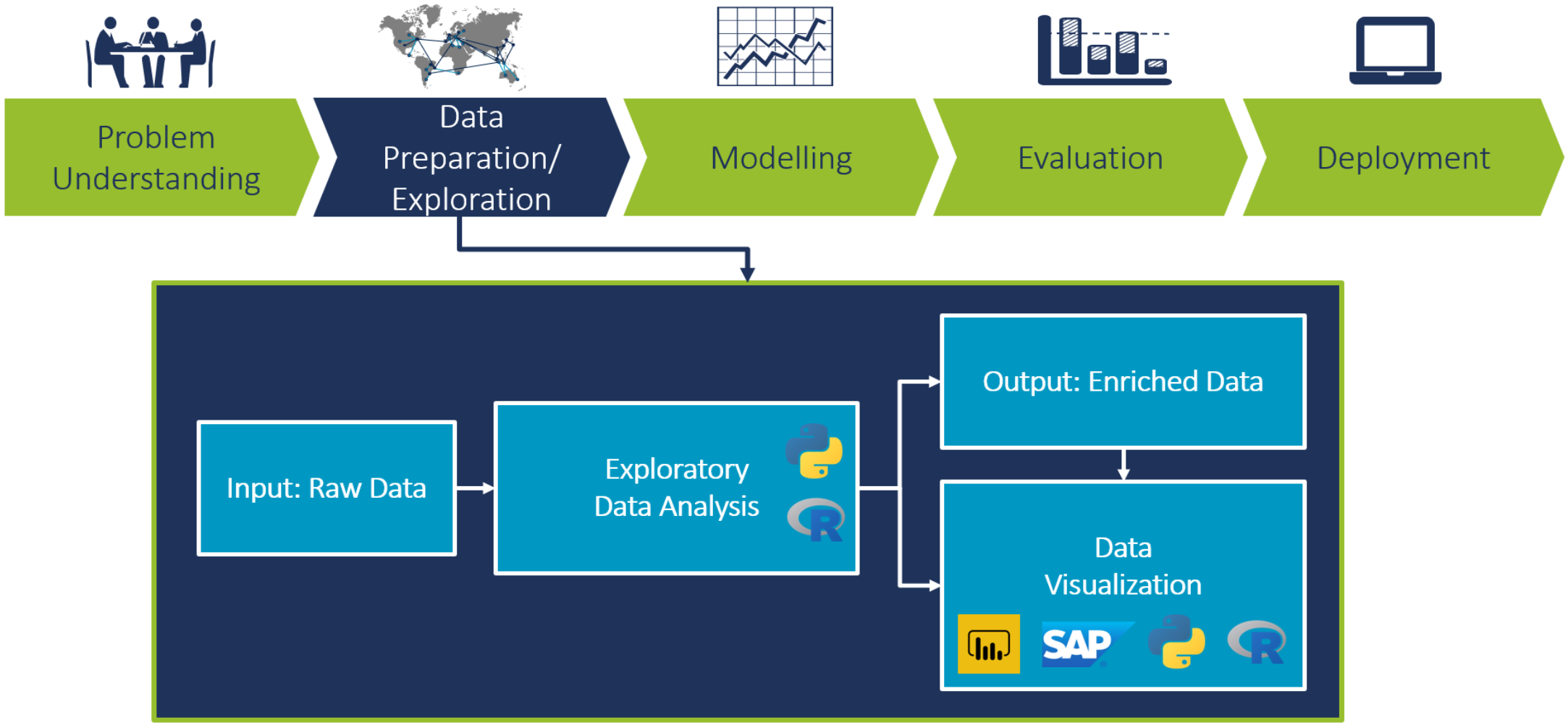
Exploratory Data Analysis(EDA) - Getting a “feel” for this critical information can help you detect mistakes, debunk assumptions, and understand the relationships between different key variables

Exploratory Data Analysis(EDA) - Such insights may eventually lead to the selection of an appropriate predictive model.

Exploratory Data Analysis



Exploratory Data Analysis



Exploratory Data Analysis -EDA checklist

- Did you specify the type of data analytic question (e.g. exploration, association causality) before touching the data?
- Did you define the metric for success before beginning?
- Did you understand the context for the question and the scientific or business application?
- Did you check for outliers? Where are the outliers?
- What kind of data do you have?
- What's missing from the data?
- What question are you trying to solve (or prove wrong)?
- What kind of data do you have?
- How can you add, change or remove features to get more out of your data?
- Is each variable one column?
- Is each observation one row?
- Do different data types appear in each table?
- Did you record the recipe for moving from raw to tidy data?
- Do different data types appear in each table?

Exploratory Data Analysis -EDA checklist

Checking the Data

Did you plot univariate and multivariate summaries of the data?

Did you check for outliers?

Did you identify the missing data code?

Exploratory Data Analysis -EDA checklist

Tidying the Data

Is each variable one column?

Is each observation one row?

Do different data types appear in each table?

Did you record the recipe for moving from raw to tidy data?

Do different data types appear in each table?

Exploratory Data Analysis -EDA checklist

Exploratory Analysis

- Did you identify missing values?
- Did you make univariate plots (histograms, density plots, boxplots)?
- Did you consider correlations between variables (scatterplots)?
- Did you check the units of all data points to make sure they are in the right range?
- Did you try to identify any errors or miscoding of variables?
- Did you consider plotting on a log scale?
- Would a scatterplot be more informative?

Exploratory Data Analysis -EDA checklist

Inference

Did you identify what large population you are trying to describe?

Did you clearly identify the quantities of interest in your model?

Did you consider potential confounders?

Did you identify and model potential sources of correlation such as measurements over time or space?

Did you calculate a measure of uncertainty for each estimate on the scientific scale?

Exploratory Data Analysis -EDA checklist

Prediction

Did you identify in advance your error measure?

Did you immediately split your data into training and validation?

Did you use cross validation, resampling, or bootstrapping only on the training data?

Did you create features using only the training data?

Did you estimate parameters only on the training data?

Did you fix all features, parameters, and models before applying to the validation data?

Did you apply only one final model to the validation data and report the error rate?

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an approach/philosophy for data analysis that employs a variety of techniques (mostly graphical) to

- Maximize insight into a data set
- Uncover underlying structure
- Extract important variables;
- Detect outliers and anomalies
- Develop parsimonious models
- determine optimal factor settings.

Exploratory Data Analysis (EDA)

EDA approach is precisely that--an approach--not a set of techniques, but an attitude/philosophy about how a data analysis should be carried out.

Maximize ins How to ensure you are ready to use machine learning algorithms in a project?

How to choose the most suitable algorithms for your data set?

How to define the feature variables that can potentially be used for machine learning?

Detect outliers and anomalies

Develop parsimonious models

determine optimal factor settings.

Exploratory Data Analysis (EDA) helps to answer all these questions, ensuring the best outcomes for the project

Exploratory Data Analysis (EDA) is an approach for summarizing, visualizing, and becoming intimately familiar with the important characteristics of a data set

Exploratory Data Analysis is valuable to data science projects since it allows to get closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired business contexts.

Exploratory Data Analysis (EDA)

Number of variables = total number of columns in the dataset

Number of observations=total number of rows

Missing values and duplicate values of the entire dataset

Various variable types (Numerical, Categorical, Boolean etc.)

Detect outliers and anomalies

Develop parsimonious models

determine optimal factor settings.

Exploratory Data Analysis (EDA) helps to answer all these questions, ensuring the best outcomes for the project

Exploratory Data Analysis (EDA) is an approach for summarizing, visualizing, and becoming intimately familiar with the important characteristics of a data set

Exploratory Data Analysis is valuable to data science projects since it allows to get closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired business contexts.

Things we need to keep in mind before performing EDA

Always ask the right questions

Performing EDA without proper questions may lead to disastrous result

Questions like which features are important?, which features are correlated to each other ? etc.

Have basic knowledge about problem domain:

Without knowing about the domain of problem it is useless ,meaningless and we will not find our answers.

if problem belonging to medical history of patient is given we need to understand some of the important terms used in determining patient health condition.

Never forget your objective

Never forget our objective whether is it classification or regression problem, we sometimes end up performing unnecessary tasks in EDA mainly when there are more number of features in a data set .

Data Cleaning

Data Cleaning means the process of identifying the incorrect, incomplete, inaccurate, irrelevant or missing part of the data and then modifying, replacing or deleting them according to the necessity.

Data cleaning is considered a foundational element of the basic data science

Inconsistent column

Missing data

Outliers

Duplicate rows:

Tidy data set

Converting data types:

String manipulation:

Data Concatenation

Data Cleaning is very much important for making your analytics and machine learning models error-free.

A small error in the dataset can cause you a lot of problem.

All your efforts can be wasted. So, always try to make your data clean.

Tidy Data is a way of structuring datasets to facilitate analysis.

Each variable must have its own column. Each observation must have its own row. Each value must have its own cell.

Methods of Exploratory Data Analysis

Univariate visualization — Provides summary statistics for each field in the raw data set

Bivariate visualization — Performed to find the relationship between each variable in the dataset and the target variable of interest

Multivariate Visualization — Performed to understand interactions between different fields in the dataset

Helps to understand the fields in the data that account for the most variance between observations and allow for the processing of a reduced volume of data

Why skipping Exploratory Data Analysis is a bad idea

Generating inaccurate models

Develop parsimonious models

Determine optimal factor settings.

Generating accurate models on the wrong data

Choosing the wrong variables for the model

Inefficient use of the resources, including the rebuilding of the model.

Key Concepts of Exploratory Data Analysis

2 types of Data Analysis

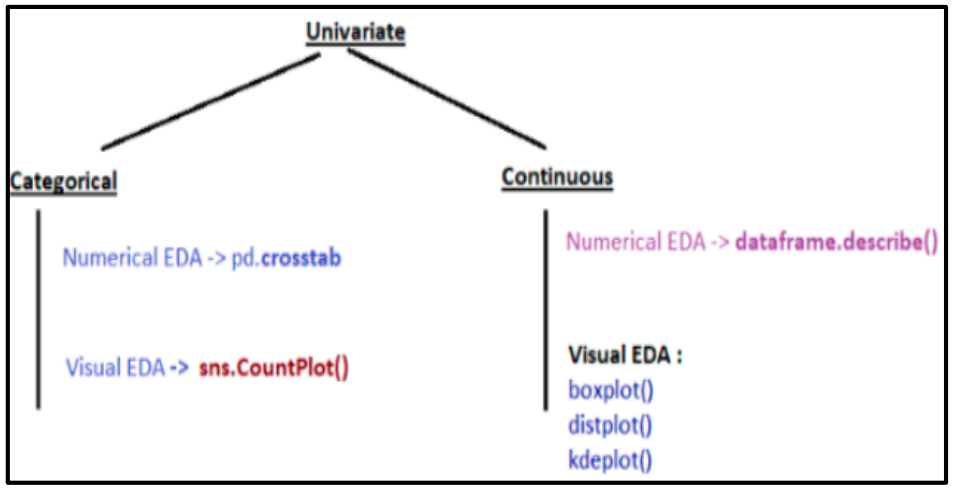
- Confirmatory Data Analysis
- Exploratory Data Analysis

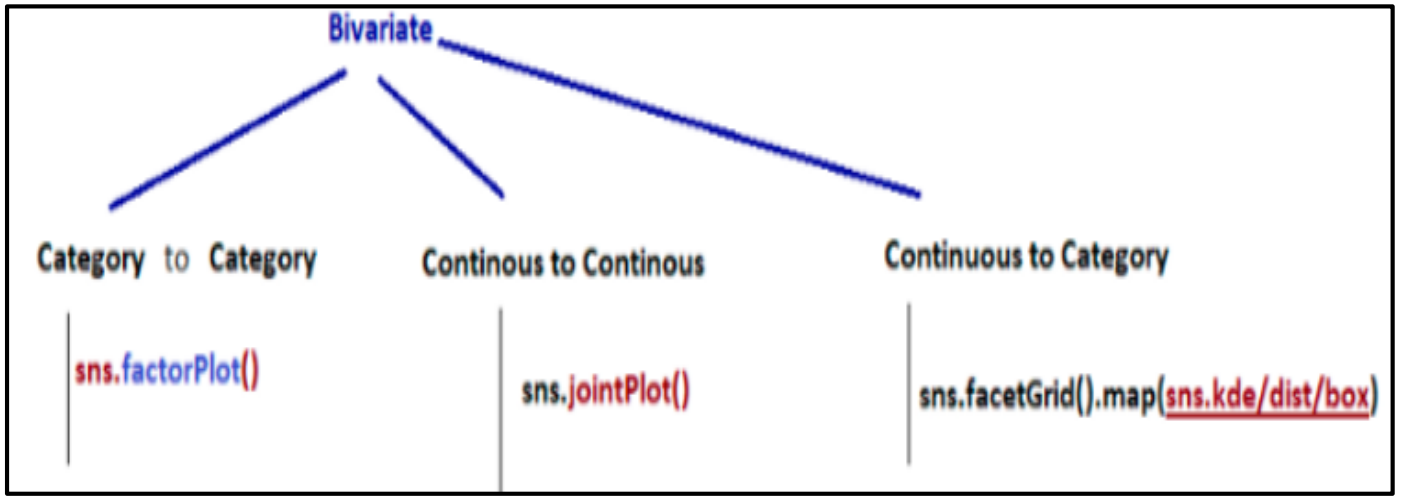
4 types of Data Analysis

- Discover Patterns
- Frame Hypothesis
- Spot Anomalies
- Check Assumptions

2 types of Data Analysis

- Univariate Analysis
- Bivariate Analysis





Stuff done during EDA

- Trends
- Distribution
- Mean
- Median
- Outlier
- Correlation
- Spread measurement (SD)
- Hypothesis testing
- Visual Exploration

Covariance

Covariance measures how two variables move together.

It measures whether the two move in the same direction (a positive covariance) or in opposite directions (a negative covariance).

When one has a high return, the other tends to have a high return as well. If the result was negative, then the two stocks would tend to have opposite returns;

when one had a positive return, the other would have a negative return.

Covariance calculations can give an investor insight into how two stocks might move together in the future.

Covariance

Looking at historical prices, we can determine if the prices tend to move with each other or opposite each other.

This allows you to predict the potential price movement of a two-stock portfolio.

In the stock market, a strong emphasis is placed on reducing the risk amount taken on for the same amount of return.

When constructing a portfolio, an analyst will select stocks that will work well together. This usually means that these stocks do not move in the same direction.

The covariance can also be used to find the standard deviation of a multi-stock portfolio.

The standard deviation is the accepted calculation for risk, and this is extremely important when selecting stocks.

Covariance

$$\text{Covariance} = \frac{\sum (\text{Return}_{ABC} - \text{Average}_{ABC}) * (\text{Return}_{XYZ} - \text{Average}_{XYZ})}{(\text{Sample Size}) - 1}$$

Day	ABC Returns (%)	XYZ Returns (%)
1	1.1	3
2	1.7	4.2
3	2.1	4.9
4	1.4	4.1
5	0.2	2.5

Table 1: Daily returns for two stocks using the closing prices

For ABC it would be (1.1 + 1.7 + 2.1 + 1.4 + 0.2) / 5 = 1.30

For XYZ it would be (3 + 4.2 + 4.9 + 4.1 + 2.5) / 5 = 3.74

Using our example on ABC and XYZ above, the covariance is calculated as:
= [(1.1 - 1.30) x (3 - 3.74)] + [(1.7 - 1.30) x (4.2 - 3.74)] + [(2.1 - 1.30) x (4.9 - 3.74)] + …
= [0.148] + [0.184] + [0.928] + [0.036] + [1.364]
= 2.66 / (5 - 1)
= 0.665

You can see that the covariance between the two stock returns is 0.665. Because this number is positive, it means the stocks move in the same direction. When ABC had a high return, XYZ also had a high return.

Correlation

Finding that two stocks have a high or low covariance might not be a useful metric on its own.

Covariance can tell how the stocks move together, but to determine the strength of the relationship, we need to look at the correlation.

The correlation should therefore be used in conjunction with the covariance, and is represented by this equation:

$$Correlation = \rho = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

where cov (X,Y) = covariance between X and Y
σ_X = standard deviation of X
σ_Y = standard deviation of Y

Correlation

The equation above reveals that the correlation between two variables is simply the covariance between both variables divided by the product of the standard deviation of the variables X and Y.

While both measures reveal whether two variables are positively or inversely related, the correlation provides additional information by telling you the degree to which both variables move together.

The correlation will always have a measurement value between -1 and 1, and adds a strength value on how the stocks move together.

Correlation

If the correlation is 1, they move perfectly together, and if the correlation is -1, the stocks move perfectly in opposite directions. If the correlation is 0, then the two stocks move in random directions from each other.

The correlation coefficient will vary from -1 to +1. A -1 indicates perfect negative correlation, and +1 indicates perfect positive correlation

Covariance just tells you that two variables change the same way, while correlation reveals how a change in one variable effects a change in the other.

Correlation

The covarince is scaled by the product of the two standard devations of the variables. This measure is called the Pearson correlation which holds true only when the relationship between two variables is linear in nature.

When the relationship is non-linear in nature Spearman correlation or rank correlation is used in order to account for the deviation from linearity.

Correlation

Pearson: The correlation number would always be in the range of -1 to +1.
A value of 1 means that the variables always move in the same direction and a value of -1 means the two always move in the opposite direction.
In the case where the variables are independent the covariance is zero which means the correlation is also zero.

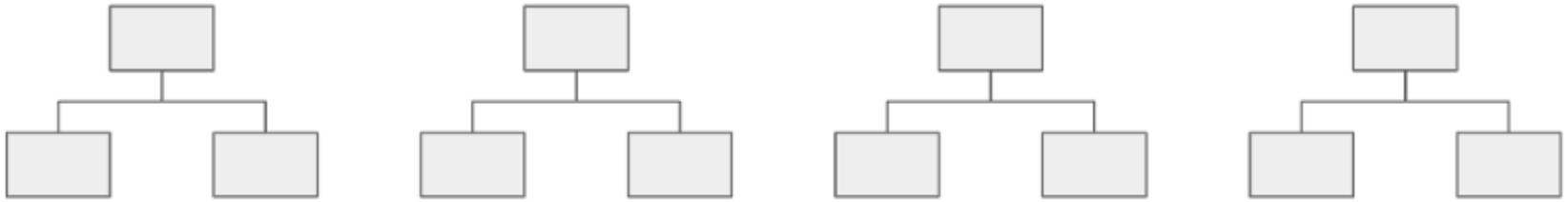
In other words the two variables do not exhibit any movement relative to each other.
Any number in between indicates that the one number moves less positively or negatively in relation to changes in another number.

Random Forest

Random Forest is an ensemble technique, meaning that it combines several models into one to improve its predictive power.

Specifically, it builds 1000s of smaller decision trees using bootstrapped datasets and random subsets of variables (also known as bagging).

With 1000s of smaller decision trees, random forests use a ‘majority wins’ model to determine the value of the target variable.



Random Forest

A collective of decision trees is called a Random Forest.

To classify a new object based on its attributes, each tree is classified, and the tree “votes” for that class.

The forest chooses the classification having the most votes (over all the trees in the forest).

Each tree is planted & grown as follows:

- If the number of cases in the training set is N, then a sample of N cases is taken at random.
  - This sample will be the training set for growing the tree.
- If there are M input variables, a number m<<M is specified such that at each node, m variables are selected at random out of the M, and the best split on this m is used to split the node. The value of m is held constant during this process.
  - The value of m is held constant during this process.
- Each tree is grown to the most substantial extent possible. There is no pruning.

Random Forest

Classifier

Classifier is a type of machine learning algorithm used to assign a class label to a data input.

An example is an image recognition classifier to label an image (e.g., "car," "truck," or "person").

Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems

Random forest builds decision trees on different samples and takes their majority vote for classification and average in case of regression

Random Forest Algorithm can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification

Random Forest Algorithm performs better results for classification problems

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems

Random Forest

The Random forest algorithm doesn't rely on single decision tree output, it analyses output from different decision trees and finally gives output on the basis of majority voting.

Random Forest is an ensemble technique, meaning that it combines several models into one to improve its predictive power.

Specifically, it builds 1000s of smaller decision trees using bootstrapped datasets and random subsets of variables (also known as bagging).

With 1000s of smaller decision trees, random forests use a 'majority wins' model to determine the value of the target variable.

Random Forest

A collective of decision trees is called a Random Forest.

To classify a new object based on its attributes, each tree is classified, and the tree “votes” for that class.

The forest chooses the classification having the most votes (over all the trees in the forest).

Each tree is planted & grown as follows:

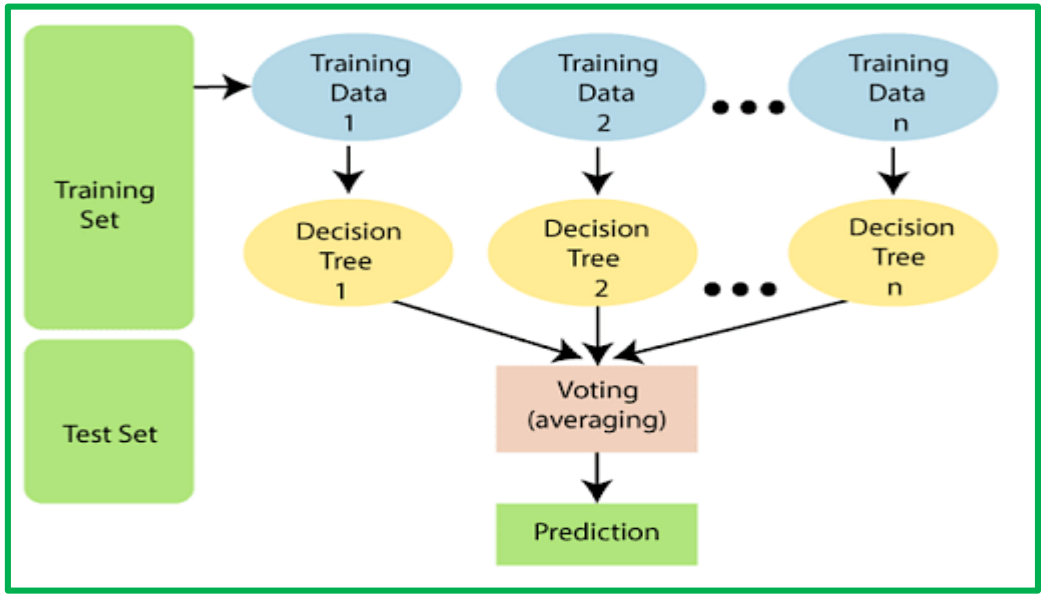
- If the number of cases in the training set is N, then a sample of N cases is taken at random.
  - This sample will be the training set for growing the tree.
- If there are M input variables, a number m<<M is specified such that at each node, m variables are selected at random out of the M, and the best split on this m is used to split the node. The value of m is held constant during this process.
  - The value of m is held constant during this process.
- Each tree is grown to the most substantial extent possible. There is no pruning.

Working of Random Forest Algorithm

Step 1: Select random samples from a given data or training set.

Step 2: This algorithm will construct a decision tree for every training data.

Step 3: Voting will take place by averaging the decision tree.



Step 4: Finally, select the most voted prediction result as the final prediction result.

Random Forest Algorithm can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification

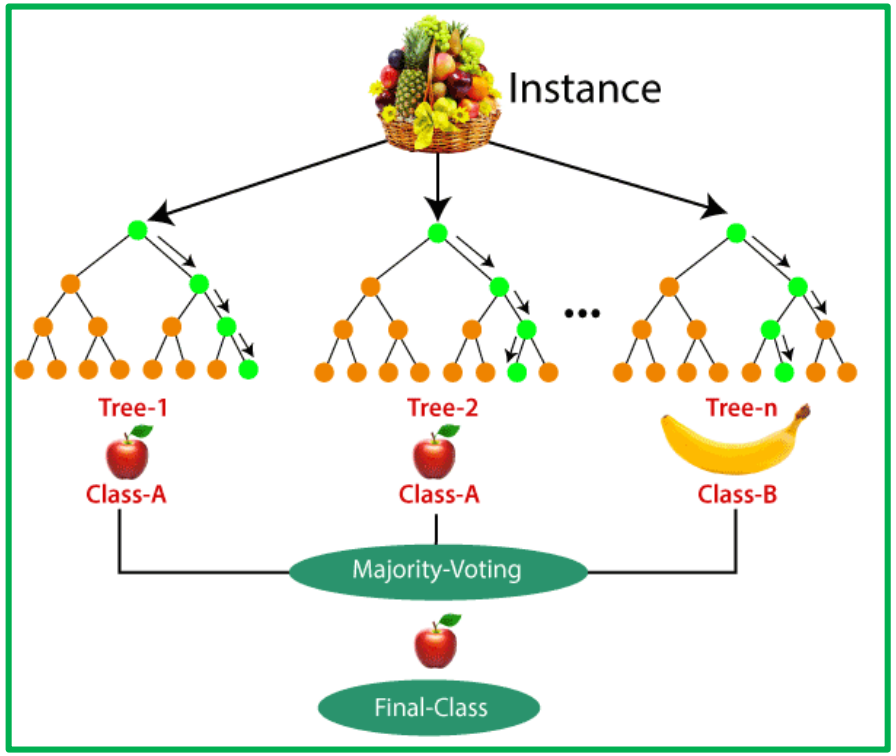
Working of Random Forest Algorithm

Dataset that contains multiple fruit images

Dataset is given to the Random forest classifier

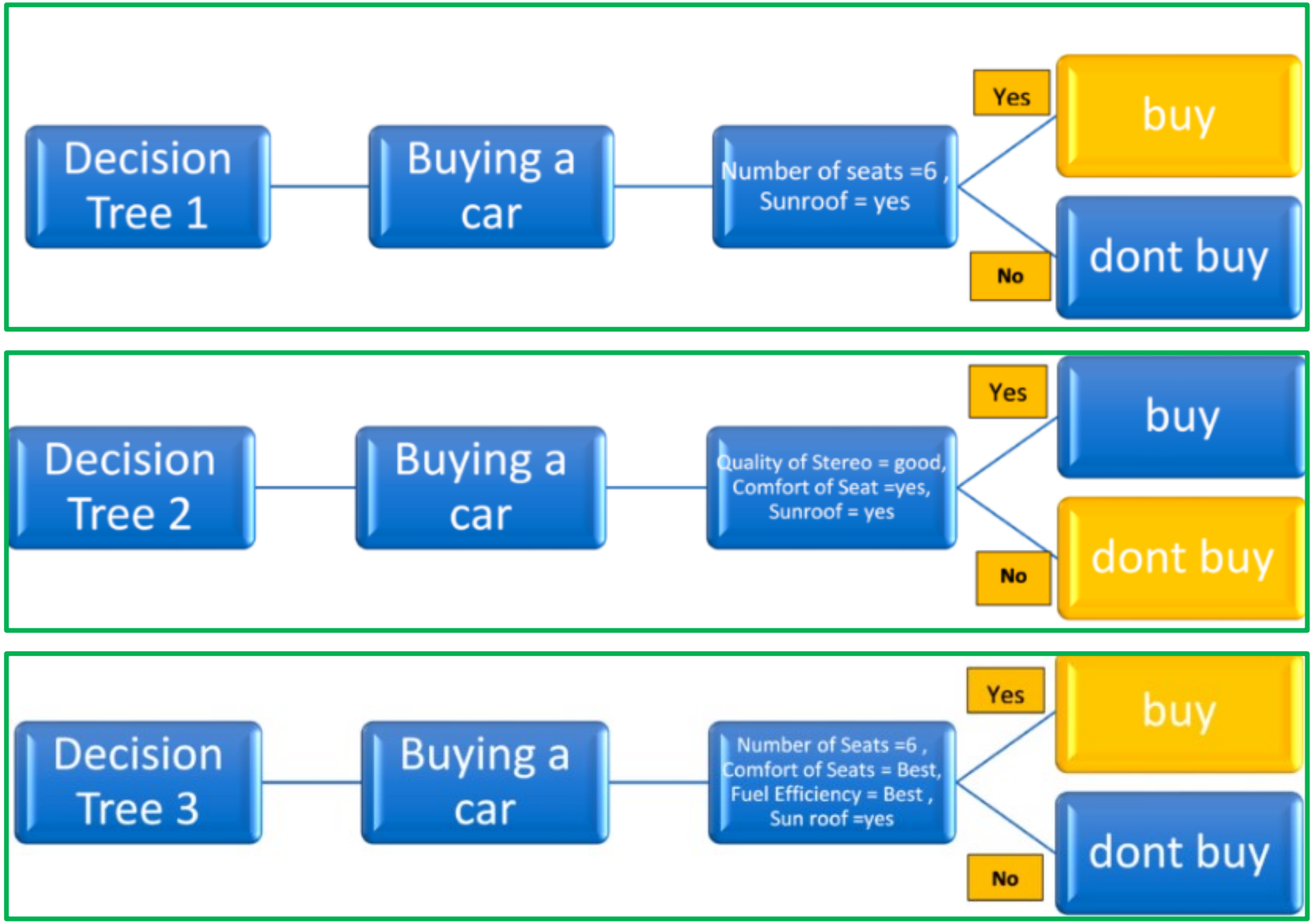
Dataset is divided into subsets and given to each decision tree

During the training phase, each decision tree produces a prediction result , and



When a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision

Working of Random Forest Algorithm



Random Forest Features

- Accuracy of Random forest is generally very high
- Its efficiency is particularly Notable in Large Data sets
- Provides an estimate of important variables in classification
- Forests Generated can be saved and reused
- Unlike other models It does nt overfit with more features

Random Forest

Ensemble learning techniques

Bagging

Bagging, an ensemble technique is known by the name Bootstrap Aggregation

Bagging chooses a randomized subset from the data set

Then Original Data is randomly selected and is given parallelly to different models(weak learners or base learners) but with replacement

Gradient Boosting

Gradient boosting is a machine learning technique used in regression and classification tasks, among others

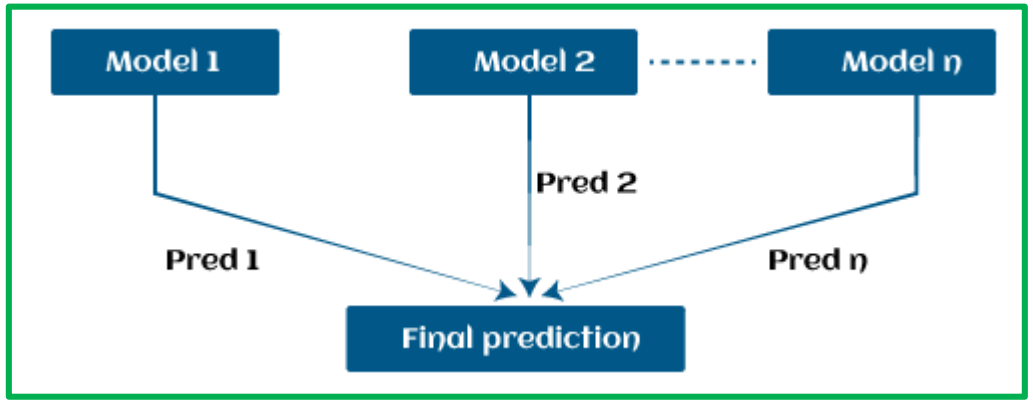
Gradient boosting gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees

Gradient boosting is a type of machine learning boosting.

The key idea is to set the target outcomes for this next model in order to minimize the error.

A Gradient Boosting Machine or GBM combines the predictions from multiple decision trees to generate the final predictions

Gradient boosting relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error



Gradient Boosting

Gradient Boosting Machine (GBM) is Forward learning ensemble methods in machine learning

Gradient Boosting Machine (GBM) is a powerful technique for building predictive models for regression and classification tasks.

Gradient Boosting Machine (GBM) helps us to get a predictive model in form of an ensemble of weak prediction models such as decision trees

Whenever a decision tree performs as a weak learner then the resulting algorithm is called gradient-boosted trees

Gradient Boosting Machine (GBM) enables us to combine the predictions from various learner models and build a final predictive model having the correct prediction.

Gradient Boosting

Weak Learner:

Weak learners are the base learner models that learn from past errors and help in building a strong predictive model design for boosting algorithms in machine learning

Decision trees work as a weak learners in boosting algorithms

Boosting is defined as the framework that continuously works to improve the output from base models.

Many gradient boosting applications allow you to "plugin" various classes of weak learners at your disposal.

Additive Model

Additive model is defined as adding trees to the model.

only a single tree must be added so that existing trees in the model are not changed

Machines, Model Stacking

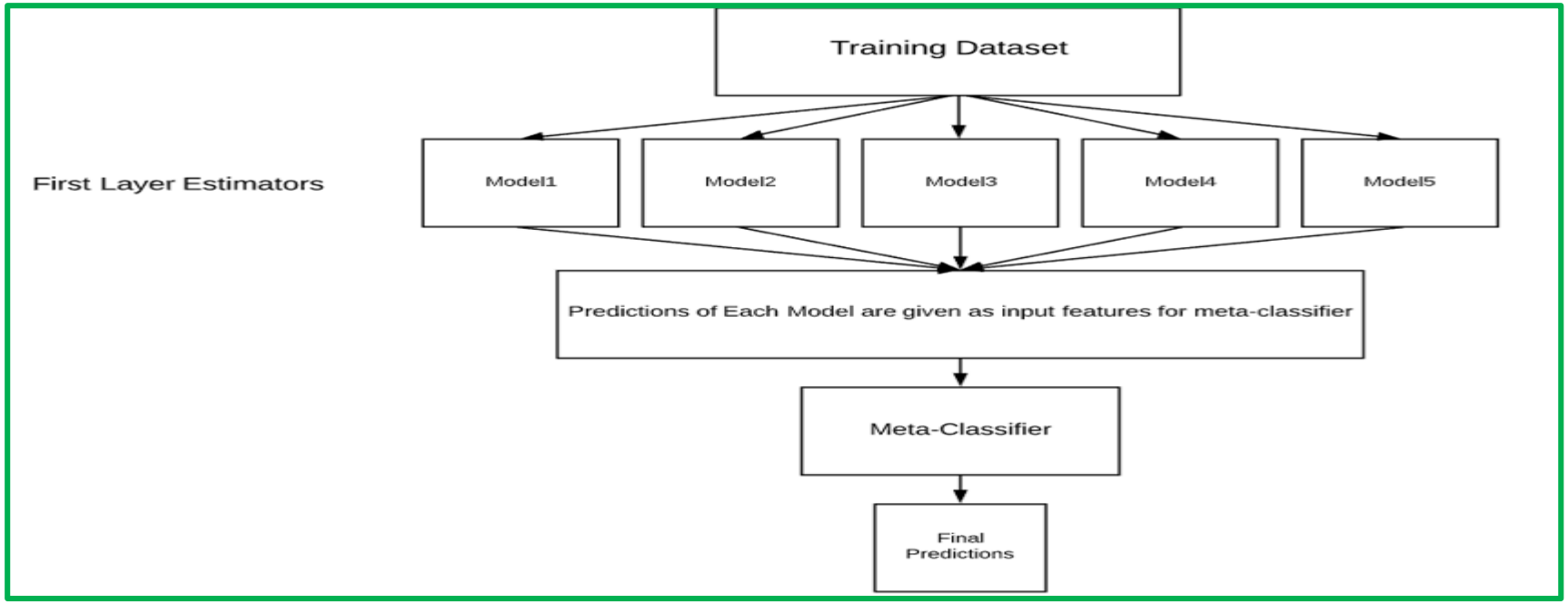
Model Stacking is a way to improve model predictions by combining the outputs of multiple models and running them through another machine learning model called a meta-learner

They can improve the existing accuracy that is shown by individual models.

Stacking is a way of ensembling classification or regression models it consists of two-layer estimators

First layer consists of all the baseline models that are used to predict the outputs on the test datasets

Second layer consists of Meta-Classifier or Regressor which takes all the predictions of baseline models as an input and generate new predictions



Machines, Model Stacking

Most of the Machine-Learning and Data science competitions are won by using Stacked models

Stacked models can improve the existing accuracy that is shown by individual models

We can get most of the Stacked models by choosing diverse algorithms in the first layer of architecture as different algorithms capture different trends in training data by combining both of the models can give better and accurate results

Installation of libraries on the system

```
pip install mlxtend
pip install pandas
pip install -U scikit-learn
```

CAT Boost

CatBoost is an algorithm for gradient boosting on decision trees

Domain expertise

Programming skills, and

Knowledge of mathematics and statistics to extract meaningful insights from data

CatBoost is developed by Yandex researchers and engineers, and is used for

Search

Recommendation systems,

Personal assistant

Self-driving cars

Weather prediction and many other tasks

CatBoost is the best option to deal with categorical features

CatBoost algorithm is based on Gradient Descent and is a powerful technique for supervised machine learning tasks

XG Boost

XGBoost (eXtreme Gradient Boosting) is a popular and efficient open-source implementation of the gradient boosted trees algorithm

Gradient boosting is a supervised learning algorithm that attempts to accurately predict a target variable by combining an ensemble of estimates from a set of simpler and weaker models

XGBoost has some more generalization capabilities than other boosting techniques

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library

XGBoost provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems

XGBoost is used for supervised learning problems, where we use the training data (with multiple features) to predict a target variable

Adaptive Boosting(AdaBoost)

Adaptive Boosting(AdaBoost) was the first successful boosting algorithm developed for binary classification. Also, it is the best starting point for understanding boosting algorithms

Adaptive Boosting(AdaBoost) is the best starting point for understanding boosting algorithms

Adaptive Boosting(AdaBoost) is adaptive in the sense that subsequent classifiers built are tweaked in favour of those instances misclassified by previous classifiers.

Adaptive Boosting(AdaBoost) is sensitive to noisy data and outliers.

Adaptive Boosting(AdaBoost) uses multiple iterations to generate a single composite strong learner.

Adaptive Boosting(AdaBoost) creates a strong learner by iteratively adding weak learners

During each phase of training, a new weak learner is added to the ensemble, and a weighting vector is adjusted to focus on examples that were misclassified in previous rounds

The result is a classifier that has higher accuracy than the weak learner classifiers.

Random Forest

Advantages of Using Random Forest ML Algorithm

Overfitting is less of an issue with Random Forests, unlike decision tree machine learning algorithms.

There is no need of pruning the random forest.

Random Forest ML Algorithms are fast but not in all cases.

A random forest algorithm, when run on an 800 MHz machine with a dataset of 100 variables and 50,000 cases produced 100 decision trees in 11 minutes.

Random Forest is one of the most effective and versatile machine learning algorithm for wide variety of classification and regression tasks, as they are more robust to noise.

It is difficult to build a bad random forest.

In the implementation of Random Forest Machine Learning algorithms, it is easy to determine which parameters to use because they are not sensitive to the parameters that are used to run the algorithm. One can easily build a decent model without much tuning.

In the implementation of Random Forest Machine Learning algorithms, it is easy to determine which parameters to use because they are not sensitive to the parameters that are used to run the algorithm.

One can easily build a decent model without much tuning.

Random Forest

Advantages of Using Random Forest ML Algorithm

- Random Forest machine learning algorithms can be grown in parallel.
- Random Forest ML Algorithms runs efficiently on large databases.
- Random forest algorithm has higher classification accuracy
- Random Forest machine learning algorithms might be easy to use but analyzing them theoretically, is difficult.
- Large number of decision trees in the random forest can slow down the algorithm in making real-time predictions.
- If the data consists of categorical variables with different number of levels, then the algorithm gets biased in favor of those attributes that have more levels. In such situations, variable importance scores do not seem to be reliable.
- When using Random Forest algorithm for regression tasks, it does not predict beyond the range of the response values in the training data.

Random Forest

Disadvantages of Using Random Forest ML Algorithm

Random Forest machine learning algorithms might be easy to use but analyzing them theoretically, is difficult.

Large number of decision trees in the random forest can slow down the algorithm in making real-time predictions.

If the data consists of categorical variables with different number of levels, then the algorithm gets biased in favor of those attributes that have more levels. In such situations, variable importance scores do not seem to be reliable.

When using Random Forest algorithm for regression tasks, it does not predict beyond the range of the response values in the training data.

Random Forest

Applications of Random Forest Machine Learning Algorithm

Random Forest algorithms are used by banks to predict if a loan applicant is a likely high risk.

They are used in the automobile industry to predict the failure or breakdown of a mechanical part.

These algorithms are used in the healthcare industry to predict if a patient is likely to develop a chronic disease or not.

They can also be used for regression tasks like predicting the average number of social media shares and performance scores.

Recently, the algorithm has also made way into predicting patterns in speech recognition software and classifying images and texts.

Thank you for Listening

Any Questions

Dr DV Ramana
Data Stratagist
Wissen Infotech

Mail Address: pythonpmg@gmail.com

To contact: +91 9959423084