Bias represents the historical average error. Basically, will your forecasts be, on average, too high (i.e., you overshot the demand) or too low (i.e., you undershot the demand)? This will give you the overall direction of the error.

Precision measures how much spread you will have between the forecast and the actual value. The precision of a forecast gives an idea of the magnitude of the errors but not their overall direction.

e = y - y`

MAPE : The Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n}\sum \frac{|e_t|}{d_t}$$

MAPE is the sum of the individual absolute errors divided by the demand (each period separately). It is the average of the percentage errors.

MAE : Mean Absolute Error

$$MAE = \frac{1}{n}\sum |e_t|$$

One of the first issues of this KPI is that it is **not scaled to the average demand**. If one tells you that MAE is 10 for a particular item, you cannot know if this is good or bad. If your average demand is 1000, it is, of course, astonishing. Still, if the average demand is 1, this is a very poor accuracy. To solve this, it is common to divide MAE by the average demand to get a %:

$$MAE\% = \frac{\frac{1}{n}\sum |e_t|}{\frac{1}{n}\sum d_t} = \frac{\sum |e_t|}{\sum d_t}$$

RMSE : Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum e_t^2}$$

Just as for MAE, RMSE is not scaled to the demand. We can then define RMSE% as such

$$RMSE\% = \frac{\sqrt{\frac{1}{n}\sum e_t^2}}{\frac{\sum d}{n}}$$

Actually, many algorithms (especially for machine learning) are based on the **Mean Squared Error (MSE)**, which is directly related to RMSE.

$$MSE = \frac{1}{n}\sum e_t^2$$

Many algorithms use **MSE as it is faster to compute and easier to manipulate than RMSE**. But it is not scaled to the original error (as the error is squared)

Decision Tree : Decision tree analysis involves making a tree-shaped diagram to chart out a course of action or a statistical probability analysis. It is used to break down complex problems or branches. Each branch of the decision tree could be a possible outcome.

Types of decision trees are based on the type of target variable we have. It can be of two types:

1. **Categorical Variable Decision Tree:** Decision Tree which has a categorical target variable then it is called a **Categorical variable decision tree.**
2. **Continuous Variable Decision Tree:** Decision Tree has a continuous target variable then it is called **Continuous Variable Decision Tree.**

Assumptions while creating Decision Tree

- In the beginning, the whole training set is considered as the **root.**
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are **distributed recursively** on the basis of attribute values.
- Order to place attributes as the root or internal node of the tree is done by using some statistical approach.

Attribute Selection Measures

Entropy : Entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information. Flipping a coin is an example of an action that provides information that is random.

Information gain : Information gain or IG is a statistical property that measures how well a given attribute separates the training examples according to their target classification. Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy.

$$Information\ Gain\ =\ Entropy(before) - \sum_{j=1}^{K} Entropy(j,\ after)$$

Gini index : You can understand the Gini index as a cost function used to evaluate splits in the dataset. It is calculated by subtracting the sum of the squared probabilities of each class from one. It favors larger partitions and is easy to implement whereas information gain favors smaller partitions with distinct values. Gini Index works with the categorical target variable “Success” or “Failure”.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Gain Ratio : Information gain is biased towards choosing attributes with a large number of values as root nodes. It means it prefers the attribute with a large number of distinct values.

$$Gain\ Ratio = \frac{Information\ Gain}{SplitInfo} = \frac{Entropy\ (before) - \sum_{j=1}^{K} Entropy(j,\ after)}{\sum_{j=1}^{K} w_j \log_2 w_j}$$

Reduction in Variance : Reduction in variance is an algorithm used for continuous target variables (regression problems). This algorithm uses the standard formula of variance to choose the best split. The split with lower variance is selected as the criteria to split the population:

$$\text{Variance} = \frac{\Sigma(X - \overline{X})^2}{n}$$

Chi-Square : The acronym CHAID stands for Chi-squared Automatic Interaction Detector. It is one of the oldest tree classification methods. It finds out the statistical significance between the differences between sub-nodes and parent node. We measure it by the sum of squares of standardized differences between observed and expected frequencies of the target variable.
It works with the categorical target variable “Success” or “Failure”. It can perform two or more splits. Higher the value of Chi-Square higher the statistical significance of differences between sub-node and Parent node.

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Where:

χ^2 = Chi Square obtained
$\sum$ = the sum of
O = observed score
E = expected score

How to avoid/counter Overfitting in Decision Trees?

1. Pruning Decision Trees.
2. Random Forest

Pruning Decision Trees

The splitting process results in fully grown trees until the stopping criteria are reached. But, the fully grown tree is likely to overfit the data, leading to poor accuracy on unseen data.

Random Forest

Random Forest is an example of ensemble learning, in which we combine multiple machine learning algorithms to obtain better predictive performance.

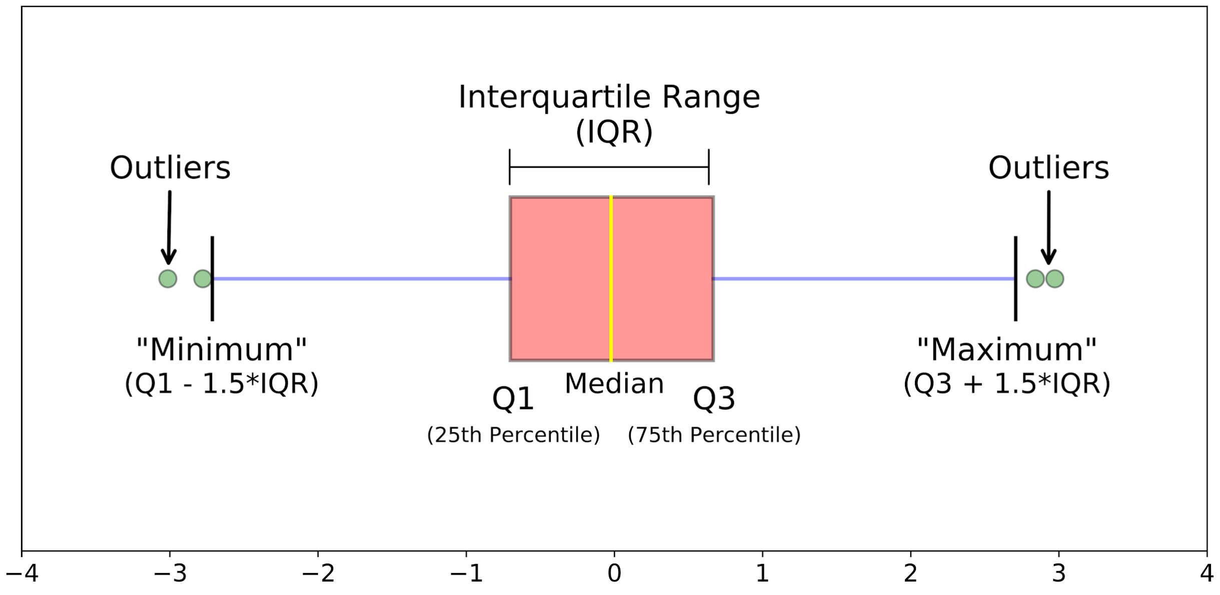
Why the name “Random”?

Two key concepts that give it the name random:

1. A random sampling of training data set when building trees.
2. Random subsets of features considered when splitting nodes.

What is a Boxplot?

For some distributions/datasets, you will find that you need more information than the measures of central tendency (median, mean, and mode). You need to have information on the variability or dispersion of the data. A boxplot is a graph that gives you a good indication of how the values in the data are spread out. Although boxplots may seem primitive in comparison to a histogram or density plot, they have the advantage of taking up less space, which is useful when comparing distributions between many groups or datasets.



Boxplots are a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”).

median (Q2/50th Percentile): the middle value of the dataset.

first quartile (Q1/25th Percentile): the middle number between the smallest number (not the “minimum”) and the median of the dataset.

third quartile (Q3/75th Percentile): the middle value between the median and the highest value (not the “maximum”) of the dataset.

interquartile range (IQR): 25th to the 75th percentile.

whiskers (shown in blue)

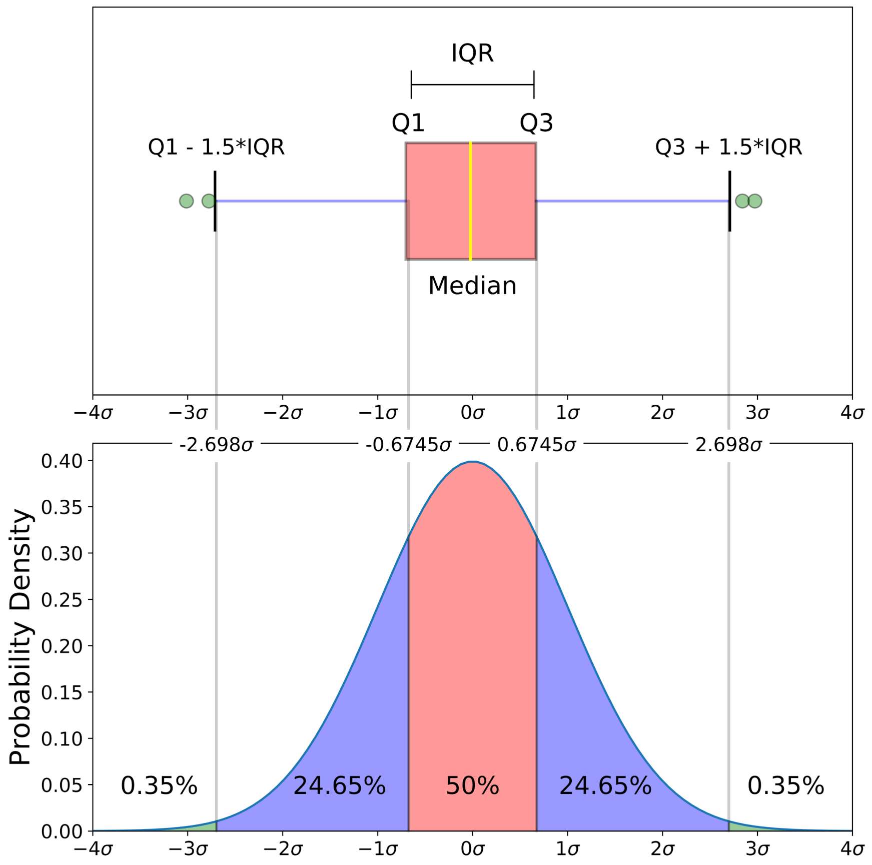
outliers (shown as green circles)

“maximum”: Q3 + 1.5*IQR

“minimum”: Q1 -1.5*IQR

What defines an outlier, “minimum”, or“maximum” may not be clear yet. The next section will try to clear that up for you.

Boxplot on a Normal Distribution



Probability Density Function

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

For more info : https://youtu.be/BE8CVGJuftI

NLTK : The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.

Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. **Stemming** is important in natural language understanding (NLU) and **natural language processing** (**NLP**).

TfidfVectorizer : Convert a collection of raw documents to a matrix of TF-IDF features.

LinearSVC : Linear Support Vector Classification. Similar to SVC with parameter kernel='linear', but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall AND there is an uneven class distribution (large number of Actual Negatives).

- *Vaibhav Saraf*