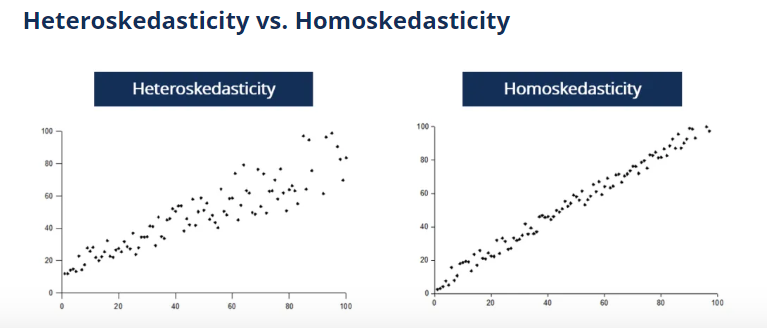
**Heteroskedasticity** refers to situations where the variance of the residuals is unequal over a range of measured values. When running a regression analysis, heteroskedasticity results in an unequal scatter of the residuals (also known as the error term).

When observing a plot of the residuals, a fan or cone shape indicates the presence of heteroskedasticity. In statistics, heteroskedasticity is seen as a problem because regressions involving ordinary least squares (OLS) assume that the residuals are drawn from a population with constant variance.

**Homoskedasticity** refers to situations where the residuals are equal across all the independent variables.

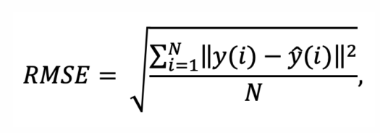
If a model is homoskedastic, we can assume that the residuals are drawn from a population with constant variance. It would satisfy one of the assumptions of the OLS regression and ensure that the model is more accurate.



Error Mertics:

1. RMSE (Root Mean Squared Error): It measures the average difference between values predicted by a model and the actual values. It provides an estimation of how well the model is able to predict the target value (accuracy).

RMSE can be expressed as :



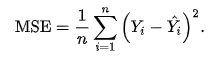
where N is the number of data points, y(i) is the ith measurement, and y ̂(i) is its corresponding prediction.

Lower the value of RMSE, better the model fit.

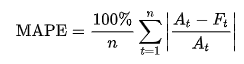
1. MAE (Mean Absolute Error): It is a measure of errors between paired observations expressing the same phenomenon.



1. MSE (Mean Squared Error): It measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss. The fact that MSE is almost always strictly positive (and not zero) is because of randomness or because the estimator does not account for information that could produce a more accurate estimate. In machine learning, specifically empirical risk minimization, MSE may refer to the empirical risk (the average loss on an observed data set), as an estimate of the true MSE (the true risk: the average loss on the actual population distribution). It is also known as MSD (Mean Squared Deviation)



1. MAPE (Mean Absolute Percentage Error):  It is a measure of prediction accuracy of a forecasting method in statistics. It usually expresses the accuracy as a ratio defined by the formula:



It is also known as MAPD (Mean Absolute Percentage Deviation). It is used as a loss function for regression problems and in model evaluation, because of its very intuitive interpretation in terms of relative error.

1. EVS (Explained Variance Score): It The explained variance score explains the dispersion of errors of a given dataset, and the formula is written as follows:



1. SK Learn Vs OLS stats model:

The main difference between these two is how they handle constants. Scikit-learn allows the user to or not to add a constant through a parameter, while Stats models’ OLS class has a function that adds a constant to a given array.

Scikit-learn's algorithm is a simple, easy learn that only require our data to be organized in the right way before we can run whatever classification, regression, or clustering algorithm we need. While Stats Models don't have a variety of options, it only offers statistics and econometric tools that are used in statistics software like Stata and R. It has a similar syntax as that of R so, for those who are transitioning to Python, Stats Models is a good choice.

Stats models and Scikit-learn are designed to meet different needs. Stats models has a stronger emphasis on statistical inference and statistical hypothesis testing whereas Scikit-learn is well-suited for projects where the prediction of unobserved values is key.

In general,

* scikit-learn is designed for prediction, while statsmodels is more suited for explanatory analysis.
* scikit-learn regularizes by default while statsmodels does not
* A model designed for prediction is best fit using scikit-learn, while statsmodels is best employed for explanatory models.

1. Decision tree regressor vs Random Forest Regressor:

The difference between the random forest algorithm and decision tree is critical and based on the problem statement. Decision trees are implemented when it involves a mixture of feature data types and easy interpretation. The random forest algorithm model handles multiple trees so that the performance is not affected. It does not require scaling or normalization

A classification tree splits the dataset based on the homogeneity of data. Say, for instance, there are two variables; salary and location; which determine whether or not a candidate will accept a job offer.

If the training data shows that 95% of people accept the job offer based on salary, the data gets split there and salary becomes a top node in the tree. This split makes the data “95% pure”. Measures of impurity like entropy are used to quantify the homogeneity of the data when it comes to classification trees.

In a regression tree, a regression model is fit to the target variable using each of the independent variables. The data is then split at several points for each independent variable.

At those points, the error between the predicted values and actual values is squared to get “A Sum of Squared Errors” (SSE). The point that has the lowest SSE is chosen as the split point. This process is continued recursively.

The random forest regression algorithm takes advantage of the ‘wisdom of the crowds’. It takes multiple (but different) regression decision trees and makes them ‘vote’.

**Advantages of Decision Trees**

1. Decision trees are easy to interpret.

2. To build a decision tree requires some data preparation from the user but normalization of data is not required.

**Disadvantages of Decision Trees**

1. Decision trees are likely to over-fit noisy data. The probability of overfitting on noise increases as a tree gets deeper.

**To Conclude:**

* A decision tree can be used for either regression or classification.
* Decision trees are easy to understand, visualize and interpret.
* The flowchart-like structure helps us in decision-making.

1. ANN Regressor

The purpose of using Artificial Neural Networks for Regression over Linear Regression is that the linear regression can only learn the linear relationship between the features and target and therefore cannot learn the complex non-linear relationship. Artificial Neural Networks have the ability to learn the complex relationship between the features and target due to the presence of activation function in each layer.

1. Bagging Vs Ada Boost Regressor

Bagging: It is a homogeneous weak learners’ model that learns from each other independently in parallel and combines them for determining the model average.

Boosting: It is also a homogeneous weak learners’ model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.

**Similarities Between Bagging and Boosting**

Bagging and Boosting, both being the commonly used methods, have a universal similarity of being classified as ensemble methods. Here we will explain the similarities between them.

* Both are ensemble methods to get N learners from 1 learner.
* Both generate several training data sets by random sampling.
* Both make the final decision by averaging the N learners (or taking the majority of them i.e Majority Voting).
* Both are good at reducing variance and provide higher stability.

**Differences Between Bagging and Boosting**

| S.NO | Bagging | Boosting |
| --- | --- | --- |
| 1. | The simplest way of combining predictions that  belong to the same type. | A way of combining predictions that  belong to the different types. |
| 2. | Aim to decrease variance, not bias. | Aim to decrease bias, not variance. |
| 3. | Each model receives equal weight. | Models are weighted according to their performance. |
| 4. | Each model is built independently. | New models are influenced  by the performance of previously built models. |
| 5. | Different training data subsets are selected using row sampling with replacement and random sampling methods from the entire training dataset. | Every new subset contains the elements that were misclassified by previous models. |
| 6. | Bagging tries to solve the over-fitting problem. | Boosting tries to reduce bias. |
| 7. | If the classifier is unstable (high variance), then apply bagging. | If the classifier is stable and simple (high bias) the apply boosting. |
| 8. | In this base classifiers are trained parallelly. | In this base classifiers are trained sequentially. |
| 9 | Example: The Random forest model uses Bagging. | Example: The AdaBoost uses Boosting techniques |