**Classification Models**

**Course Leads: Ben Burkholder**

[**https://www.udacity.com/course/classification-models--ud978**](https://www.udacity.com/course/classification-models--ud978)

**Jocelyn (Yuan) Li**

**February 2017**

**About this Course**

The Classification Models course provides students with the foundational knowledge to use classification models to create business insights. You will learn:

* How classification modeling differs from modeling with numeric data
* To use binary classification models to make predictions of binary outcomes
* To use non-binary classification models to make predictions of non-binary outcomes

Throughout this course you’ll also learn the techniques to apply your knowledge in a data analytics program called **Alteryx**. At the end of the course, you’ll complete a project based on the principles in the course.

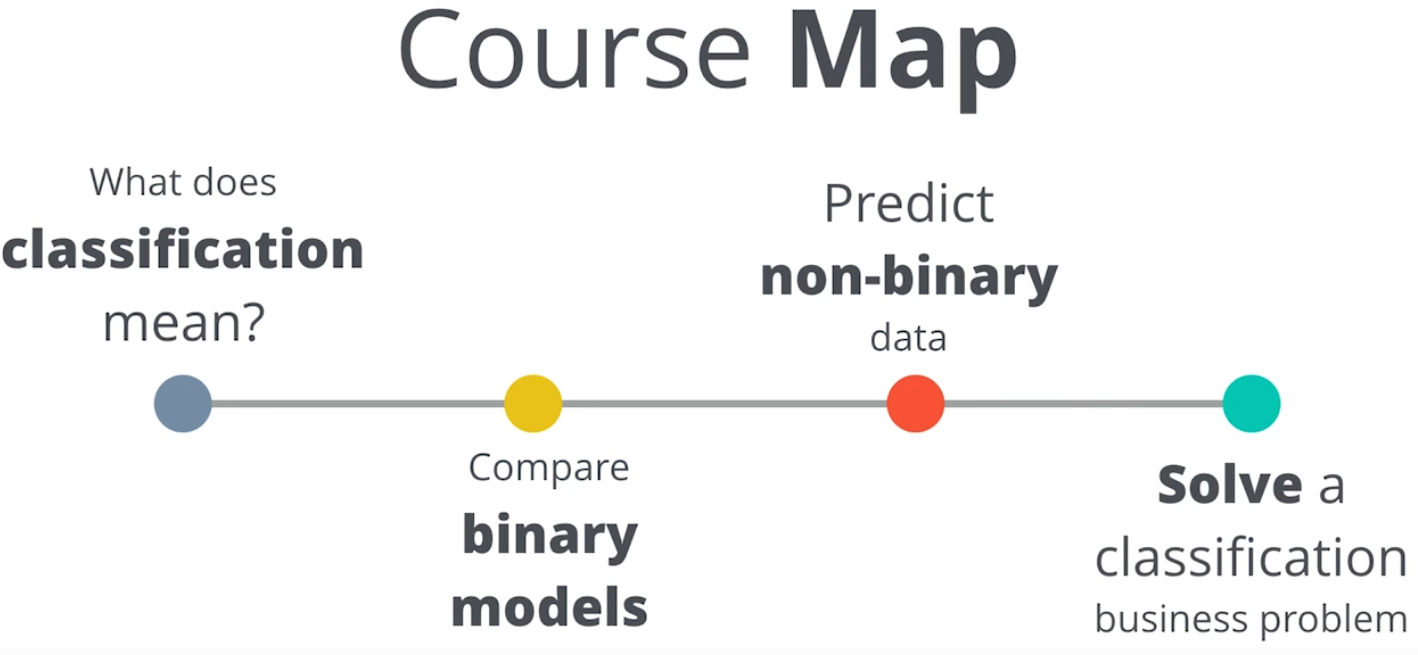
This course is part of the Business Analyst Nanodegree.

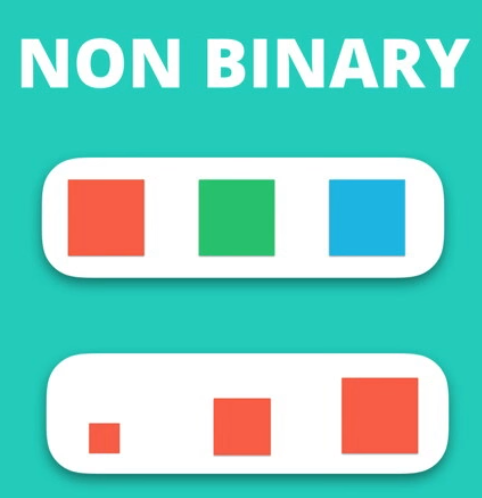
**Key words: logistic regression, decision trees, random forest model, boosted model**

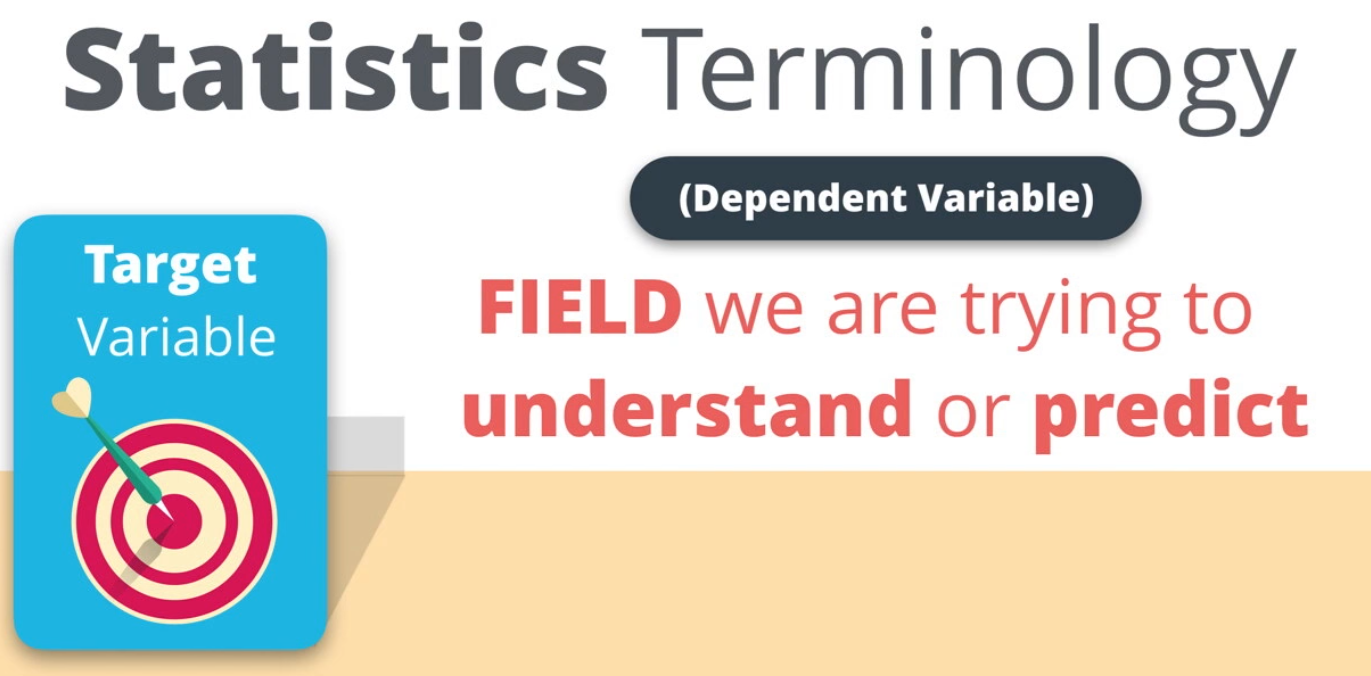
**Why Take This Course**

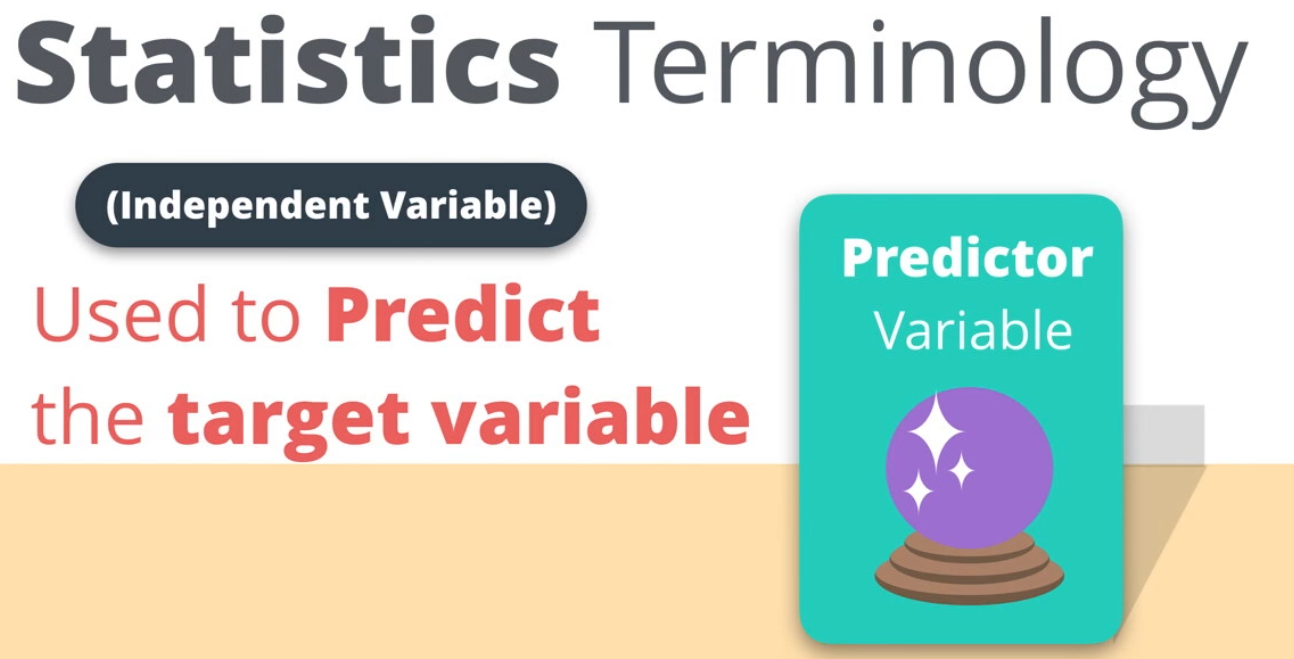
Predictive analytics has proved to be a powerful tool to help businesses analyze data and predict future outcomes and trends. In this course, you’ll learn how to use **classification predictive models to solve business problems such as predicting whether or not a customer will respond to a marketing campaign, the likelihood of default on a loan, or which product a customer will buy**. You'll learn this through improving your fluency in Alteryx, a data analytics tool that enables you prepare, blend, and analyze data quickly. This course is ideal for anyone who is interested in pursuing a career in business analysis, but lacks programming experience.

1. **Course Outline**

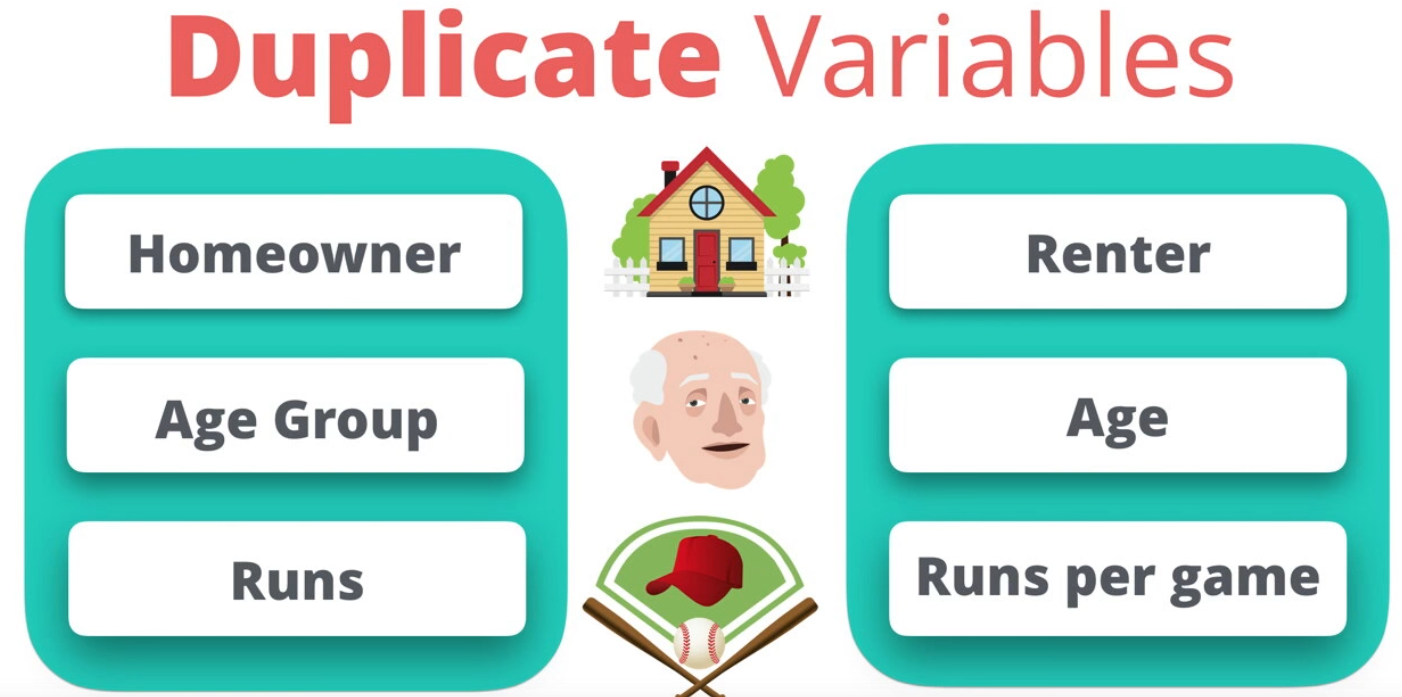






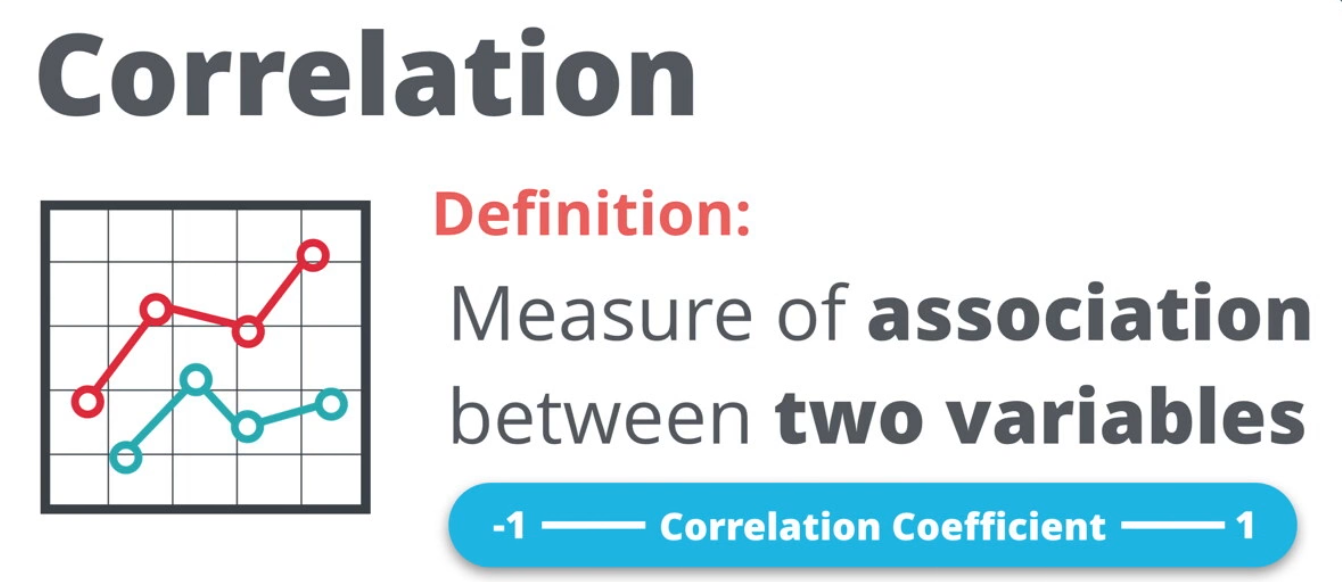


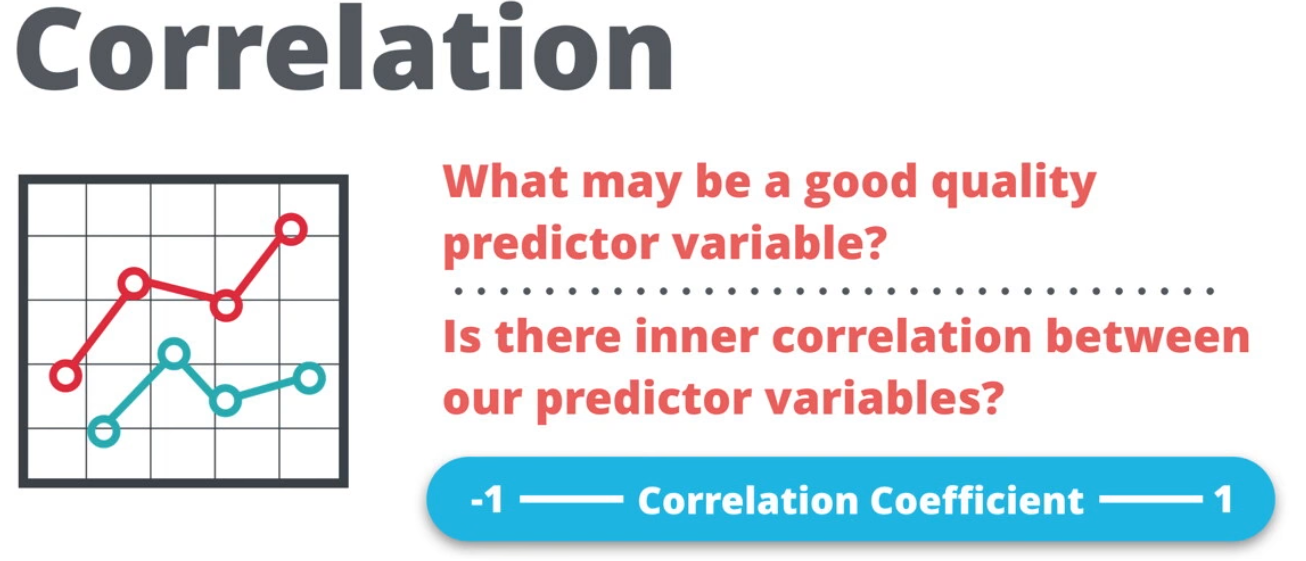
1. **Duplicated Variables**



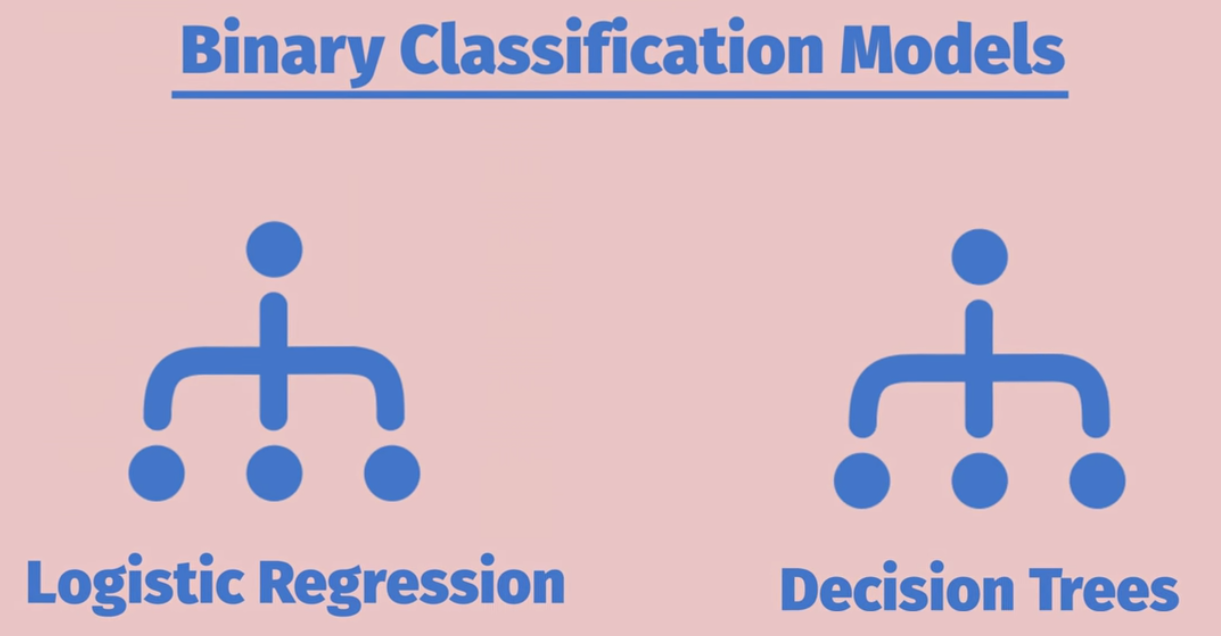
1. **Correlation Coefficients**

* [**Pearson Correlation Coefficient**](http://wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient)
* [**Spearman's Rank Correlation Coefficient**](http://en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficient)
* [**Hoeffiding's Independence Test**](http://en.wikipedia.org/wiki/Hoeffding's_independence_test)





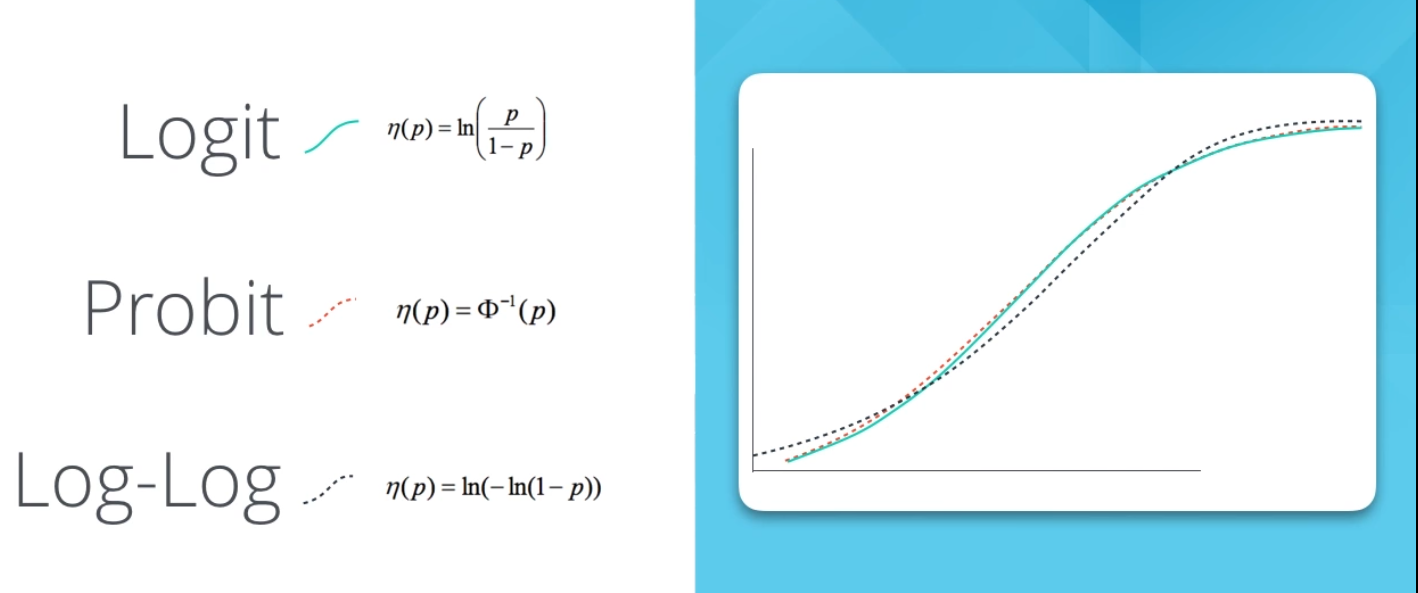
1. **Binary Classification**



1. **Logistic Regression**

## Introduction to Logistic Regression

Logistic regression is one of the most basic forms of regression modeling. It’s part of a family of “generalized linear models” or GLM for short. This basically means that **the formula is very similar to that of a linear regression.** However since the target variable is binary, instead of a continuous numeric variable, the target variable has to be modified to fit this GLM formula. See the video below for more on the structure of Logistic Regression.



1. **Logistic Regression – Stepwise**

**Stepwise regression** is a semi-automated process of building a model by successively adding or removing variables based solely on the t-statistics of their estimated coefficients.

Note that the **Stepwise Regression** tool is a tool to help you reduce and figure out which predictor variables have a good chance of being in the model, but **it is not a tool that can automatically find all of the appropriate predictor variables in one run.**

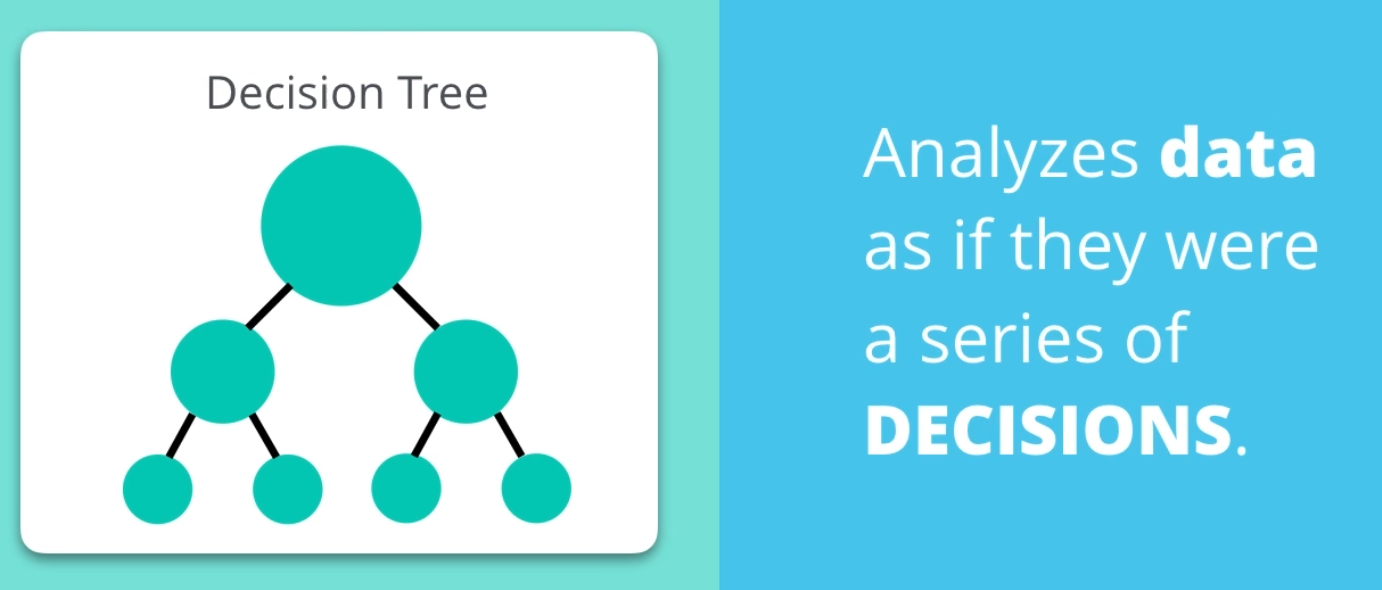
There is still a lot of work that needs to get done to explore the predictor variables that the Stepwise Regression tool gives you. The Stepwise Regression tool will speed up the process for you in choosing predictor variables.

#### AIC vs BIC

This information is from the [**Stepwise Regression Tool description on Alteryx's website**](http://downloads.alteryx.com/Alteryx/Help/Stepwise.htm): "A choice of two different adjusted fit measures are provided to the user, the Akaike information criterion (or AIC) and the Bayesian information criterion (or BIC). These two measures are similar to one another, but the BIC places a larger penalty on the number of variables included in the model, typically resulting in a final model with fewer variables than is the case when the AIC is used."

1. **Decision Tree Model**

A **decision tree**is a decision support tool that uses a **tree-like graph** or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm.



**cross-validation** involves [partitioning](https://en.wikipedia.org/wiki/Partition_of_a_set) a [sample](https://en.wikipedia.org/wiki/Statistical_sample) of [data](https://en.wikipedia.org/wiki/Data) into [complementary](https://en.wikipedia.org/wiki/Complement_(set_theory)) subsets, performing the analysis on one subset (called the **training set**), and validating the analysis on the other subset (called the validation set or **testing set**). To reduce [variability](https://en.wikipedia.org/wiki/Variance), multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

**Root Node Error:** A percentage of how many of the data points went to the incorrect terminal node (predicted incorrectly) when all of the data points are validated against themselves within the entire training set (the Estimation dataset).

**Pruning Table:** Lists out the levels in the decision tree with their related error terms with cross-validation samples.

**Confusion Matrix**: A matrix (or table) that lists out all of the possible prediction results when we validate our model against our validation set. This confusion matrix is one of the best methods to review the accuracy and precision of your model as well as to understand any model bias in classifying your data points.

**Readings:** [**https://datamining.bus.utk.edu/Documents/Decision-Trees-for-Predictive-Modeling-(Neville).pdf**](https://datamining.bus.utk.edu/Documents/Decision-Trees-for-Predictive-Modeling-(Neville).pdf)

1. **Model Comparison**

What we are going to do look for is the overall accuracy of each of the models as well as the lift/gains chart. The greater the area between the lift curve and the baseline, the better the model

* **lift/gains chart**

Lift is a measure of the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model.

[**http://www2.cs.uregina.ca/~dbd/cs831/notes/lift\_chart/lift\_chart.html**](http://www2.cs.uregina.ca/~dbd/cs831/notes/lift_chart/lift_chart.html)

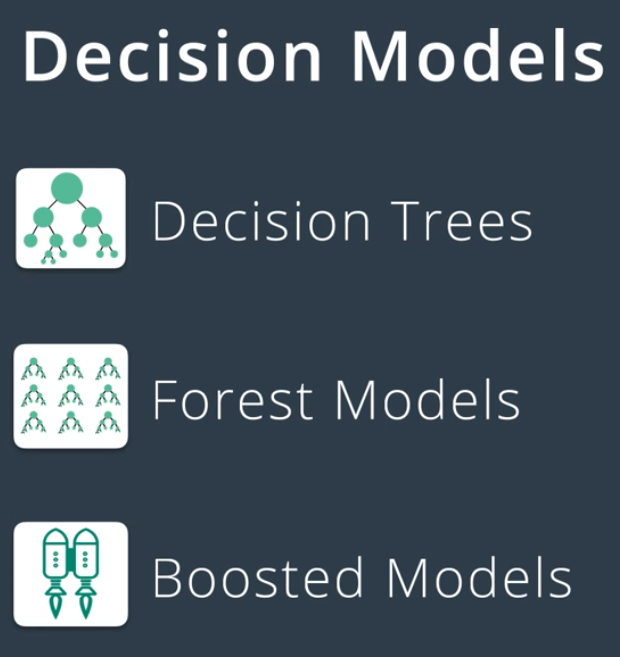
* **Precision vs Recall chart**

[**https://www.quora.com/What-is-Precision-Recall-PR-curve**](https://www.quora.com/What-is-Precision-Recall-PR-curve)

* **ROC chart**

[**http://www.dataschool.io/confirm-email/**](http://www.dataschool.io/confirm-email/)

1. **Non-Binary Classification Problems**



1. **Forest Model**

**Decision Trees**

Decision Trees are prone to an error called **overfitting**, where the model fits the sample data too well, and as a result, does not predict future results as well as it should.

**Random Forest Model**

* A Forest Model creates hundreds of trees, called an **ensemble of decision trees**
* Different randomly generated chunks of the original data create each tree.
* It looks at the results as a whole to make a prediction.
* The most often results will be the best

Each individual tree created still has overfitting issues, but when you look at the results as a whole, the overfitting **gets averaged out** by all of the other trees.



**bootstrapping** is any test or metric that relies on [random sampling with replacement](https://en.wikipedia.org/wiki/Random_sampling_with_replacement).

#### Important Definitions

* **Out of the Bag Error Rate (OOB estimate of error rate)**

Explains how well the model performed with the cross-validation set in the estimation data. This gives a good understanding of how solid the model performs with just the estimation data.

You can think of it in the same terms as an R-squared.



* **Confusion Matrix**

Shows again how well the model performed on the original, estimation data.

Compared to the "Out Of The Bag Error Rate", the confusion matrix does a better job at representing where errors occurred in classifying the data.

* **The Percentage Error for Different Number of Trees graph**

Helps us see what the correct number of trees is to use, so we can avoid over computing.

What we are looking for is the number of trees it takes to minimize the error of each of the items, so basically, where does it flatline?

After we determine the ideal number of trees, we can change subsequent Forest Models and run our data with the smaller number of Decision Trees.

* **Predictor Variables**

Which predictor variables matter the most in relation to this model? This is very helpful in determining which variables are most associated with our data on and we can focus on for future analysis.

1. **Boosted Model**

Forest Models might give us a better estimate than decision trees, but they're **computationally intensive.**

What we need is a model that can be both **accurate AND fast**. What we'll use to achieve this balance is known as the **Boosted Model**.

**How the Boosted model avoids overfitting**

* Instead of creating a bunch of random trees, the boosted model **makes one tree**.
* Algorithm performs an analysis on the errors of the tree to identify the biggest source of error.
* Changes the tree to reduce that error.
* Does the analysis again to find the next biggest error.
* Makes a change to reduce it.
* Does this over and over until it can’t make the tree any better and we have our finished Boosted Model.

To learn more of the mathematical foundation of this model just check out [**this link**](https://en.wikipedia.org/wiki/Gradient_boosting). **(Gradient boosting)**