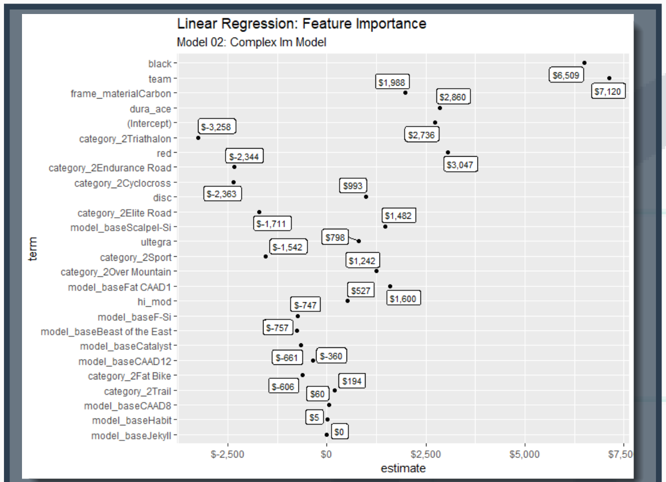
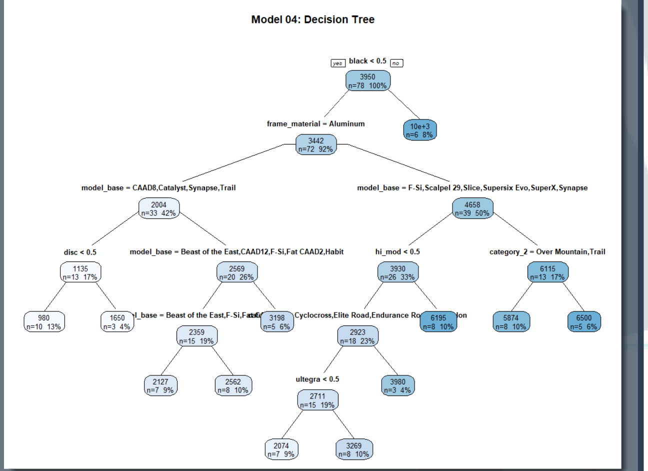
Regression (Machine Learning)



Linear Regression with Multiple Predictors

Each predictor (term) is interpretable meaning the value (estimate) indicates an increase/decrease in the target



Decision Tree (Regression)

Decisions are based on binary rules that split the nodes and terminate at leaves. Regression trees estimate the value at each node.

Summary:

- **Common Applications in Business**: Used to predict a *numeric value* (e.g. forecasting sales, estimating prices, etc).
- **Key Concept:** Data is usually in a rectangular format (like a spreadsheet) with one column that is a *target* (e.g. price) and other columns that are *predictors* (e.g. product category)
- **Gotchas:**
    - **Preprocessing:** Knowing when to preprocess data (normalize) prior to machine learning step
    - **Feature Engineering:** Getting good features is more important than applying complex models.
- **Parameter Tuning:** Higher complexity models have many parameters that can be tuned.
- **Interpretability:** Some models are more explainable than others, meaning the estimates for each feature means something in relation to the target. Other models are not interpretable and require additional tools (e.g. LIME) to explain.

Terminology:

- **Supervised vs Unsupervised:** Regression is a supervised technique that requires training with a "**target**" (e.g. price of product or sales by month). The algorithm learns by identifying relationships between the target & the **predictors** (attributes related to the target like category of product or month of sales).
- **Classification vs Regression:** Classification aims to predict classes (either binary yes/no or multi-class categorical). Regression aims to predict a numeric value (e.g. product price = $4,233).
- **Preprocessing**: Many algorithms require preprocessing, which transforms the data into a format more suitable for the machine learning algorithm. A common example is "**standardization**" or scaling the feature to be in a range of [0,1] (or close to it).
- **Hyper Parameter & Tuning:** Machine learning algorithms have many parameters that can be adjusted (e.g. learning rate in GBM). Tuning is the process of systematically finding the optimum parameter values.
- **Cross Validation**: Machine learning algorithms should be tuned on a validation set as opposed to a test set. Cross-validation is the process of splitting the training set into multiple sets using a portion of the training set for tuning.
- **Performance Metrics (Regression)**: Common performance metrics are **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**. These measures provide an estimate of model performance to compare models to each other.

R Cheat Sheet

Parsnip (Machine Learning):

- **Model List (start here first)**
- **Linear Regression & GLM**
- **Decision Tree**
- **Random Forest**
- **Boosted Trees (XGBoost)**
- **SVM: Poly & Radial**

Keras (Deep Learning)

H2O (ML & DL Framework)

MLR (ML Framework)

Python Cheat Sheet

Scikit-Learn (Machine Learning):

- **Linear Regression**
- **GLM (Elastic Net)**
- **Decision Tree (Regressor)**
- **Random Forest (Regressor)**
- **AdaBoost (Regressor) , XGBoost**
- **SVM (Regressor)**

Keras (Deep Learning)

H2O (ML & DL Framework)

Resources

- Business Analysis With R Course (DS4B 101-R) - Modeling - Week 6
- Business Science Problem Framework
- Ultimate R Cheat Sheet | Ultimate Python Cheat Sheet

Data Science Courses for Business

Business Science University
university.business-science.io

version: 1.1

Machine Learning Algorithms - Regression

Key Attributes Table

Popular Algorithms	Type	Key Concepts	Feature Range Standardization [0, 1]	Results Interpretable?	Key Parameters
Linear Regression	Linear	Simplest method - Uses OLS to reduce error and find the	Not Required	Yes - Model terms indicate magnitude / direction of each features contribution	N/A
GLM (Generalized Linear Model) LASSO, Ridge Regression, Elastic Net	Linear	Linear method that penalizes irrelevant features using a concept called "Regularization", where the weight of irrelevant features is reduced to make their effect on the model lower. L1 Regularization - Called LASSO regression L2 Regularization - Called Ridge Regression Elastic Net: Combines L1 & L2 Regularization	Required (but see Application Note below). **Application Note**: In practice, some algorithms (i.e. R's glmnet::glmnet()) implement standardization internally and re-scale prior to returning term estimates and predictions. This means that features need not be scaled prior to use.	Yes, if standardization is performed internally to algorithm. Model terms indicate magnitude / direction of each features contribution	Penalty (alpha) - How much to penalize the parameters Mixture (L1 Ratio) - Ratio between L1 and L2 Regularization
Decision Tree	Tree-Based (Non-Linear)	A decision tree is a set of decision rules. Each rule is considered a node with a split being a binary decision. The decisions terminate at a leaf.	Not Required	Yes - Decision Tree Plots show rule-based decisions that show how to arrive at model prediction	Max Tree Depth - How many splits for the longest tree Min Samples Per Leaf / Node - How many samples in each end node (leaf) Cost Complexity (Cp) / Min Impurity - Instructs when to stop (create a leaf) if additional information gain is not above a Cp threshold
Random Forest	Tree-Based (Non-Linear)	Ensemble learning method where many trees are created on sub-samples of data set and combined using averaging. This process controls overfitting, typicallying leading to a more accurate model. However, because the models are combined, the decision rules become incomprehensible. This process is often called "Bagging".	Not Required	No (see Application Note) **Application Note:** Feature importance can be obtained with additional methods for global (Variable Importance) and local (e.g. LIME) model understanding.	See Decision Tree Key Parameters, and: Replacement - whether or not to draw samples with replacement Number of Features - How many columns to use when sampling Number of Trees - How many trees to average
GBM (Gradient Boosted Machine) XGBoost	Tree-Based (Non-Linear)	Implements a technique called "Boosting" to build decision trees of weak prediction models and generalizes using a loss function. The weak learners converge to a strong learner.	Not Required	No (see Application Note) **Application Note:** Feature importance can be obtained with additional methods for global (Variable Importance) and local (e.g. LIME) model understanding.	See RandomForest Key Parameters, and: Learning Rate (eta) - The rate that the boosting algorithm adapts Loss Reduction (gamma) - The loss function to use during splitting Sample Size - The proportion of data exposed to the model during each iteration
SVM (Support Vector Machine)	Kernel Basis (Polynomial or Radial) (Non-Linear)	An algorithm that uses a kernel to transform the feature space to linearly seperable boundaries, and then applies a margin penalizing points that are incorrectly measured outside of the margin. The kernel transformation (i.e. radial, polynomial) makes it possible to perform linear separations within non-linear data.	Required (but see Application Note below). **Application Note**: In practice, some algorithms (i.e. R's kernlabs::ksvm()) implement standardization internally and re-scale prior to returning term estimates and predictions. This means that features need not be scaled prior to use.	No (see Application Note) **Application Note:** Feature importance can be obtained with additional methods for global (Variable Importance) and local (e.g. LIME) model understanding.	Kernel - Polynomial or Radial Basis Function Cost / Regularization - Cost of predicting sample on wrong side of the SVM margin Margin (Epsilon) - Specifies region where no penalty is applied Degree (Polynomial) - Degree of Polynomial. Use 1 for linear, 2 or more for flexible (quadratic) Scale Factor (Polynomial) - Factor to adjust bias/variance Gamma or Sigma (Radial) - Factor to adjust bias/variance
Deep Learning (Neural Network)	Neural Network (Non-Linear)	Learning algorithms with input and output and layers in between where the model parameters are learned. The user develops the architecture of the neural network, and the algorithm learns the model through iteratively seeking to minimize a cost function.	Required	No (see Application Note) **Application Note:** Feature importance can be obtained with additional methods for global (Variable Importance) and local (e.g. LIME) model understanding.	Many tuning parameters & architecture decisions