Prof. Pedram Jahangiry
Utah State University: **Huntsman School of Business**

Machine Learning Type	Model	Model Type and use case	Description	Pros	Cons	Hyperparameters
Supervised	**Linear regression**	Linear - Parametric Used for regression only	Finds the "best fit" through all the data points.	- highly interpretable (giving significancy results) - very fast training because of closed form solution - no hyperparameter tuning required	- validity of linear regression assumptions - cannot capture complex relationships	none
	Polynomial regression	Linear - Parametric Used for regression only	Extending linear regression model to capture non-linearities	- interpretable for low values of d (giving significancy results) - Can capture polynomial relationships	- need to choose the right polynomial degree - notorious tail behavior (sensitive to outliers)	d: degree of polynomial
	Penalized regression: Ridge, LASSO and Elastic Net	Linear - Parametric Used for regression only	Linear method that penalizes irrelevant features using regularization L1 regularization: LASSO L2 regularization: Ridge combining L1 and L2: Elastic net	- Can be used for feature selection (reducing the dimension of the feature space) - interpretable	- Requires feature scaling	penalty (how much to penalize the parameters) L1 ratio: ration between L1 and L2 regularization
	Logistic regression	Linear - Parametric Used for classification only	Basically the adaptation of linear regression to classification problems.	- probabilitsitc model (the outputs are probabilities) - highly interpretable (giving significancy results) - easy to understand - fast and efficient	- validity of linear regression assumptions - sensitive to extreme values - cannot capture complex relationships	- the same as penalized regression if regularization is used
	KNN	Non-linear - Non-parametric Used for both regression and classification	Make prediction for a new observation by finding similarities ("nearness") between it and its k nearest neighbors in the existing dataset.	- Intuitive and simple - Easy to implement for multi class problem - Few parameters/hyper parameters - No assumption (non parametric)	- Choice of K - Slow (memory based approach) - Curse of dimensionality - Hard to interpret - Requires feature scaling - Not good with multiple categorical features	- K value - distance metrics
	SVM	Kernel basis (non-linear) Linear SVM is parametric Kernel SVM is non-parametric Used for both regression and classification	Uses a kernel to transform the feature space to linearly separable boundaries	- SVM can be memory efficient! uses only a subset of the training data (support vectors) - Can handle non linear data sets - Can handle high dimensional spaces (even when D>N) - Linear SVM are not very sensitive to overfitting (soft margin; regularization) - Can have high accuracy (even compared to NN)	- Requires feature scaling - No probability outcome! - Does not perform well with noisy data - Limited interpretability (specially for Kernel SVM) - Memory intensive: Long training time when we have large data sets.	- Kernel: linear, rbf, poly, ... - C: Cost of misclassification - Gamma (for rbf): how far the influence reach - d (for poly): degree of polynomial
	Decision Trees	Tree-based (non-linear) Non-parametric Used for both regression and classification	- Progressively divide data sets into smaller data groups based on a descriptive feature , until they reach sets that are small enough to be described by some label	-Easy to interpret and visualize - Can easily handle categorical data without the need to create dummy variables - Can easily capture Non linear patterns - Can handle data in its raw form (no preprocessing needed). - No assumption (non parametric) - Can handle colinearity efficiently	- Sensitive to noisy data. It can overfit noisy data. Small variations in data can result in the different decision tree - Can lead to overfitting - Poor level of predictive accuracy	- Max tree depth, min samples per leaf (node), min samples split - Cost complexity alpha - Criterion: gini/entropy/ ...
	Random Forest	Ensemble method (non-linear) Non-parametric Used for both regression and classification	- Many trees are created on bootstrapped data and combined using averaging.	All the advantages of Decision Trees + - Typically more accurate - Avoid overfitting by reducing the model variance. - very flexible and parallelizable! - No data preprocessing (no feature scaling) - Great with high dimensionality	- no interpretability - complexity - many hyper parameters - slow on large data sets	DTs parameters + - m: subset of features - B: number of bootstrapped trees
	Boosting (XGboost)	Ensemble method (non-linear) Non-parametric Used for both regression and classification	- Implements boosting to build decision trees of weak prediction models and generalizes using a loss function.	All the advantages of Random Forests + - Regularization for avoiding overfitting - Efficient handling of missing data - In-built cross validation capability - Cache awareness and out-of-core computing - Tree pruning using depth-first approach - Parallelized tree building	- no interpretability - many hyper parameters	RF parameters + - Regularization terms

Prof. Pedram Jahangiry
Utah State University: **Huntsman School of Business**

Machine Learning Type	Model	Model Type and use case	Description	Pros	Cons	Hyperparameters
Unsupervised	**Principle Component Analysis (PCA)**	Non- Parametirc Used for dimension reduction	Principal components are vectors that define a new coordinate system in which the first axis goes in the direction of the highest variance in the data. The second PC is orthogonal to PC1 and etc.	- Reducing the number of features to the most relevant predictors is very useful in general. - Dimension reduction facilitates the data visualization in two or three dimensions. - Before training another supervised or unsupervised learning model, it can be performed as part of EDA to identify patterns and detect correlations . - Machine learning models are quicker to train , tend to reduce overfitting (by avoiding the curse of dimensionality), and are easier to interpret if provided with lower dimensional datasets.	- Hard to interpret - Requires feature scaling	None
	K-Means	Cab be both Parametric and Non-Parametirc Used for clustering the data	- Uses a measure of similarity to detect groups within data set: K means is an algorithm that repeatedly partitions observations into a fixed, pre-specified number of clusters	- Simple to understand - The k means algorithm is fast and works well on very large datasets - Can help visualize the data and facilitate detecting trends or outliers.	- Need to choose k before running the algorithm - Requires feature scaling - Poor performance with clusters of irregular shapes - Not applicable for categorical data - Unable to handle noisy data	K: Number of clusters - Distance metrics
	Hierarchical clustering	Cab be both Parametric and Non-Parametirc Used for clustering the data	- Uses a measure of similarity to detect groups within data set: Hierarchical clustering is an iterative procedure used to build a hierarchy of clusters	- The optimal number of clusters can be obrained by the model itself,	- The choice of distance metrics and linkage methods can be tricky - Requires feature scaling	- Distance metrics - Linkage methods