|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Supervised Algorithms | Purpose | Variants | Assumptions | Example Use |
| k-Nearest Neighbor | A simple baseline algorithm that builds models that classifies based on either the average of the closest training set neighbors (regressor) or the majority rules (classifier).  A good starting point model. | * KNeighborRegressor * KNeighborClassifier | 1. Small dataset 2. Few Features 3. Non-sparse dataset 4. Cannot extrapolate beyond training set 5. Must adjust n\_neighbor | To build a two-way classification model from a small dataset where the points separate out into fairly discrete groups where a more advanced algorithm doesn’t improve accuracy. |
| Linear Regression | Highly scalable and fast algorithm to build predictive regression models.  Particularly good when dataset is large, highly dimensional, and sparse. | * Ridge regularization – reduces coefficient magnitude (L2) * Lasso regularization – reduces coefficient magnitude and sets some feature coefficient to zero (L1) | 1. Can’t adjust feature coefficient or y-intercept parameters 2. For regularization, alpha must be set 3. Many data points 4. Many features 5. A somewhat normalized feature distribution | Build a regression model to predict the property value of the Boston housing market based on historical data.  In general predict the relationship between any two variables. |
| Logistic Regression | Linear model with an imposed cost function effective for building binary classification models.  Shares the same benefits and assumptions as linear regression. | * Can impose L1 or L2 regularization | 1. Adjust alpha and C parameter | Build a classification model to predict cancer or non-cancer outcomes based on many features inputs.  Fair accuracy score but is often an initial classification model to build. |
| Linear Support Vector Machine | Similar to Logistic Regression but effective at building >2 way classification models using the one-vs-all method. | * Impose L1 or L2 regularization | 1. Adjust alpha and C parameters | Build a 3 way classification model to predict individual low, medium, and high risk of diabetes group. |
| Naïve Bayes | Similar to the linear models but is even more efficient because it builds models/makes predictions by looking at per-class statistics from each feature.  **Only used for classification**. A good baseline model that is less accurate than linear models but can be built first to see performance. Great for very large datasets and highly dimensional data. | * Gaussian (Continuous) * Bernoulli (Binary) * Multinomial (Count) | 1. Choose the correct type of classifier 2. Alpha for Multinomial and Bernoulli | Create a classification model to filter spam email or classify documents based on sparse text count data. |
| Decision Trees | An algorithm that defines rules that best splits the data points into its classes. Similar to a game of 20 questions.  Very fast, simple algorithms that doesn’t require scaling/pre-processing of the data. Can provide information about feature importance.  Generally yields worst accuracy than linear or Naïve Bayes model. | * DecisionTreeClassifier * DecisionTreeRegressor | 1. Must pre-prune using max\_depth, max\_leaf\_nodes, or min\_samples\_leaf 2. Binary and/or continuous data | Best used to visualize and quantify how the tree splits features to classify points. What features are most critical for separating out the data points into cancer or non-cancerous tumors?  Tree-based algorithms are best used when features have very different meanings. |
| Random Forest | Ensemble tree method that builds many trees through bootstrapping (randomly selected features, number of features, and number of data points). Trees are then merged to build an overall good, stable model fit.  Very robust and powerful algorithm that performs better than a single decision tree, but is not good for high-dimensional sparse data. In which case, linear models would be better. | * RandomForestClassifier * RandomForestRegressor | 1. Doesn’t require data scaling 2. Must pre-prune using n\_estimator, max\_features, max\_depth 3. Large datasets 4. Low-Moderate level of dimensionality | Best used for classification problems when you have a range of features but only some are important or if features have skewed distributions. For example, you are trying to classify patients who are highly likely to be readmitted and you collect lots of different types of data that may or may not be relevant. |
| Gradient Boosted | Another ensemble tree method that builds multiple trees but that are shallow (uses few features) and corrects for the poor performance of previous trees. Iteratively creating better trees.  Almost always performs better than random forest but is slower to train the model. Requires more parameter tuning than Random forest.  Best to perform random forest first for classification and then improve accuracy with Gradient boosted. | * GradientBoostedClassifier * GradientBoostedRegressor | * Doesn’t require data scaling * Must pre-prune using n\_estimator, max\_features, max\_depth * Must set learning\_rate * Large datasets * Low-Moderate level of dimensionality | Used to improve the performance of classification models that performed well using Random Forest.  Trees are faster models to build and generally much more accessible. |
| Kernalized Support Vector Machines | Can build complex, powerful models with higher accuracy by adding interactions or polynomials of the features that increase dimensional space and Gaussian kernel that can better fit data points.  Best used for medium-sized datasets of features with similar meaning. Requires scaling of data and careful parameter setting. | * LinearSVC * RegressorSVC | * Must tune C and Gamma parameters * Manually code in kernel trick * Requires pre-processing of data * 10,000-90,000 samples * Features with similar meaning | Build a model for classifying high and low risk for developing IBS based on transcriptomics data or microbiome community structure matrix. Or for fitting data where one class label could exist in separate clusters, predicting people with eating disorders (anorexia and obesity). |
| Neural Networks | A generalization of linear models but performs many processing layers to come to conclusion. Neural network builds highly complex models calculated by hidden layers based on feature interactions.  Used on large datasets. Very sensitive to scaling of the data and choice of parameters. Requires long time to train. | * MLPClassifier * MLPRegressor | * Features are similar in meaning * Requires pre-processing * Requires careful parameter tuning: alpha, max\_iter, hidden\_layer\_sizes, solver, activation | Best used to model datasets with highly interconnected features from complicated systems where simpler models fail fit the data.  Deep learning is commonly used for a range of complex purposes: radiology imagery, language translation, self-driving cars |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unsupervised Algorithms | Purpose | Variants | Assumptions | Example Use |
| Principal Coordinate Analysis (PCA) | Method of reducing dimensionality of a dataset (decomposition) by rotating and dropping features in a way that becomes uncorrelated (i.e., orthogonal).  Not used as a final model but is often helpful for data visualization or transforming the data in such a way that is more informative for supervised model building. | * Non-negative matrix factorization * Manifold learning with t-SNE | * Features should be scaled | A flexible method that is best used for datasets with many more features than sample numbers where not all features are important.  For example, looking at whether fish community structure of different sites separate out or whether cancer vs non-cancer sample separate out. |
| Non-Negative Matrix Factorization (NMF) | Another decomposition method similar to PCA but instead returns components and coefficients as non-negative numbers. Thus making it slightly easier for feature extraction.  Often performs worst than PCA at reconstructing or encoding data, but is good for finding patterns |  | * Scale features * Used on datasets where features are non-negative | Best for data that is created as an overlay of several independent sources. For example, an auto track |
| Manifold Learning with t-SNE | Another similar decomposition method that is often used for better visualization. |  |  |  |
| k-Means Clustering | A scalable, simple algorithm that creates clusters based on assigning center points to groups of data points that reduce the distance within group.  A decomposition method that characterizes clusters based on cluster means. | 1. Vector quantization | * Must set n\_clusters parameter * Similar variance among clusters * Discrete clusters, non-stretched | Best used for relatively simple dataset where we expect the groups to be similar sized. Doesn’t handle noise well. |
| Agglomerative Clustering | A group of clustering methods that  It works by declaring each point as its own center and merges clusters based on Euclidean distance until some stopping criteria.  Similar in structure to a bifurcating phylogeny tree. | 1. *Ward*: Default. Merges clusters so that variance w/n clusters increase the least. 2. *Average*: Merges clusters with smallest average distance among all points. 3. *Complete*: Merges two clusters with smallest max distance between points. | * Set stopping criteria: number of cluster and linkage criteria * Must set method to use * Fails at separating complex shapes | Best used for datasets with s |
| DBSCAN | An algorithm that identifies clusters based on density and space. The algorithm iteratively finds points by dropping core points randomly and determining cluster or noise through finding points around it in a certain distance.  Can capture complex shapes and identify noise. |  | * Do not have to pre-set number of clusters * Set min\_samples and eps to define core samples parameters * Good idea to scale data | Best used on datasets with complex cluster shapes and lots of noise. |