
Live Session 1

Machine Learning II:

K-NN

Designed by: Sudipta Dasmohapatra & Babak Zafari
Delivered by: Tommy Jones



Welcome Back: Pop Quiz!

Match the ML method, left, with its definition, right.

A. Supervised learning

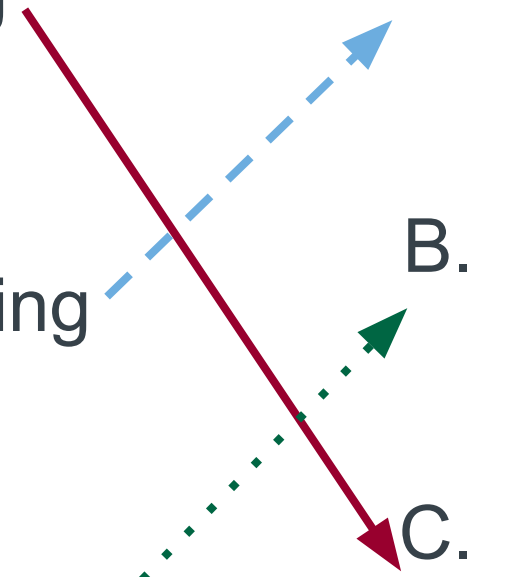
B. Unsupervised learning

C. Reinforcement learning

1. We want to find patterns independent of any known outcome.
2. An algorithm interacts with its environment within a structure of rewards and penalties.
3. We want to predict a known outcome.

Welcome Back: Pop Quiz!

Match the ML method, left, with its definition, right.

- 
- 1. Supervised learning
 - 2. Unsupervised learning
 - 3. Reinforcement learning
- A. We want to find patterns independent of any known outcome.
 - B. An algorithm interacts with its environment within a structure of rewards and penalties.
 - C. We want to predict a known outcome.

Topics We Will Cover

Week 1	Tuesday / Jan 14	K-nearest Neighbors	Supervised
Week 2	Tuesday / Jan 21	Support Vector Machines	Supervised
Week 3	Tuesday / Jan 28	Artificial Neural Networks	Supervised
Week 4	Tuesday/ Feb 4	Principal Component Analysis	Unsupervised
Week 5	Tuesday / Feb 11	Clustering Methods – Part 1	Unsupervised
Week 6	Tuesday / Feb 18	Clustering Methods – Part 2	Unsupervised

Assignments

Project 1: Due Monday January 27th at 3 AM Eastern

Project 2: Due Monday February 10th at 3 AM Eastern

Quiz 1: Due Monday February 10th at 3 AM Eastern

Final Project: Due Monday February 24th at 3 AM Eastern

Quiz 2: Due Monday February 24th at 3 AM Eastern

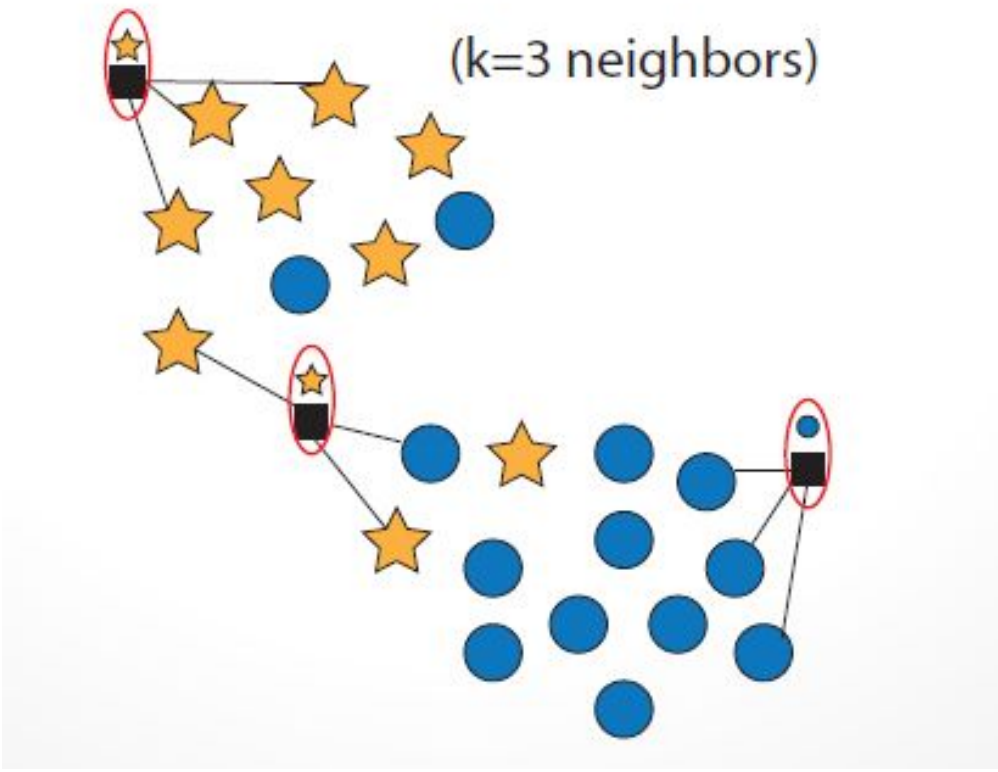
K-Nearest Neighbors

k-NN: Intuition

Identify several cases that are most similar to a given observation

Use the information from those “neighbors” to classify or predict the new observation

The nearest neighbors are defined by the value of “k”



Distance Measures & Variable Scaling

“Nearest” implies a distance measure.

Euclidean distance is most common.

Common definition of distance.

Straight line between points.

Many distance measures exist.

Won't cover more for K-NN.

Euclidean is a good start.

Euclidean distance is sensitive to scale.

Variables with bigger numbers will overwhelm the calculation.

Similar to “influential observations” in regression.

So, for k-NN, we rescale variables using normalization or standardization when using Euclidean distance.

Euclidean Distance

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Categorical Variables

- Categorical variables can not be used directly in distance functions, such as Euclidean.
- In order to use them, they should be converted to dummy/indicator variable or a different distance measure, that handles categorical variables, should be used.
- If they are used in combination of numerical variables (i.e., mixed data), they could be scaled similar to the numerical variables. But this is an optional step and could use these variables they way they are (0/1). In that case, you can attach them to the set of already-scaled numerical variables to form the final datasets (training, validation, etc.) used in your analysis.

Confusion Matrix

- **True Negative (TN):** Correctly predicted as negative
- **False Positive (FP):** Incorrectly predicted as positive
- **False negative (FN):** Incorrectly predicted as negative
- **True Positive (TP):** Correctly predicted as positive

- **Accuracy** = $\frac{TN+TP}{Total\ records}$

- **Sensitivity(Recall)** = $\frac{TP}{TP+FN}$

- **Precision** = $\frac{TP}{TP+FP}$

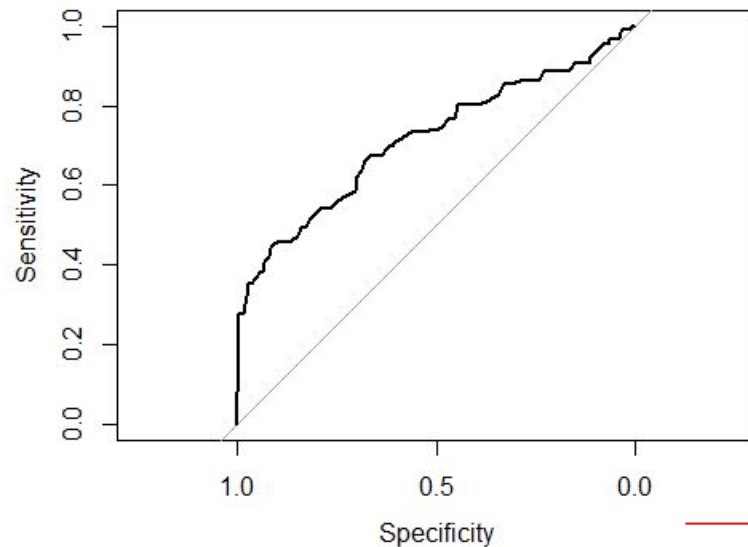
- **F-1 score:** harmonic mean of precision and recall

- **Specificity** = $\frac{TN}{TN+FP}$

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

ROC Curve

- A popular graph for plotting two measures (*Specificity*, *Sensitivity*) is Receiving Operating Curve (ROC)
 - It plots the pair as the cutoff decreases from 1 to 0 (i.e., each point on curve represents a different value of the cut-off ranging from 0 to 1)
 - The line in the middle represents our benchmark: randomly guess TRUE or FALSE for each record
- ROC curves are useful for comparing different classifiers, since they take into account all possible thresholds. The overall performance of a classifier is given by the **area under the ROC curve (AUC)**, a value between 0.5 and 1!
 - The closer the curve to the top left corner, the larger area under the ROC → the better the classifier.



- It is common to see plots with “1-specificity” (i.e., false positive rate) values on the x axis.
- In those plots, the range of the axis is flipped to start from 0 going to 1.
- Regardless of the form, the interpretation of the plot stays the same.

Coding knn in R

In coding knn in our class, we try two different approaches:

- **Using the knn() function from the “class” package.**
 - This function doesn’t automatically scale the variables. We should do that before passing the data to the function.
 - In scaling the data, we should first split the data and use the information from the training data to scale the test data.
 - This function needs a pre-set value of k to be provided and does not search for an optimal value of k.
 - To get the predicted probabilities (rather than predicted labels), we should add the prob=TRUE argument to the function.
- **Using the train() function the “caret” package and setting the method=“knn”.**
 - We can set up the desired cross validation parameters using trainControl() function.
 - The train() function can either use already-scaled datasets or scale the variables on its own using the preProcess argument.
 - The tuneLength argument allows us to search over a set of k values in search of the best one.
 - To get the predicted probabilities (rather than predicted labels), we should use the type=“prob” in the predict function. This will generate two columns, where often the second column contain the probability of belonging to the positive class.

Demo

Exercise

Download and save *redwine.csv* dataset. You plan to apply the knn algorithm to this data.

Download *week-01-exercises.R* and follow the instructions in the comments.

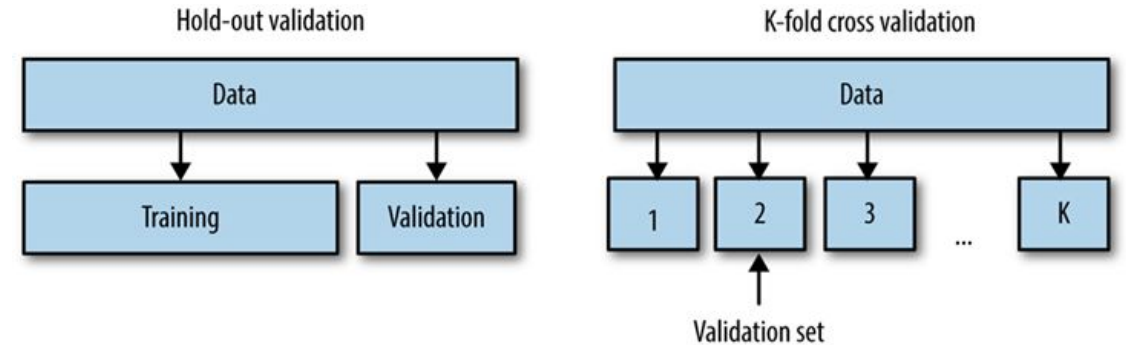
Backup

K-NN

- The k-nearest-neighbors (k-NN) is a method commonly used for classification. It can also be used for regression.
- It's data-driven, not model-driven and makes no assumptions about the data.
- Here k is the number of nearby neighbors to be used to classify the new record.
- Process: For a given record to be classified, identify nearby records. “Near” means records with similar predictor values x_1, x_2, \dots, x_p . Classify the record as whatever the predominant class is among the nearby records (the “neighbors”).
- The **Euclidean Distance** function is commonly used to measure distance.
- When measuring distance, one or more attributes can have very large values, relative to the other attributes. For example:
 - Income may range from 50,000 to 100,000 whereas age takes on values up to 100.
 - In this case, the values of income will overwhelm the contribution of age.
- To avoid this situation, we should standardize the variables.

Cross Validation and Scaling of the Variables

- In predictive modeling framework, we split the data into training and validation.



- So the question is:
 - How should we apply the rescaling?
 - Should we do it before or after the data split?
 - If after, should we rescale training and validation sets independently?
- The proper way of scaling variables:
 - **We always split the data before any imputation or scaling and rescale training and validation separately.**
 - **Since we're pretending 'test' (i.e., validation) to be an unseen dataset, we will use information from train data to impute the test dataset.**
 - **In doing so, we use the mean and standard deviation of the training dataset to standardize the validation dataset.**