Missingness, Imputation and KNN

What is Missingness?

Missingness occurs when at least some of an observation's values are not present within the dataset. We say that the absent values are "missing," and that the observation itself is "incomplete."

- A value could be missing because of many reasons (e.g., human error, carelessness in handling, an undefined mathematical computation, etc.).
- These reasons are often unknown by the data scientist who ultimately receives the data set.

Many statistical methods and machine learning techniques have difficulty incorporating incomplete observations in their algorithms; they simply don't know what to do when there isn't any data to crunch!

What to do with Incomplete Observations?

If we just delete all problematic observations from our dataset, wouldn't that fix the problem of our algorithms halting?

- While the solution of complete-case analysis seems to be the quickest and easiest on the surface, it can severely limit the amount of information available for our tests.
- With fewer observations in our dataset we have a smaller sample size, increasing the standard errors of any estimates we make.

Consider the case in which we have 100 variables within our dataset. Suppose an observation has 99 recorded values, and only 1 missing.

- If we were to take a complete-case standpoint, the entire observation would be omitted from the analysis -- including the 99 recorded values!
- What if we could fill in this 1 missing value? Then this observation would be complete and wouldn't cause us any problems.

The process of "filling in" missing values is called imputation; we will discuss methods of imputation a bit later.

Types of Missingness

Not only is there information contained within the recorded values of an observation with missingness, but sometimes the patterns of missingness can reveal information about the dataset.

1) An analysis of the missingness in our dataset can often help us decide what to do when faced with a set of incomplete observations.
2) There are three main types of missingness:
    - Missing Completely at Random (MCAR)
    - Missing at Random (MAR)
    - Missing Not at Random (MNAR)

Missing Completely at Random (MCAR)

1) When data are missing completely at random, each piece of data in the overall dataset has an equally likely chance of being absent.
    - The reason for the missingness is neither related to the observed variables nor related to the unobserved variables of interest; they are independent.
2) MCAR data is the best case scenario for missing data in general, because its manifestation is truly "completely at random"; unfortunately, data that is MCAR is also often the rarest form of missing data.
3) When data are MCAR, it is ok to ignore these observations; their deletion will not end up biasing your results because there is no underlying pattern that they reveal.
4) Examples of MCAR data:
    - You write a survey with 99 questions and distribute it to everyone in your office. You randomly select a few people to answer an additional 100th question.
        ■ The missing responses to the 100th question are MCAR because each individual was equally likely to not respond.
    - A piece of lab equipment is supposed to take a measurement on each specimen that it encounters. For one arbitrary specimen, the equipment malfunctions and the measurement is not recorded.
        ■ The missing measurement is MCAR because the malfunction had nothing to do with the specimen and could theoretically have happened to any other specimen.

Missing at Random (MAR)

1. When data are missing at random, the chance that a piece of data is missing is dependent on variables for which we have complete information within our overall dataset.
    - The probability a piece of data is missing depends on available information that we have already collected; they are not independent.
2. MAR is the next-best scenario for missing data after MCAR because, although each observation has a different likelihood of missing, we theoretically can estimate this likelihood.
3. When data are MAR, it is acceptable to drop these observations from our analysis if we control for the factors that are related to the missingness and adjust for their effects, we can avoid bias in our model.
4. Examples of MAR data:
    - You write a survey and ask both men and women to submit responses. In the survey, there is a section that asks questions about various sports teams. The women you surveyed are more likely to not respond to the sports-related questions.
        ■ The missing responses to the sports-related questions are MAR because they depend in part on another measured variable: gender.

- A photocopier takes a log of the amount of copies it is supposed to make, and the amount of ink it uses for each job. For particularly large amounts of copies, the machine has a higher chance of malfunctioning and thus not recording the amount of ink it used for the job.
    - ■ The missing ink measurements are MAR because they depend in part on another measured variable: the amount of copies.

Missing Not at Random (MNAR)

1. When data are missing not at random, the chance that a piece of data is missing is dependent on the actual value of the observation itself.
    - The value of the missing piece of data is directly related to the reason why it is missing in the first place.
2. MNAR is the worst-case scenario for missing data because it is non-ignorable. We cannot theoretically accurately estimate the missing values because the reason they are missing is not captured within our dataset.
3. When data are MNAR, it is not appropriate to drop these observations from our analysis; doing so would leave us with a biased dataset, and thus our analyses would return biased models.
4. Examples of MNAR data:
   You write a survey and ask individuals to report their weight. Individuals who are particularly overweight tend to not answer this question.
        - ■ The missing responses to the question about weight are MNAR because the actual weight measurement itself is related to the probability that the measurement will be missing.

   A scale is supposed to measure the weight of various items, but is not sensitive enough to detect weights that are less than 5 pounds, and thus does not record weights for such items.
        - ■ The missing weight measurements are MNAR because the actual weight measurement itself is related to the probability that the measurement will be missing.

Methods of Imputation

Mean value imputation procedure:

1) Compute the average of the observed values for a variable that has missingness.
2) Impute the average for each of the missing values.

❖ Advantages:

- One of the simplest ways of dealing with missing data because of its relatively straightforward approach.

❖ Disadvantages:

- Can distort the distribution of the variable and underestimate the standard deviation.
- Can distort relationships between variables by dragging correlation estimates towards 0.

Simple random imputation procedure:

- For each missing value in a variable, randomly select a complete value of the same variable; impute this randomly selected value.
- Repeat the process until all values are complete.

❖ Advantages:

- Uses true, observed values to fill in missingness.

❖ Disadvantages:

- Can amplify outlier observation values by having them repeat in the dataset.
- Can induce bias into the dataset.

Regression prediction procedure:

- Assume an underlying, linear structure exists in the data.
- Give weights to a subset of the complete variables.
- Use a relationship between the complete variables and the complete observations to impute missing observations.

❖ Advantages:

- Uses true, observed values to fill in missingness.
- Uses the relationships among multiple variables to fill in missingness.

❖ Disadvantages:

- Must make assumptions about the structure of the data.
- Can inappropriately extrapolate beyond the scope of available information in our dataset.

Pros & Cons of Imputation

The pros of imputation:

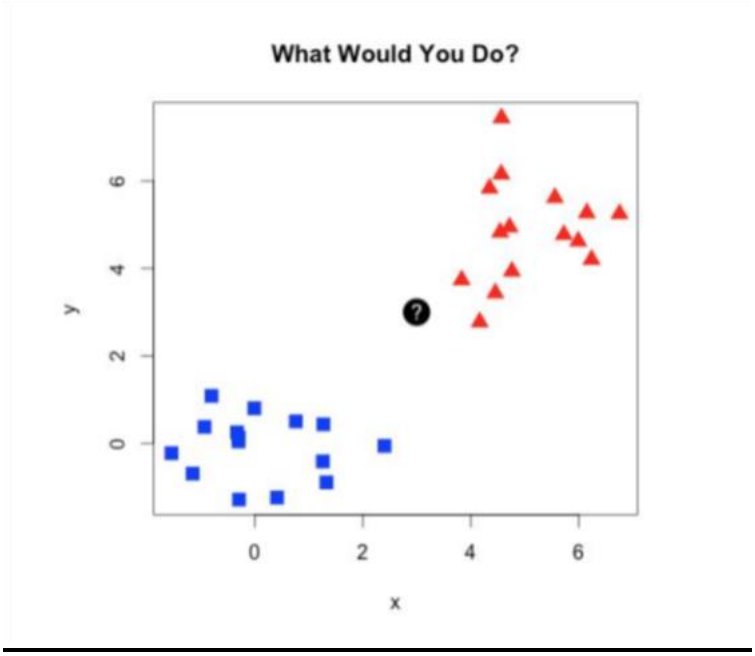
- Helps retain a larger sample size of your data.
- Does not sacrifice all the available information in an observation because of sparse missingness.
- Can potentially avoid unwanted bias.

The cons of imputation:

- The standard errors of any estimates made during analyses following imputation can tend to be too small.
    - ■ The methods are under the assumption that all measurements are actually "known," when in fact some were imputed.

- Can potentially induce unwanted bias.

K-Nearest Neighbors



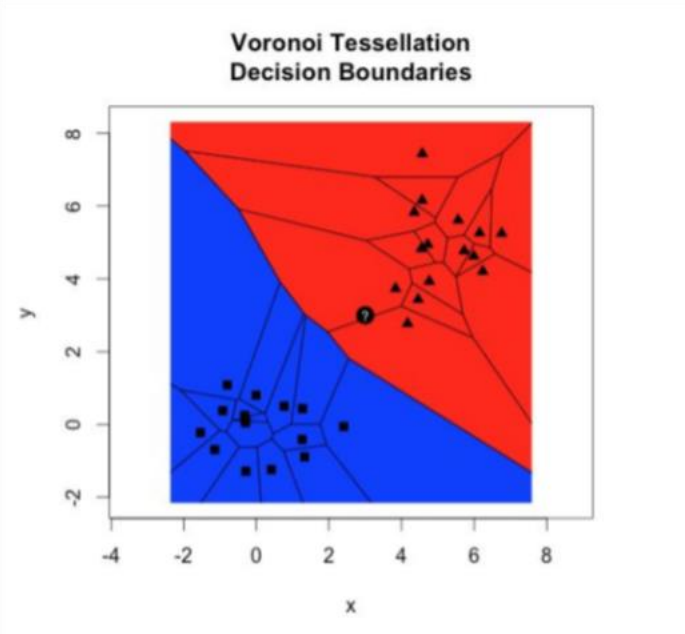
Introduction to K-Nearest Neighbors

1) The basic idea: Observations that are closest to an arbitrary point are the most similar.
2) Can be used in both classification and regression settings (i.e., can have output take the form of class membership or property values).
3) For K-Nearest Neighbors we find the K closest observations to the data point in question, and predict the majority class as the outcome.
    - For 1-Nearest Neighbors, the single closest observation is the sole vote.

Voronoi Tessellation: Classification

1) The KNN algorithm partitions the feature space into different regions that represent classification rules; these regions are called Voronoi tessellations.
    - Boundaries represent areas where distances are equal in respect to different observations.
2) By following the Voronoi tessellations, the overall decision boundary has the flexibility to be non-linear.

    1NN Example:



Limitations of 1- Nearest Neighbor:

1) While the algorithm is very simple to understand and implement, its simplicity comes along with some drawbacks:

- 1NN is unable to adapt to outliers; a single outlier can dramatically change the Voronoi tessellations, and thus the decision boundaries.
- There is no notion of class frequencies (i.e., the algorithm does not recognize that one class is more common than another).

2) One way to get around these limitations and to add some stability is to consider more neighboring points (increasing the value of K), and assessing the majority vote.

- What happens when we choose all neighbors?

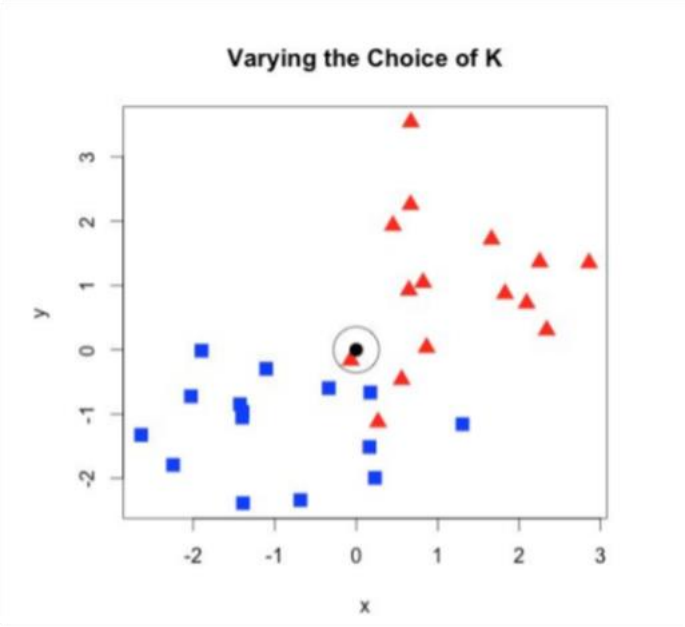
The Choice of K

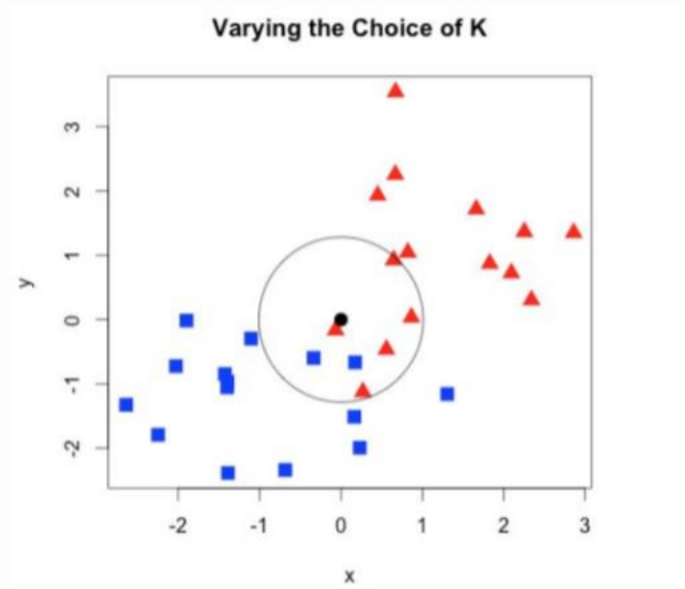
As we vary K the predicted classification rule will change, thus the choice of K has a large effect on the algorithm's performance. In general:

- Small values of K:
  - ■ Highlight local variations.
  - ■ Are not robust to outliers.
  - ■ Induce unstable decision boundaries.
- Large values of K:
  - ■ Highlight global variations.
  - ■ Are robust to outliers.
  - ■ Induce stable decision boundaries.

❖ NB: We will revisit the choice of K in more detail when we discuss the topic of **cross-validation**; however, in practice, a good balance is typically achieved with:

$$K = \text{sqrt}(n)$$





The Choice of Distance Measure

1) As we change the way we measure the distance between two points in our feature space, the classification rule will change. The choice of distance measure also has a large effect on the algorithm's performance.
2) The most common distance measure for continuous observations is called the **Euclidean distance**, defined as:

$$D(x1,x2) = sqrt(sum(|x1-x2|^2))$$

3) Euclidean distance is the "familiar" distance we typically use in everyday life; it is symmetric, treats all dimensions equally, and thus is sensitive to large deviations in a single dimension.

1) The most common distance measure for categorical observations is called the **Hamming distance**, defined as:

$$D(x1,x2) = sum(1) \text{ where } x1 != x2$$

2) Hamming distance looks at each attribute between observations and compares whether or not the observations are the same; each similarity is ignored while each difference is penalized.
   - The measure is symmetric and treats all dimensions equally.

1) Although rarely used, there are a plethora of other distance measure choices. One family of distance functions is called the **Minkowski p-norm**, defined as:

$$D(x1,x2) = \mathbf{p}root(sum(|x1-x2|\hat{}\mathbf{p}) \text{ where } x1 != x2$$

As we vary **p**, we define distance measures that each have different behaviors:

- **p** = 1: Manhattan block distance (adds each component separately).
- **p** = 2: Euclidean distance.
- **p** → ∞: Maximum distance, Logical Or (the largest difference among all attributes dominates the distance measure).

Breaking Ties

What do we do if there is a tie? More specifically, how do we decide to classify an observation whose K-nearest neighborhood has an equal number of maximum group memberships?

Some methods for breaking ties:

1) If there are only two groups, we can easily get around this by using an odd K. Why doesn't this work when there are more than two groups?
2) Use the maximum prior probability to uniformly decide all ties.
3) Randomly choose the group; for G groups:
   - Roll a G-sided die that has equally likely outcomes for each group.
   - Roll a G-sided die that has weighted outcomes for each group.
4) Use the 1NN to break the tie.

Pros & Cons of K-Nearest Neighbours

Pros of K-Nearest Neighbors:

- The only assumption we are making about our data is related to proximity (i.e., observations that are close by in the feature space are similar to each other in respect to the target value).
- We do not have to fit a model to the data since this is a non-parametric approach.

Cons of K-Nearest Neighbors:

- We have to decide on K and a distance metric.
- Can be sensitive to outliers or irrelevant attributes because they add noise.
- Computationally expensive; as the number of observations, dimensions, and K increases, the time it takes for the algorithm to run and the space it takes to store the computations increases dramatically.
    - Why is this bad? We want more data!