Missingness Imputation & KNN

Outline

- Missing Data
- Methods of imputation
- K-nearest neighbors

Missing data mechanisms

- Missing Completely at Random (MCAR)
  - Missing value (y) neither depends on x nor y
  - Example: some survey questions asked of a simple random sample of original sample
- Missing at Random (MAR)
  - Missing value (y) depends on x, but not y
  - Example: Respondents in service occupations less likely to report income
- Missing not at Random (NMAR)
  - The probability of a missing value depends on the variable that is missing
  - Example: Respondents with high income less likely to report income

Deal with missing data

- Use what you know about
  - Why data is missing
  - Distribution of missing data
- Decide on the best analysis strategy to yield the least biased estimates
  - Deletion Methods
  - Single Imputation Methods such as Mean(Medium)/Random substitution
  - Model-Based Methods such as KNN

Mean(medium) value imputation

- Replace missing value with sample mean/medium
- Run analyses as if all complete cases
- Advantages:
  - Can use complete case analysis methods
- Disadvantages:
  - Reduces variability
  - Weakens covariance and correlation estimates in the data (because ignores relationship between variables)

Simple random imputation

- Replace missing value with randomly selected value from data set
- Run analyses as if all complete cases
- Advantages:
  - Use true values to fill the missingness
- Disadvantages:
  - Amplify outlier observation values by having them repeat in the dataset
  - Induce bias into the dataset

k-nearest Neighbors

- Can be used in both regression and classification.
- Basic idea: find the K closest observations to the data point in question, and predict the majority class/mean value as the outcome

KNN classification application:

Let’s assume a money lending company who is interested in making the money lending system comfortable & safe for lenders as well as for borrowers. The company holds a database of customer’s details and it will calculate a credit score for each customer. The calculated credit score helps the company and lenders to understand the credibility of a customer clearly.

The customer’s details could be:

- Educational background details.
  - Highest graduated degree.
  - Cumulative grade points average (CGPA) or marks percentage.
  - The reputation of the college.
  - Consistency in his lower degrees.
  - Whether taken the education loan or not.
  - Cleared education loan dues.
- Employment details.
  - Salary.
  - Year of experience.
  - Got any onsite opportunities.
  - Average job change duration.

The company uses these kinds of details to calculate credit score of a customer. The process of calculating the credit score from the customer’s details is expensive. To reduce the cost of predicting credit score, they realized that the customers with similar background details are getting a similar credit score. So, they decided to use already available data of customers and predict the credit score using it by comparing it with similar data. These kinds of problems are handled by the k-nearest neighbor classifier for finding the similar kind of customers.

KNN classification and regression

Algorithm:

Missing values: Y_m, observations X_m
Complete values: Y_i, observations X_i

- Calculate the distance between X_m and each observation X_i
- Determine the K observations that are closest to X_m
- For regression , assign Y_m the mean of the Y measurements among the K selected observations
- For classification, classify Y_m as the most frequent class Y among the K selected observations

Distance Measure

The distance is calculated using one of the following measures

- Euclidean Distance (most common and for continuous variable)

$$D(x_1, x_2) = \sqrt[2]{\sum_d \mid x_{1d} - x_{2d} \mid^2}$$

Example: distance between (3,4) and (1,2)

Euclidean: ((3-1)^2+(4-2)^2)^0.5=2.828

Manhattan: |3-1|+|4-2|=4

Minkowski: if p=3 ((3-1)^3+(4-2)^3)^(1/3)=2.520

- Manhattan Distance (for continuous variable)

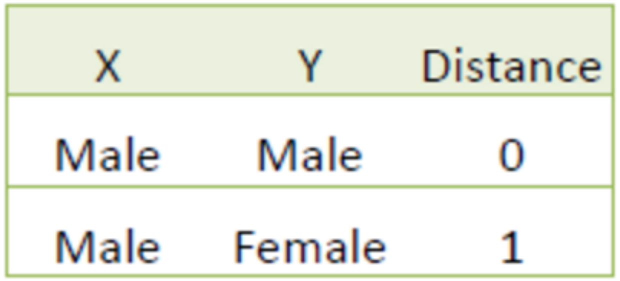
$$D(x_1, x_2)=\mid x_{1d}-x_{2d}\mid$$

- Minkowski Distance (customized and for continuous variable)
  - choice of p

$$D(x_1, x_2) = \sqrt[p]{\sum_d \mid x_{1d} - x_{2d} \mid^p}$$

- Mahalanobis Distance (for categorical variable)
  - $x_1=x_2$ D=0
  - $x_1!=x_2$ D=1

$$D(x_1, x_2) = \sum_d 1_{x_{1d} \neq x_{2d}}$$

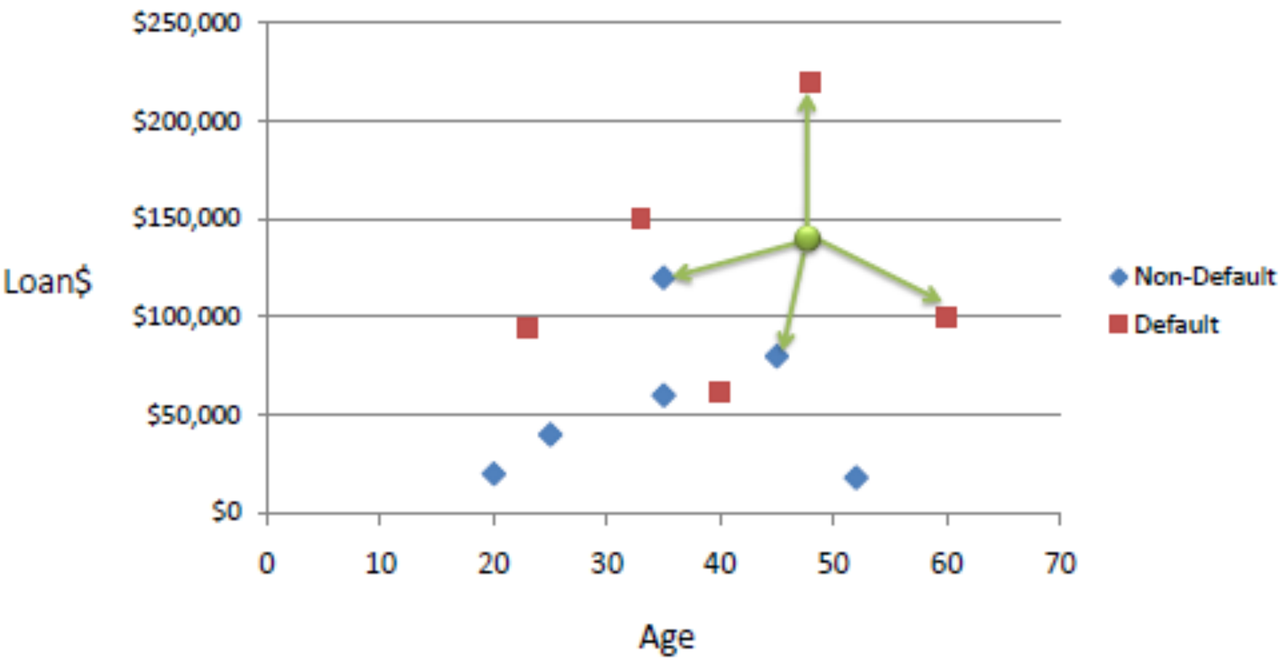


X	Y	Distance
Male	Male	0
Male	Female	1

Mahalanobis: 0+1=1

Example

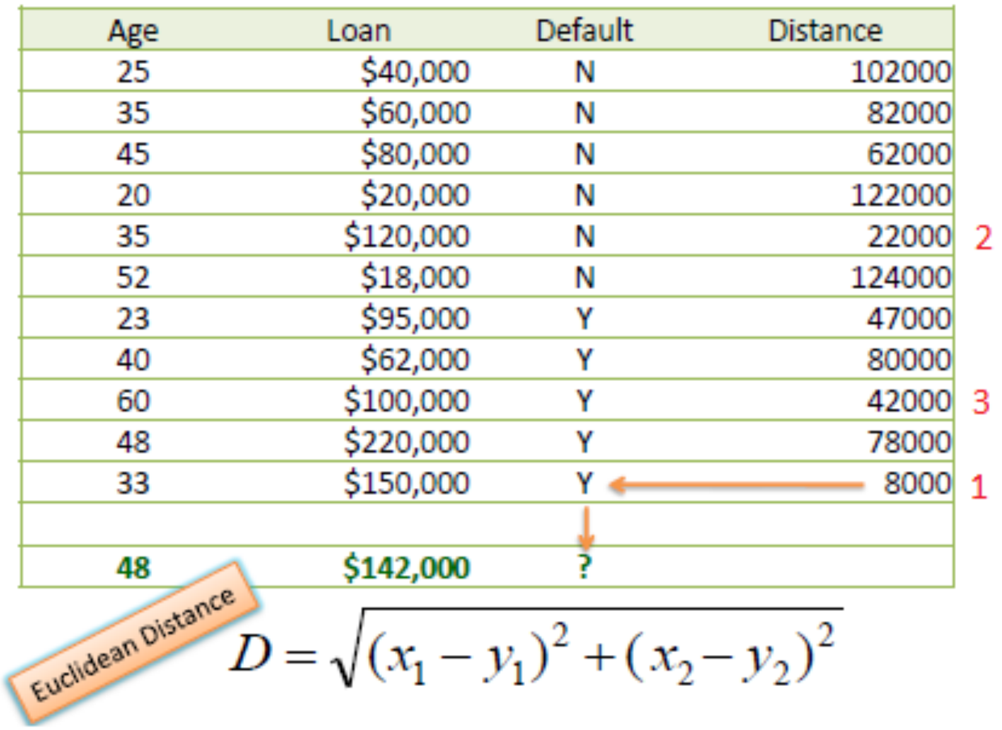
Consider the following data concerning credit default. Age and Loan are two numerical variables (predictors) and Default is the target.



We can now use the training set to classify an unknown case (Age=48 and Loan=$142,000) using Euclidean distance. If K=1 then the nearest neighbor is the last case in the training set with Default=Y.

Euclidean distance between (48, 142000) and (33, 150000):

D = Sqrt[(48-33)^2 + (142000-150000)^2] = 8000.01 >> Default=Y



Age	Loan	Default	Distance	
25	$40,000	N	102000	
35	$60,000	N	82000	
45	$80,000	N	62000	
20	$20,000	N	122000	
35	$120,000	N	22000	2
52	$18,000	N	124000	
23	$95,000	Y	47000	
40	$62,000	Y	80000	
60	$100,000	Y	42000	3
48	$220,000	Y	78000	
33	$150,000	Y	8000	1
48	**$142,000**	**?**		

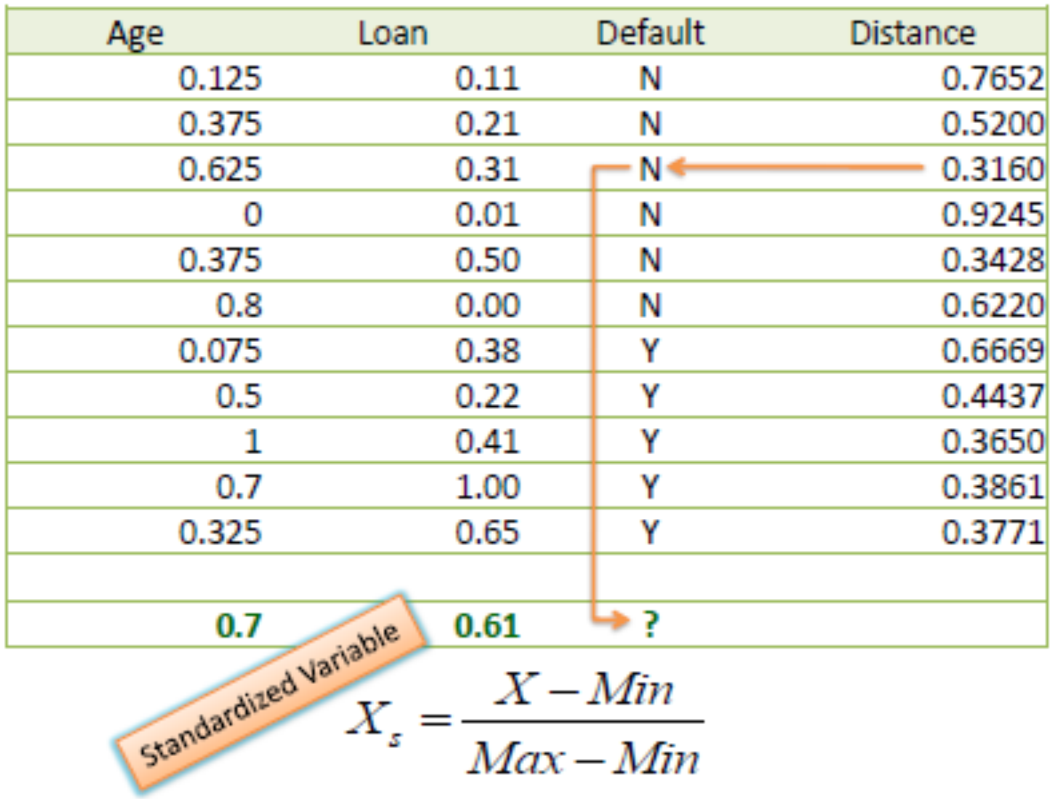
Euclidean Distance

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

With K=3, there are two Default=Y and one Default=N out of three closest neighbors. The prediction for the unknown case is again Default=Y.

Normalized Distance

- One major drawback in calculating distance measures directly from the training set is in the case where variables have different measurement scales or there is a mixture of numerical and categorical variables.
- For example, if one variable is based on annual income in dollars, and the other is based on age in years then income will have a much higher influence on the distance calculated.
- One solution is to normalized the training set as shown below.



Age	Loan	Default	Distance	
0.125	0.11	N	0.7652	
0.375	0.21	N	0.5200	
0.625	0.31	N	0.3160	1
0	0.01	N	0.9245	
0.375	0.50	N	0.3428	2
0.8	0.00	N	0.6220	
0.075	0.38	Y	0.6669	
0.5	0.22	Y	0.4437	
1	0.41	Y	0.3650	3
0.7	1.00	Y	0.3861	
0.325	0.65	Y	0.3771	
0.7	0.61	?		

Standardized Variable

$$X_s = \frac{X - Min}{Max - Min}$$

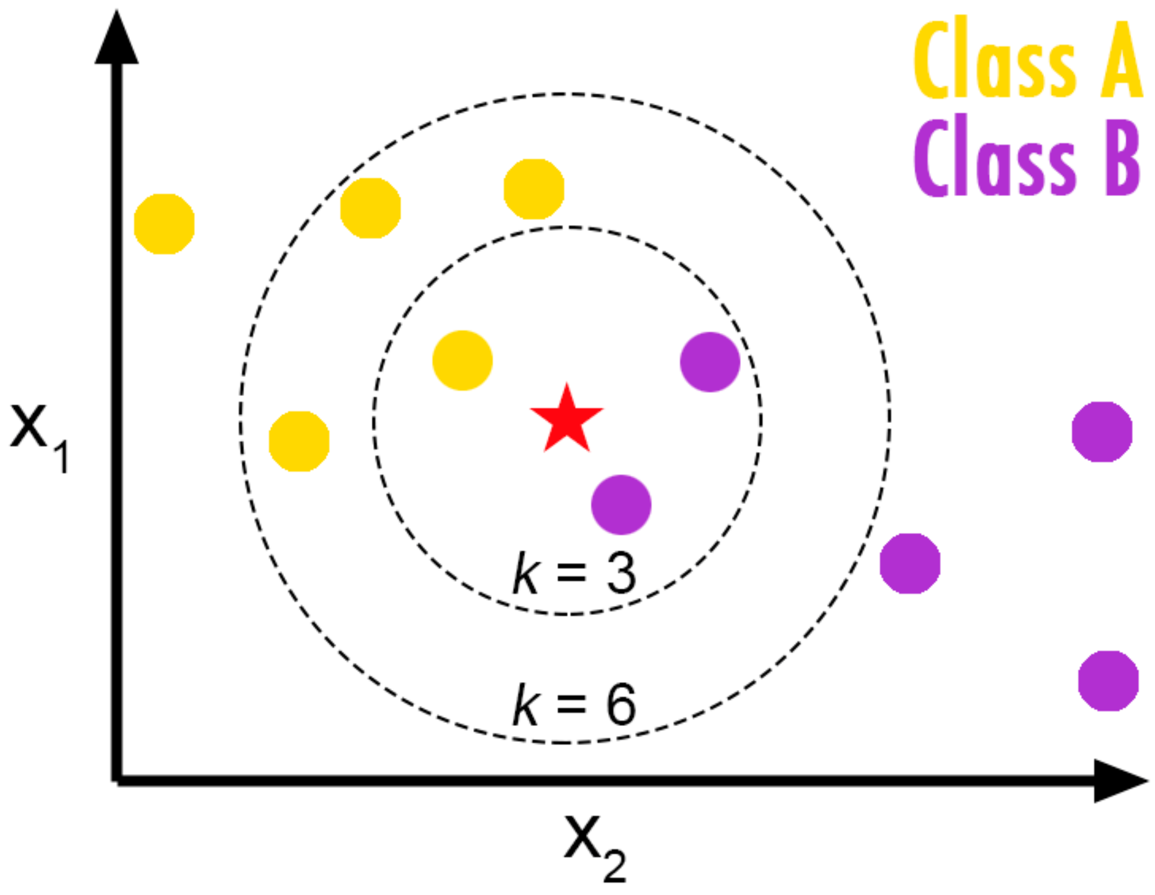
Normalization rescales the values into a range of [0,1]. This might be useful in some cases where all parameters need to have the same positive scale. However, the outliers from the data set are lost.

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

for example: (25, 40000)
(25-20)/(60-20)=0.125
(40000-18000)/(220000-1800)=0.11

Using the normalized distance on the same training set, when K=1, the unknown case returned a different neighbor N.
With K=3, there are two Default=N and one Default=Y out of three closest neighbors. The prediction for the unknown case is Default=N.

Choice of K



A small value of K :

- The noises have a higher influence on the result
- Cause overfitting

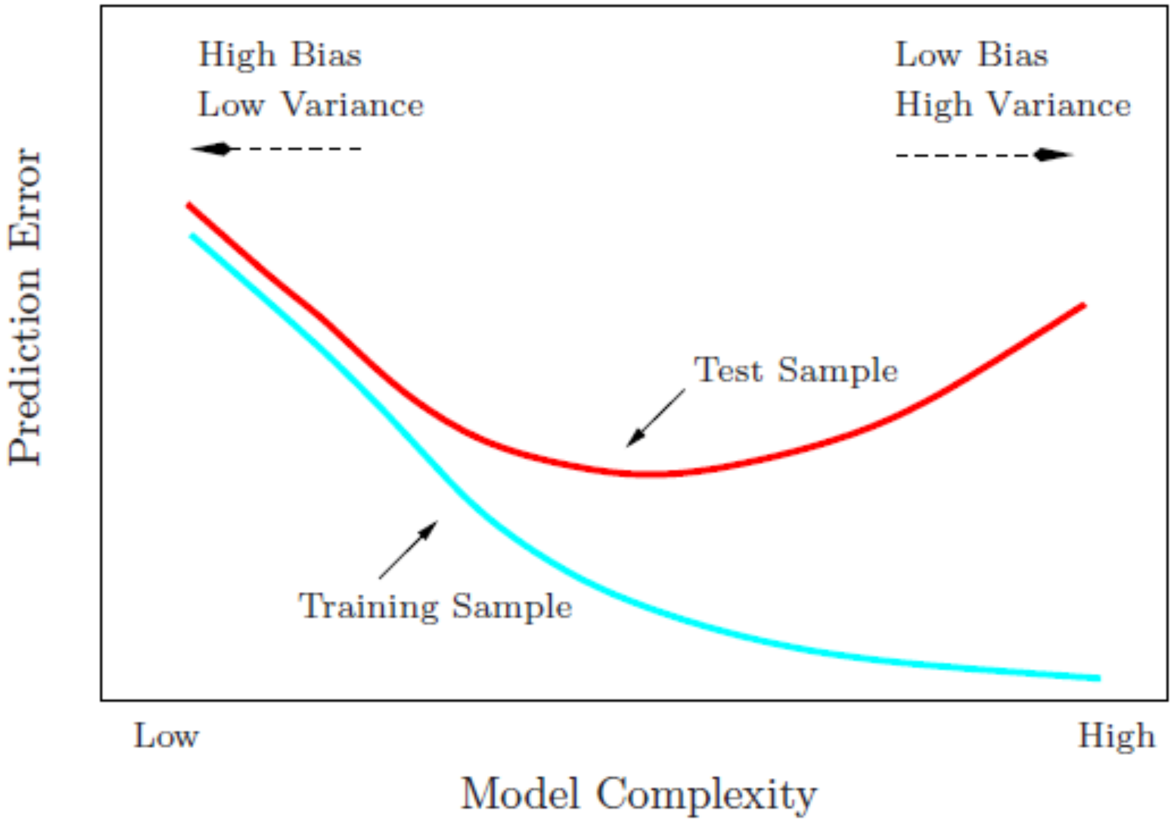
A large value of K :

- Computationally expensive
- Defeat the basic idea behind KNN

A simple approach to select k is k = n^(1/2)

Training/Test Data Set

- training set: a set of data used to discover potentially predictive relationships
- training error: the error of applying the statistical learning method to the observations used in its training
- test set: a set of data used to assess the strength and utility of a predictive relationship
- test error: the average error that results from using a statistical learning method to predict the response on a new observation, one that was not used in training the method



Select Tuning Parameter: k

K-folder cross validation: estimates can be used to select best model, and to give an idea of the test error of the final chosen model. (5 or 10-folder)

- Randomly divide the data into K equal-sized parts.
- Leave out part k, put it to the side
- fit the model to the other K – 1 parts (combined) as the training set
- Use part k as the test set to estimate the prediction error
- Repeat this process K times, once each for the different splits of the data
- The cross-validation error can be obtained by computing the weighted average of K folders

Regression mean squared error: $CV_K = \sum_{k=1}^{K} \frac{n_k}{N} MSE_k$

- Select the k in which the cross-validation error is smallest

KNN advantages and disadvantages

Advantages of K-nearest neighbors algorithm:

- KNN is simple to implement.
- KNN executes quickly for small training data sets.
- Don’t need any prior knowledge about the structure of data in the training set.

Limitation to K-nearest neighbors algorithm:

- When the training set is large, it may take a lot of space.
- For every test data, the distance should be computed between test data and all the training data. Thus a lot of time may be needed for the testing.