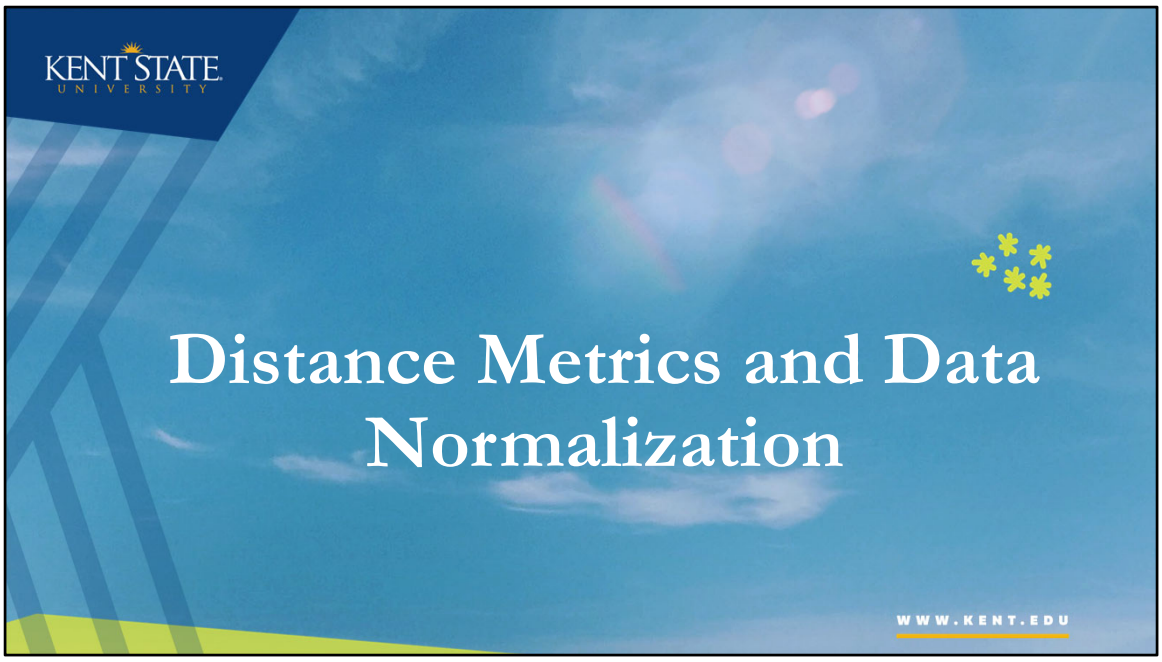
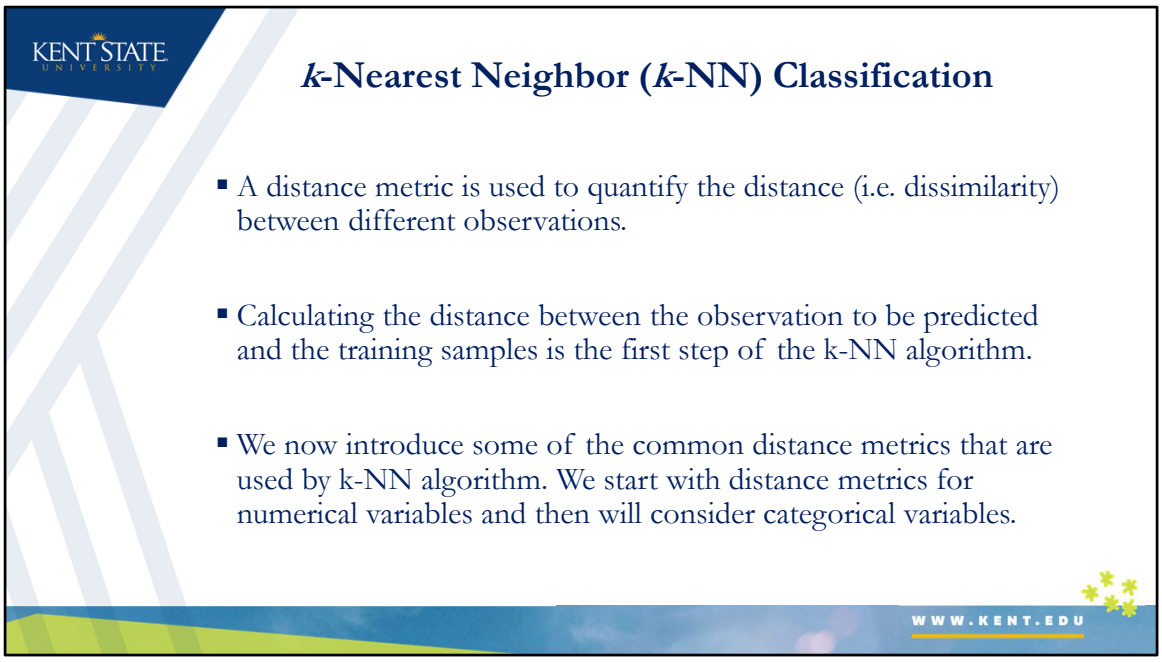


In this module, we expand our discussion on distance measures and the concept of normalization. As we saw in k-NN, we use distance as a proxy for similarity. As distance calculations may be affected by the units of the measurement, it is also important that we normalize that data so that the measurements can be compared. This is no different from comparing cost of living indices across countries with different currencies where we standardize or normalize the indices to a single currently.



Let us now take a look at some distance metrics both for categorical variables, and for quantitative variables.

KENT STATE UNIVERSITY

Common Distance Metrics

- Suppose *p* and *q* are two n-dimensional vectors. The following distance metrics are commonly used in *k*-NN algorithm:
  - City Block or Manhattan distance: $d(p,q) = \sum_{i=1}^{n} |p_i - q_i|$
  - Euclidian distance: $d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$
  - Minkowski distance: $d(p,q) = (\sum_{i=1}^{n} |p_i - q_i|^c)^{1/c}$
- As you can see, for c = 1 , c = 2, the Minkowski metric becomes equal to the Manhattan and Euclidean metrics respectively.

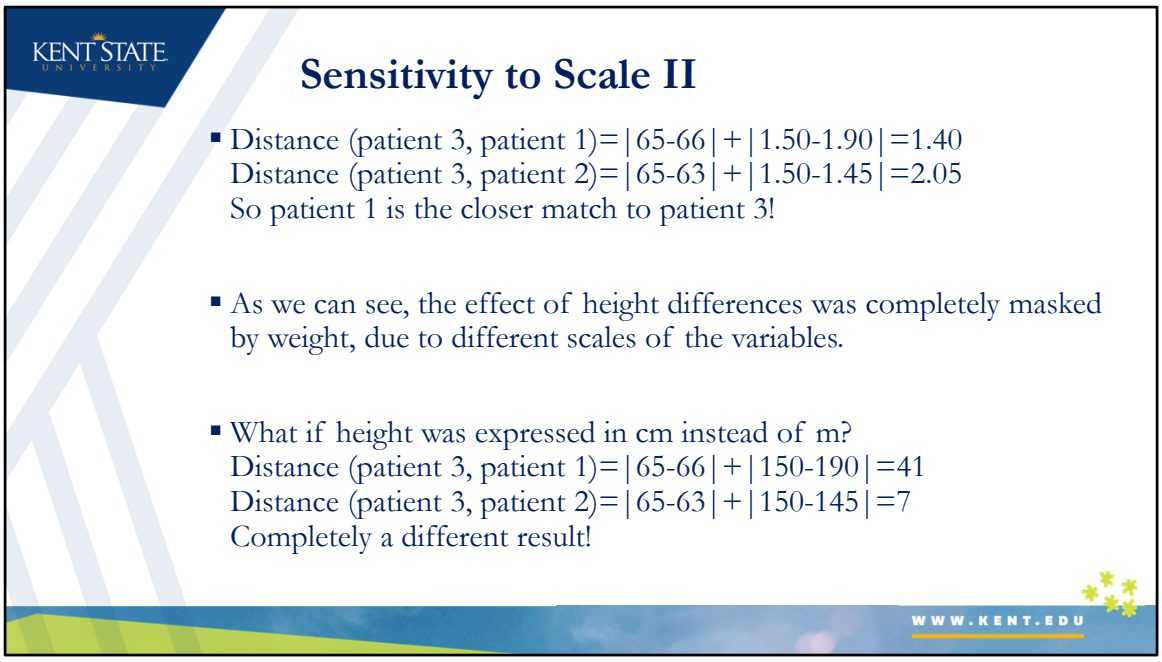
WWW.KENT.EDU

We noted in the previous module that the Euclidian distance was commonly used with k-NN. Here are two other measures which we use. Note that all three measures are slight variations of each other. Again, any measure we choose should have low computational load as prediction is done as needed, and distances are calculated for each data value against the whole training set.

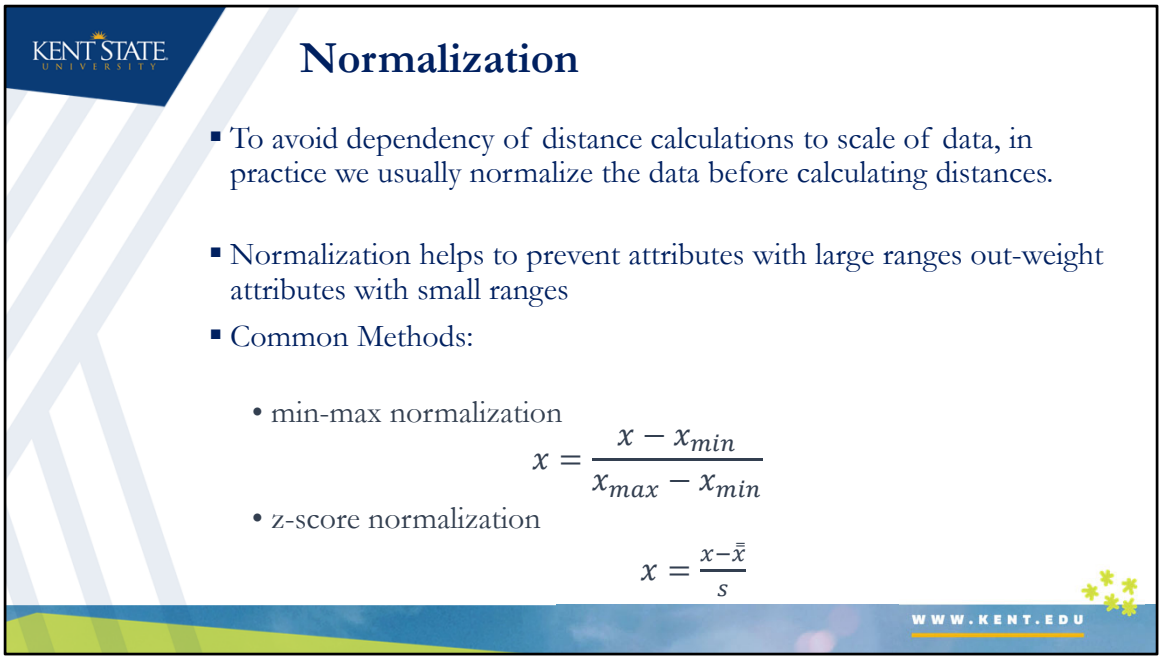
Sensitivity to Scale I

- Calculation of distance is very sensitive to the scale of individual attributes.
- Suppose we want to calculate the distance between the records of patients using only two attributes (variables) weight and height:
  - Patient 1: weight = 66kg height=1.90m
    Patient 2: weight = 63kg height=1.45m
    Patient 3: weight = 65kg height=1.50m
- Which of the patients 1 and 2 is the closest match (i.e. nearest neighbor) to patient 3? Assume City Block is used as the distance metric.

If you remember your beginning statistics, especially the calculation of variances, you will realize that distance measures, like variances, are affected by the scale or units of measurement.



As you can see the result is affected by the units of measurement. One approach to solve this problem is the concept of standardization or nomalization where we convert all values of a variable into a unitless range of numbers. It is then possible to directly compare the effect or magnitude differences between variable values.



While both the min-max approach and z-score normalization approaches are commonly used, the latter is a more robust measures as the standard deviation In the denominator uses all values in its calculation. Further, once the data has been standardized according to the z-score formula, the data will have a mean of 0 and a variance of 1. This allows direct comparison of values across variables. For example, if you receive a Z-score of 1.4 in your exam, and another student in a totally different class receives a Z-score of 1.2, then we can state that you did better in your class compared to what the other student did in his or her class.

KENT STATE UNIVERSITY

Normalization Example

Age	min-max (0-1)	z-score
44	0.421	0.450
35	0.184	-0.450
34	0.158	-0.550
34	0.158	-0.550
39	0.289	-0.050
41	0.342	0.150
42	0.368	0.250
31	0.079	-0.849
28	0.000	-1.149
30	0.053	-0.949
38	0.263	-0.150
36	0.211	-0.350
42	0.368	0.250
35	0.184	-0.450
33	0.132	-0.649
45	0.447	0.550
34	0.158	-0.550
65	0.974	2.548
66	1.000	2.648
38	0.263	-0.150

Note that all negative values of z-score indicate scores less than the average. So, age of 35 is less than the average age for that group. This information cannot be determined if we use the min-max approach.

KENT STATE UNIVERSITY

Data Transformation: Normalization

- The preProcess ( ) function that is in the 'caret' package is a powerful method which has implemented a number of data processing and transformation methods.
- The function implements min-max normalization using "range" as the method or z-score scaling when using "center" and "scale" as input method parameters.
- The function, however, creates a model which needs to be applied to the data using the predict ( ) function.

WWW.KENT.EDU

Let us now implement normalization in R using the caret library.

Example: Data Normalization

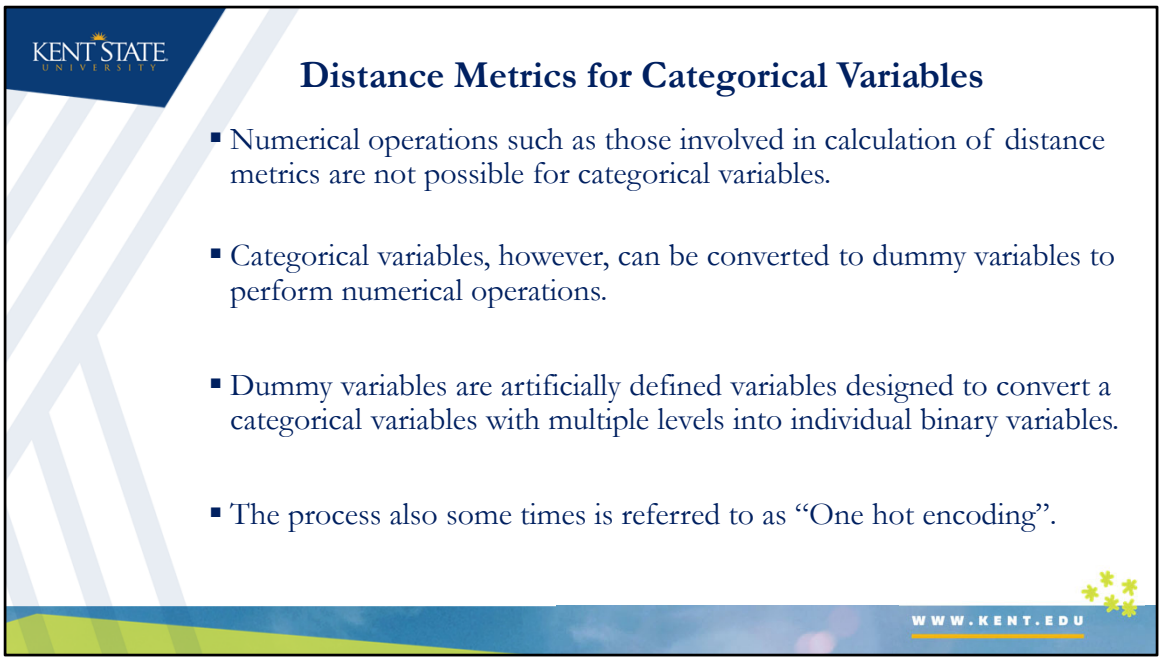
```
library(caret)
summary(cars)
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  

Model_range_normalized<-preProcess(cars,method = "range
")
cars_normalized=predict(Model_range_normalized,cars)
summary(cars_normalized)
##      speed             dist        
##  Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.3810   1st Qu.:0.2034  
##  Median :0.5238   Median :0.2881  
##  Mean   :0.5429   Mean   :0.3473  
##  3rd Qu.:0.7143   3rd Qu.:0.4576  
##  Max.   :1.0000   Max.   :1.0000  
```

```
Model_z_normalized<-preProcess(cars,method = c("center","s
cale"))
summary(predict(Model_z_normalized,cars))

##      speed               dist        
##  Min.   :-2.15597   Min.   :-1.5903  
##  1st Qu.:-0.64301   1st Qu.:-0.6589  
##  Median :-0.07565   Median :-0.2709  
##  Mean   : 0.00000   Mean   : 0.0000  
##  3rd Qu.: 0.68083   3rd Qu.: 0.5053  
##  Max.   : 1.81555   Max.   : 2.9888  
```


Code found here can be found in the github account for the class. Note that after normalization, the mean and standard deviation will become 0 and 1 respectively on the data that was used to normalize, as long as we use the z-normalization approach. This is not true if we use the range approach as seen above.



How do you calculate distances when dealing with categorical variables? One approach is to code them as numeric. This is a common technique that is frequently used when we discuss Deep Learning. For now, the concept is to code these categorical variables as Dummy Variables that take on values 0 and 1. An example follows.

Dummy Variables: Example

Original Data →

```
    Price Mileage  Type
214 19981   24323 sedan
299 21757    1853 sedan
460 15047   12305 sedan
728 15327    4318 sedan
162 20628   20770 sedan
718 16714   26328 sedan
```

```
levels(carSubset$Type)
[1] "convertible" "coupe"       "hatchback"   "sedan"       "wagon"
```

Data with Dummy Variable →

	Mileage	convertible	coupe	hatchback	sedan	wagon
214	24323	0	0	0	1	0
299	1853	0	0	0	1	0
460	12305	0	0	0	1	0
728	4318	0	0	0	1	0
162	20770	0	0	0	1	0
718	26328	0	0	0	1	0

Here note that we have created the same number of dummy variables as there are categories for Car Types. The dummy variables take on a value 1 if a particular car belongs to that subtype. 0 otherwise.

KENT STATE UNIVERSITY

Dummy Variables Implementation

```
library(caret) #install if you need

library(ISLR)

dummy_model<-dummyVars(~race, data=Wage)
head(predict(dummy_model,Wage))
```

```
##        race.1. White race.2. Black race.3. Asian race.4. Other
## 231655             1             0             0             0
## 86582              1             0             0             0
## 161300             1             0             0             0
## 155159             0             0             1             0
## 11443              1             0             0             0
## 376662             1             0             0             0
```


Dummy variables can be easily accommodated as part of the caret package. Remember to load the caret and ISLR packages before using the dummyVars function.

This concludes our module on distance calculation. Make sure to examine the code on the github account.