CS167: Machine Learning

kNN using **scikit-learn** library activity (continued)
Data Normalization

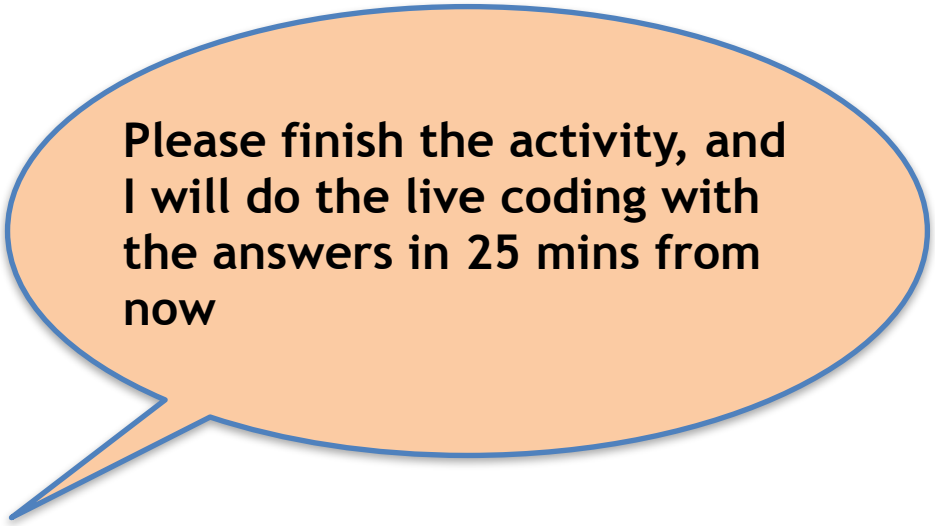
Wednesday, February 18th, 2026

Announcements

- Notebook #2: kNN and Normalization
  - to submit, download the `.ipynb` file from Colab
  - directly upload to Blackboard
  - due next Monday 02/23 by 11:59pm

kNN implementation using scikit-learn library

- I did the demo last time; however, we didn't get a chance to finish the experiments with the kNN implementation using scikit-learn.



Today's Agenda

- Data normalization — *z-score*

- Data normalization coding

Normalization

- Normalizing data:
  - rescale attribute values so they're about the same
  - adjusting values measured on different scales to a common scale

A Simple Normalization

- One simple method of normalizing data is to replace each value with a proportion relative to the max value.

- For example, the oldest person on the Titanic was 80, so:

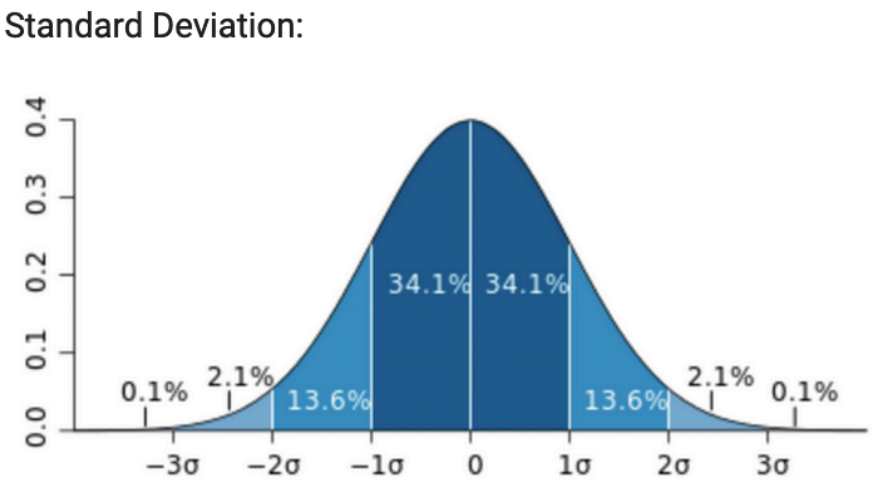
age	replaced by
80	80/80 = 1
50	50/80 = 0.625
48	48/80 = 0.6
25	25/80 = 0.3125
4	4/80 = 0.05

Quick question:
What is the range of these newly normalized values?

Z-Score: Another Normalization Method

- **Idea**: rather than normalize to proportion of max, normalize based on how many standard deviations they are away from the mean

- **Standard Deviation**: usually represented as σ(sigma), a kind of 'average' distance from the average value
  - a low standard deviation indicates that the values tend to be close to the mean
  - a high standard deviation indicates that the values are spread out over a wider range



A Gaussian distribution

Standard Deviation Calculation

- **Standard Deviation:** usually represented as σ(sigma), a kind of 'average' distance from the average value
  - Find the mean, represented as μ:mu
  - Then, for each number, subtract the mean and square the result
  - Then, find the mean of those squared differences
  - Take the square root of that and we are done

- Let μ be the mean, then standard deviation of x_1 , x_2,…, x_N is:

$$\sigma = \sqrt{\frac{(x_1-\mu)^2+(x_2-\mu)^2+...+(x_N-\mu)^2}{N}}$$

Computing the Z-Score

- After computing the corrected sample standard deviation, to normalize, replace each value x_i with it's ***z-score*** based on the mean (μ) and standard deviation (σ) of it's column.

$$Z - score : \frac{x_i - \mu}{\sigma}$$

z-score also known as **standardization**

Computing the Z-Score

- For example: On the Titanic:
  - sex mean(0:male, 1:female): 0.35
  - sex standard deviation: 0.48
  - age mean: 29.7
  - age standard deviation: 13

$$Z - score : \frac{x_i - \mu}{\sigma}$$

	sex	age
example 1	1	50
example 2	0	48

	sex	age
example 1	1	50
example 3	1	25

Z-Score for male: $(0 - 0.35)/0.48 \approx -0.73$
Z-Score for female: $(1 - 0.35)/0.48 \approx 1.35$

Z-Score for age 50: $(50 - 29.7)/13 \approx 1.56$
Z-Score for age 48: $(48 - 29.7)/13 \approx 1.41$
Z-Score for age 25: $(25 - 29.7)/13 \approx -0.36$

Group Exercise

- Group Exercise: Take the next few minutes to talk to your neighbors and work on the in-class activity.
    - Link to Google Form: Day08 data normalization

Going back to our running example

- For example: On the Titanic:
  - sex mean(0:male, 1:female): 0.35
  - sex standard deviation: 0.48
  - age mean: 29.7
  - age standard deviation: 13

$$Z - score : \frac{x_i - \mu}{\sigma}$$

	sex	age
example 1	1	50
example 2	0	48

	sex	age
example 1	1	50
example 3	1	25

Z-Score for male: $(0 - 0.35)/0.48 \approx -0.73$
Z-Score for female: $(1 - 0.35)/0.48 \approx 1.35$

Z-Score for age 50: $(50 - 29.7)/13 \approx 1.56$
Z-Score for age 48: $(48 - 29.7)/13 \approx 1.41$
Z-Score for age 25: $(25 - 29.7)/13 \approx -0.36$

Distance Computation Before Normalization

	sex	age
example 1	1	50
example 2	0	48

distance: $\sqrt{(1-0)^2 + (50-48)^2} \approx 2.24$

	sex	age
example 1	1	50
example 3	1	25

distance: $\sqrt{(1-1)^2 + (50-25)^2} = 25$

Without z-score normalization, the distances are skewed

Distance Computation After Normalization

	sex	age
example 1	1.35	1.56
example 2	-0.73	1.41

distance:

$$\sqrt{(1.35 - -0.73)^2 + (1.56 - 1.41)^2} \approx 2.09$$

	sex	age
example 1	1.35	1.56
example 3	1.35	-0.36

distance:

$$\sqrt{(1.35 - 1.35)^2 + (1.56 - -0.36)^2} = 1.92$$

With z-score normalization, the distances are more comparable

Today's Agenda

- Data normalization — *z-score*

- Data normalization coding

Computing the Z-Score on Titanic

- Called on a data frame, will replace values given in `to_replace` with `value`. Let's use this to make the `sex` column of the dataset numeric.

```
titanic['sex'] = titanic['sex'].replace(to_replace='female', value=1)
titanic['sex'] = titanic['sex'].replace(to_replace='male', value=0)
titanic.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	0	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	1	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	1	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	1	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	0	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

Computing the Z-Score on Titanic

- Now that we have the data as 1s and 0s, let's calculate the mean and standard deviation

```
s_mean = titanic.sex.mean()
s_std = titanic.sex.std()

#replace column with each entry's z-score
titanic.sex = (titanic.sex - s_mean)/s_std
titanic.head()
```

$$Z - score : \frac{x_i - \mu}{\sigma}$$

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	-0.734928	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	1.359146	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	1.359146	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	1.359146	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	-0.734928	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

Group Programming Exercise #1

- Normalize each of the predictor columns in the iris dataset
  - Note: you need a way to transform the new reading that you will make the prediction on so that the new one and the training data will all be on the same scale. How can you do that?