DATA 8

Fall 2020

Tutoring Section 15

Classification and Final Review

Slides by Kevin Miao

Logistics

- A u-GSI might be joining our section at any point to evaluate me
- Project Parties (Zoom links on Piazza):
  - **Tues 12/1 7-8pm**
  - **Wed 12/2 8-9pm**
- On my website, you will find a list of exam questions categorized per topic so you can practice
  - Remember,
    - **Study smart, not hard!**

Today

- Classification, Decision Boundaries and the ML Paradigm

- [Review] **Pick one:**
    - Hypothesis Testing/Inference
    - A/B Testing
    - Regression Inference

Worksheet

Link: **https://tinyurl.com/d8tutweek15**

Classification

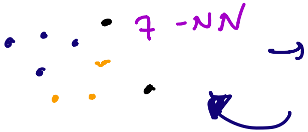
- **Two main problems in ML**
    - I want to predict a number given inputs
        - Linear Regression (Last couple of weeks)
    - I want to predict a category given inputs
        - Classification
            - Example 1: Can we predict whether a person has cancer based on their lifestyle, lab results?
            - Example 2: Can we predict whether a person will end up buying a certain product on a webshop based on their behavior?

Classification

- To create a **classifier**, we need:
  - **Data/Observations:** Data where we both have the classes and features
  - **Attributes:** Characteristics of an individual. Categorical variables will be denoted with 1s and 0s.
  - **Population**: A larger group of individuals which we try to predict
- **ML Paradigm:**
  - Training Data: Practice Exam with access to solutions
  - Testing Data: Final Exam where we only have access to the solutions after we apply our classifier to it.
  - Typical Split: **80/20** for **Training/Test**

Q1-2

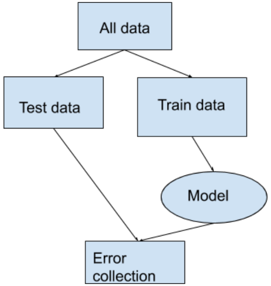
1.1 Sue and Avery are deciding on which kind of nearest neighbors classifier they want to use. Avery says that using a larger number of neighbors will *always* result in more accurate predictions. Sue disagrees. Who is right, and why?
(Hint: Think about what happens when we have a data set with *n* points and we use an *n*-nearest neighbors classifier.)



no, sue is right - A larger n will not always result in a higher accuracy!

2.1 In order to make the model as accurate as possible, should we use all of our data to train the model?

yes, we do split it! that way we can see how accurate it is!



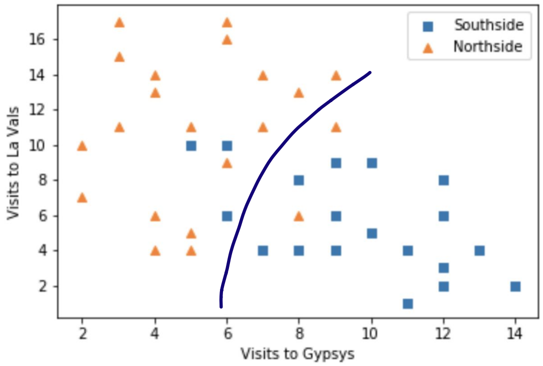
2.2 How should we split our data into training and testing sets? Why?

We want to sample 80% / 20% into the training / test sets for it to be representative.

Q3

A student is trying to build a classifier that classifies Berkeley students as residents of Northside or Southside. The student has a random sample of Berkeley students all of whom live on Northside or Southside. For each student she records whether the student lives on Northside or Southside, the number of times the student went to La Val's (on Northside) in the last 6 months, and the number of times the student went to Gypsy's (on Southside) in the last 6 months.

3.1 Draw a decision boundary for a 5 nearest neighbor classifier on the scatter plot below.



Hypothesis Testing

Roulette is a casino game in which a ball falls into one pocket in a spinning wheel, and players bet on the color of the pocket in which the ball falls. If the player correctly picks the color, they win. 18 of the 38 pockets in a roulette wheel are red. You play 30 games of roulette, bet on red every time, and win 20 of those games. You become suspicious about the fairness of this roulette game.

Hypothesis Testing

1.1 State a null and alternative hypothesis to see whether the roulette game is biased towards red.

Null Hypothesis:

Alternative Hypothesis:

1.2 With your alternative hypothesis in mind, choose a test statistic and calculate its observed value. Your test statistic should be large for data favoring the alternative hypothesis.

Test Statistic:

Observed Value:

Hypothesis Testing

1.3 Complete the function `prop_wins_in_30_games()`, which takes in no arguments and simulates playing the game 30 times and returns the proportion of wins when you guess red every time.

```
def prop_wins_in_30_games():
     proportions = ____________________________________________
     thirty_games = ___________________________________________
     prop_wins = ______________________________________________
     return ___________________________________________________
```

1.4 Complete the code below to simulate an empirical distribution of the test statistic using 10000 iterations, storing the statistics in an array called `simulated_statistics`.

```
simulated_statistics = make_array()
for i in ____________________:
    prop_win = ______________________________________________
    simulated_statistics = ___________________________________
```

Hypothesis Testing

1.5 Write a line of code to calculate the p-value.

1.6 Suppose you find a p-value of 0.0103. What do you conclude about the null hypothesis, at a p-value cutoff of 5%?

A/B Testing

We are examining the weights of a population of cats and dogs. You are given a random sample from this population, stored in the table `pets`, which has two columns. The first column `‘Animal’` contains a string, either `‘Cat’` or `‘Dog’`. The second column `‘Weight’` contains the weights of each of the animals in pounds as floats. You notice that the average weight of dogs in your sample is 2 pounds heavier than the average weight of cats in your sample.

A/B Testing

2.1 State a null and alternative hypothesis to see if dogs weigh more than cats on average in the population.

Null Hypothesis:

Alternative Hypothesis:

2.2 With your alternative hypothesis in mind, choose a test statistic and calculate its observed value. Your test statistic should be large for data favoring the alternative hypothesis.

Test Statistic:

Observed Value:

A/B Testing

2.3 Complete the function `one_shuffled_table_stat()` which takes in no arguments and returns one value of the test statistic.

```
def one_shuffed_table_stat():
    shuf_table = ____________________________________________
    shuf_weights = __________________________________________
    shuf_tbl_with_weights = _________________________________
    grouped_with_mean = _____________________________________
    dog_mean = ______________________________________________
    cat_mean = ______________________________________________
```

2.4 Complete the below code to simulate an empirical distribution of 5000 test statistics under the assumptions of the null hypothesis.

```
diffs = make_array()
for i in np.arange(5000):
    diff = ________________________________________________
    diffs = _______________________________________________
```

A/B Testing

2.5 Write a line of code to calculate your p-value.

```
p_value = ____________________________________________________
```

2.6 Suppose you find a p-value of 0.13. What do you conclude, at a p-value cutoff of 5%?

Regression

Assume: students is representative of the population

You take a sample of Data 8 students and ask them about their daily consumption of coffee and their midterm exam scores. Assume you are given a table `students` with the columns `cups` and `score`. The column `cups` contains the daily consumption of coffee and `score` contains the midterm exam score for each student from the sample. You perform linear regression and find a slope of 0.13 points per cup of coffee. Stephanie is considering buying coffee for her discussion to increase their scores, but is not sure if this will work.

Regression

3.1 State a null and alternative hypothesis to see whether this slope was due to randomness in your sample.

Null Hypothesis: The midterm scores unrelated to coffee intake. Slope is ∅.
Alternative Hypothesis: " related " . Slope is not ∅.

3.2 Write a function `slope` that takes in a table, `tbl`, and returns the slope of the least-squares line using the first column to predict values of the second column.

```
def slope(tbl):
    x = tbl.column('cups')
    y = tbl.column('score')
    x_su = (x - np.average(x)) / np.std(x)
    y_su = (y - np.average(y)) / np.std(y)
    r = np.average(x_su * y_su)
    slope = r * np.std(y)/np.std(x)

    return slope
```

Regression

3.3 Complete the code to generate 5000 bootstrap resample slopes and then calculate a 95% confidence interval for the slope. Assume the function `slope(tbl)` has been implemented correctly.

```
slopes = make_array()
for i in np.arange(5000):
    resample_slope = slope( students.sample())
    slopes = np.append(slopes, resample_slope)

left_end = percentile( 2.5 , slopes)
right_end = percentile( 97.5 , slopes)
interval = make_array(left_end, right_end)
```

3.4 Suppose you find the confidence interval [0.02, 0.24]. What do you conclude about your hypotheses at a p-value cutoff of 5%? What about at a p-value cutoff of 10%? What about at a p-value cutoff of 1%?

95% CI → Reject Null → 0 not in [0.02, 0.24]

90% CI → narrower

99% CI → wider → need to calculate percentiles again!

Regression

3.5 Your friend who hasn't taken Data 8 looks at your result and asks you what the confidence interval means. Which one of the following is a correct response?

~~a)~~ For 95% of students, there is a relationship between coffee consumption and midterm score.

not true; IDK

~~b)~~ There is a 95% probability that the true slope is between 0.02 and 0.24.

[✓ c)] There is a 95% probability that our sampling process (and the code above) produces an interval that contains the true slope.

Thank You

- **This was our last tutoring section. You did it!** 🎉
- **Finals:**
  - I believe in you! Good Luck. An exam guide is on my website
- **After Data 8:**
  - Become involved with course staff!
    - Lab Assistant, Tutor, TA
  - Take cool ML/Stats classes
- It was an honor to meet you all! Feel free to stay in touch with me/each other (LinkedIn/Social Media) and feel free to always email me if you have any questions.