TUTORIAL 7 HELP SHEET

EXERCISE 7A: COMPUTING K-NEAREST NEIGHBORS MANUALLY

PART A

HINT 1: Check Lecture 7 Slide 8

HINT 2: The formula explained (if you are still confused)

$$D_{\text{Euclidean}}(\underbrace{x_i}_{\text{obs i vector}}, \underbrace{x_j}_{\text{obs j vector}}) = \sqrt{\sum_{s=1}^{p} (\underbrace{\overbrace{x_{is}}^{\text{observation i Predictor s value}} - \underbrace{x_{js}}_{\text{observation j Predictor s value}}}_{\text{think } age_i - age_j})^2}$$

think $x_k = (age_k, Income_k)$

PART B

HINT 3: Check Lecture 7 Slide 17

HINT 4: If you are still not sure, its basically $N^{\#\text{ neighbors}}_{\text{observation}} = \{\text{set of neighbouring obs}\}$

PART C

HINT 5: Check Lecture 7 Slide 9. The formula explained would basically be the same as hint 2 but with an absolute value.

PART D

HINT 6: Think about the differences between the distance measures that exist even if they produce similar results.

PART E

HINT 7: Check lecture 7 slide 40.

HINT 8: Basically take the average of the k nearest neighbours expenditure.

EXERCISE 7B: APPLICATION TO BOSTON HOUSING DATA SET

PART A

HINT 9: Try read_csv() and mutate(across()).

HINT 10: Try initial_split(), training() and testing() (remember to use set.seed()).

PART B

HINT 11: Check Lecture 7 Slide 34. If a question does not specify a distance metric, use the default.

HINT 12: Try kknn(formula, train, test, k)

PART C

HINT 13: Check Lecture 7 Slide 36 (Specifically the "AUC=" bit).

HINT 14: Try rmse_vec(true, predicted) from the Yardstick package.

PART D

HINT 15: Check Lecture 7 Slide 36 (Combine your code from part C with

HINT 16: Think about trying to Combine your code from part C with the mapping function on that slide.

HINT 17: Try writing out a couple values of K with repetitive code and then think of a way to change it into a for loop.