# Final Exam

## Data Science for Studying Language & the Mind

### Instructions

The exam is worth **138 points**. You have **2 hours** to complete the exam.

- The exam is closed book/note/computer/phone except for the provided reference sheets
- If you need to use the restroom, leave your exam and phone with the TAs
- If you finish early, you may turn in your exam and leave early

### (5 points) Preliminary questions

Please complete these questions *before* the exam begins.

(a) **(1 point)** What is your full name?

(b) **(1 point)** What is your penn ID number?

(c) **(1 point)** What is your lab section TA's name?

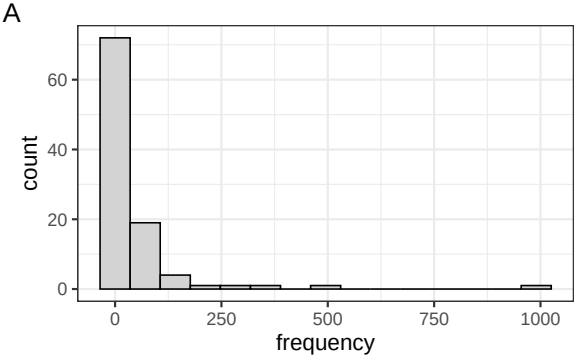
(d) **(1 point)** What is today's date?

(e) **(1 point)** Sign your name to agree you will not discuss this exam with anyone until December 19th at 2:00pm EST.

## 1. (16 points) Sampling distribution

In natural language, words follow a Zipfian distribution, where a few words are highly frequent and many words are rare. Figure A shows a histogram of this distribution.

A

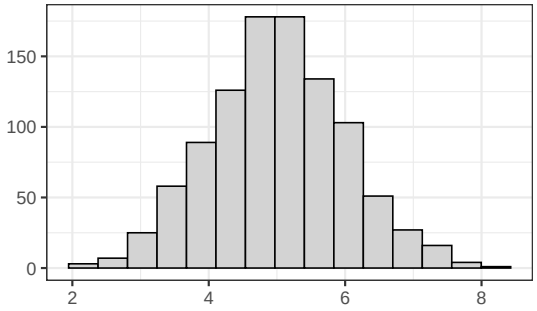



(a) **(2 points)** Which descriptive statistic should we use to summarize the central tendency of these data?

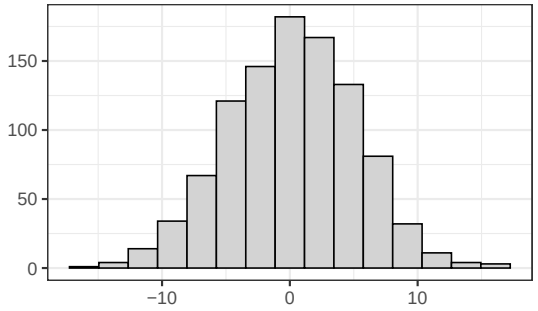
(b) **(2 points)** Which descriptive statistic should we use to summarize the spread of these data?

(c) **(2 points)** Suppose we run the code `rnorm(1000, mean=5, sd=1)`. Which figure below shows the histogram of the resulting data.

B




C




☐ Figure B
☐ Figure C
☐ Not enough information to determine this

(d) **(2 points)** True or false, the probability density function that generated the data in figures B *and* C is given by the equation: $p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$

- ☐ True
- ☐ False

(e) **(6 points)** Given the following code, explain what happens in each step:

```
library(infer)
boot_dist <- data %>%
    specify(response =  frequency) %>%
    generate(reps = 1000, type = "bootstrap") %>%
    calculate(stat = "mean")
```

(i) Explain `specify(response = frequency)`

(ii) Explain `generate(reps = 1000, type = "bootstrap")`

(iii) Explain `calculate(stat = "mean")`

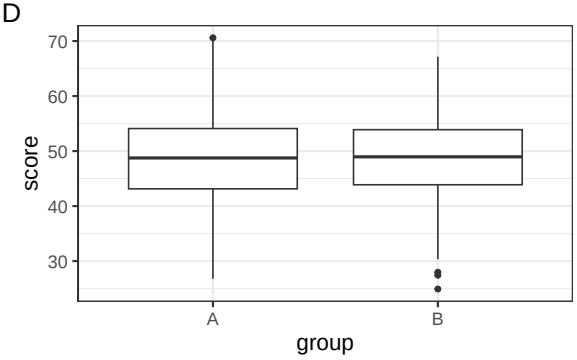
(f) **(2 points)** True or false, bootstrapping the sampling distribution requires that the data are approximately normally distributed.

- ☐ True
- ☐ False

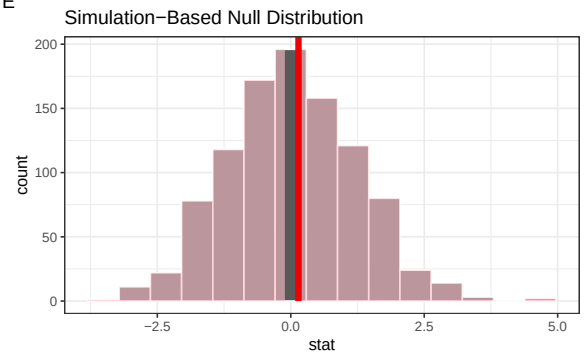
## 2. (12 points) Hypothesis testing

Suppose a teacher wanted to compare test scores of students in two different sections of a class (Group A and Group B). They visualized the difference in mean test scores betwen groups with a boxplot (figure D). And then used `infer` to construct the null distribution for the difference in means. They used `visualize()` and `shade_p_value()` to visualize the data (figure E).

D




E




(a) **(3 points)**What is the null hypothesis?

(b) **(2 points)**If the null hypothesis is true, how likely is our observed pattern of results?

- ☐ $p = 0.908$
- ☐ $p = 0.012$
- ☐ $p = 192$
- ☐ $p < 0.05$
- ☐ Not enough information to determine this

(c) **(2 points)**True or false, we should reject the null hypothesis.

- ☐ True
- ☐ False

(d) **(3 points)** Explain why you chose True or False for question (c) above.

(e) **(2 points)** Why do we pose a null hypothesis? Choose one.

- ☐ It is the hypothesis most likely to be true.
- ☐ It allows us to generate predictions based on prior beliefs.
- ☐ It is the hypothesis for which we can simulate data.
- ☐ It ensures that the alternative hypothesis is proven false.

## 3. (24 points) Modeling true or false

(a) **(2 points)** The goal of a regression model is to classify observations into distinct categories.

☐ True
☐ False

(b) **(2 points)** Model specification involves finding the best fitting free parameters.

☐ True
☐ False

(c) **(2 points)** The equation $y = \beta_0 \cdot 1 + \beta_1 \cdot x + \beta_2 \cdot x_2$ expresses $y$ as a weighted sum of inputs.

☐ True
☐ False

(d) **(2 points)** Regression and classification are both considered to be unsupervised learning problems.

☐ True
☐ False

(e) **(2 points)** In gradient descent, we use linear algebra to find the closed form solution to the parameter estimates.

☐ True
☐ False

(f) **(2 points)** The ordinary least squares solution could arrive at a local minimum and miss the true global minimum.

☐ True
☐ False

(g) **(2 points)** The highest possible $R^2$ value is 1 (100%).

☐ True
☐ False

(h) **(2 points)** The lowest possible $R^2$ value is 0 (0%).

☐ True
☐ False

(i) **(2 points)** An overfit model performs poorly on the sample, but well when predicting new values.

☐ True
☐ False

(j) **(2 points)** The error bars on our parameter estimates will become smaller as we increase our sample size.

☐ True
☐ False

(k) **(2 points)** Nearest-neighbor regression can be used for classification problems.

☐ True
☐ False

(l) **(2 points)** The output of the logistic function is bounded by -1 and 1.

☐ True
☐ False

## 4. (15 points) Model specification

Suppose we measure the reaction times (in milliseconds) of both native and non-native speakers as they process words of varying frequency in English (measured as occurrences per million words). We store these data in a tibble called `rt_by_speaker`. The first 6 rows of this tibble are printed below for your reference.

```
# A tibble: 6 x 3
  WordFrequency ReactionTime SpeakerType
          <dbl>        <dbl> <chr>
1          38.8         773. Non-native
2          45.4         754. Non-native
3          81.2         711. Non-native
4          51.4         495. Native
5          52.6         851. Non-native
6          84.3         719. Non-native
```

Suppose we specify the following model with `lm`:

```
model <- lm(ReactionTime ~ 1 + WordFrequency * SpeakerType, data = rt_by_speaker)
summary(model)
```

```
Call:
lm(formula = ReactionTime ~ 1 + WordFrequency * SpeakerType,
    data = rt_by_speaker)

Residuals:
     Min       1Q   Median       3Q      Max
-105.959  -28.485   -7.466   28.491  119.740

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                       716.2767    18.9246  37.849  < 2e-16 ***
WordFrequency                      -3.4735     0.3613  -9.614 1.00e-15 ***
SpeakerTypeNon-native             143.5619    29.2683   4.905 3.81e-06 ***
WordFrequency:SpeakerTypeNon-native   1.8589     0.5308   3.502 0.000703 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 47.5 on 96 degrees of freedom
Multiple R-squared:  0.8752,    Adjusted R-squared:  0.8713
F-statistic: 224.5 on 3 and 96 DF,  p-value: < 2.2e-16
```

(a) **(3 points)** For each of the following, circle the option that best describes the type of model we fit.

(i) **(1 point)** Supervised or unsupervised
(ii) **(1 point)** Regression or classification
(iii) **(1 point)** Linear or linearlizable nonlinear

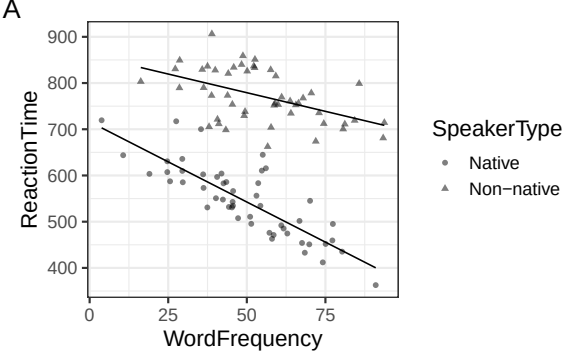
(b) **(3 points)** Write the model's specification as a mathematical expression:

(c) **(3 points)** What is the model's predicted reaction time for a Non-Native speaker with a word frequency of 1? Write your answer as an unsimplified expression (eg. 2 + 3 * 5, not 17):

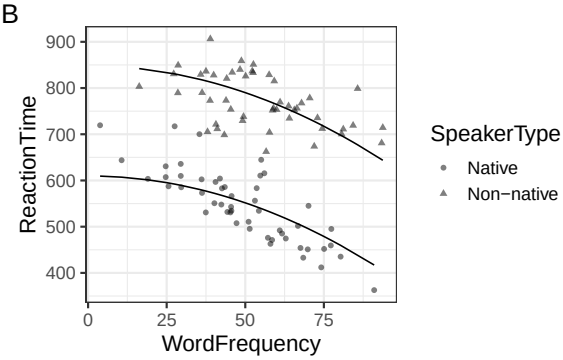
(d) **(3 points)** What is the model's predicted reaction time for a Native speaker with a word frequency of 0? Write your answer as an unsimplified expression (eg. 2 + 3 * 5, not 17):

(e) **(3 points)** Circle the figure that is most likely to be the plot of the model specified to `lm`? Choose one.

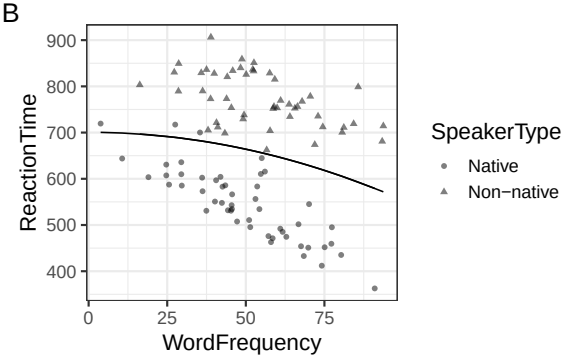
A



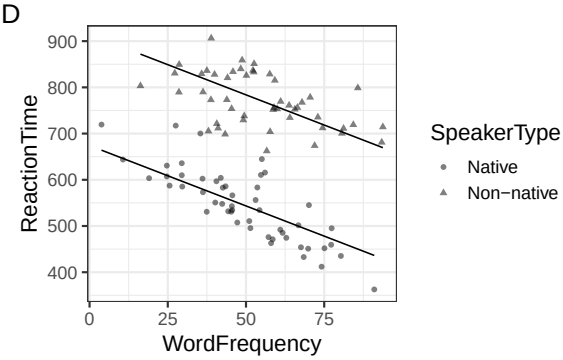

B




B

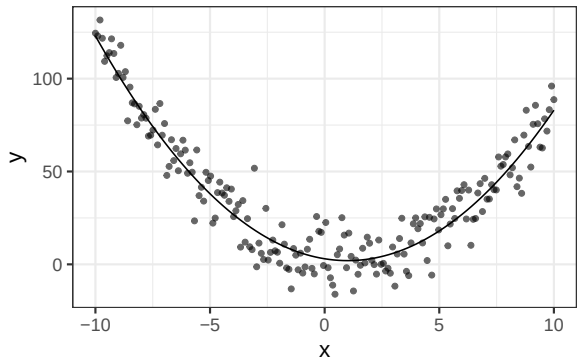



D



## 5. (12 points) Applied model specification

Suppose we encounter the following dataset, plotted here.




We specify and fit these data with `lm`, which gives the following summary:

```
Call:
lm(formula = y ~ x + I(x^2), data = poly_data)

Residuals:
    Min      1Q  Median      3Q     Max
-23.610  -7.336  -0.086   7.439  33.278

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.85268    1.08822   2.621  0.00944 **
x           -1.95787    0.12503 -15.659  < 2e-16 ***
I(x^2)       1.02483    0.02409  42.539  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.26 on 197 degrees of freedom
Multiple R-squared:  0.9125,    Adjusted R-squared:  0.9116
F-statistic:  1027 on 2 and 197 DF,  p-value: < 2.2e-16
```

(a) **(2 points)** What type of polynomial is included in the model specification?

☐ Constant
☐ Linear
☐ Quadratic
☐ Cubic
☐ Quartic

(b) **(3 points)** Write the *fitted model* as a mathematical expression:

(c) **(2 points)** In class we learned about two ways to linearize a nonlinear model. Which option best describes what we have done here?

☐ Expanding the input space by adding new terms
☐ Transforming an existing term

(d) **(2 points)** Given the predicted model (shown with the black line on the figure), what does the model predict for the value of $y$ when $x = -7.5$ (approximate is fine)?

(e) **(3 points)** Suppose we fit the model specification `y ~ 1`. Explain why this would be an overfit or underfit model.

## 6. (15 points) Model fitting

Suppose we fit the polynomial model from section 5 with iterative optimization via `optimg`:

```
optimg(data =poly_data, par = c(1,1,1), fn=SSE, method = "STGD")


$par
[1]  2.843624 -1.957815  1.024964

$value
[1] 20736.36

$counts
[1] 8

$convergence
[1] 0
```

(a) **(2 points)** Which of the following best describes what `optimg` and `lm` return?

- ☐ *nearly* the same parameter estimates
- ☐ identical parameter estimates
- ☐ completely different parameter estimates
- ☐ depends on whether one finds a local minimum
- ☐ not enough information to determine this

(b) **(2 points)** What is the cost function used by `optimg`? Choose one.

- ☐ Sum of squared error
- ☐ STGD
- ☐ standard error
- ☐ standard deviation
- ☐ Not enough information to determine this

(c) **(2 points)** How many steps did our iterative optimization algorithm take?

(d) **(2 points)** What was the value of the cost function for the optimal parameters according to `optimg`?

(e) **(2 points)** What parameters did `optimg` initialize at?

- ☐ `c(0, 0, 0)`
- ☐ `c(1, 1, 1)`
- ☐ `c(2.850417, -1.957891, 1.024866)`
- ☐ Not enough information to determine this

(f) **(2 points)** Which approach does `optimg` use to estimate the free parameters? Choose one.

- ☐ Ordinary least-squares solution (OLS)
- ☐ Gradient descent
- ☐ Either Gradient descent or OLS
- ☐ None of the above

(g) **(3 points)** Given the model specified in the code to `lm`, fill in the missing values for the first 6 rows of the output vector $\mathbf{y}$ and the input matrix $\mathbf{X}$.

$$\begin{bmatrix} \_\_\_\_\_ \\ \_\_\_\_\_ \\ \_\_\_\_\_ \\ \_\_\_\_\_ \\ \_\_\_\_\_ \\ \_\_\_\_\_ \end{bmatrix} = \begin{bmatrix} 1 & \_\_\_\_\_ & 100 \\ 1 & \_\_\_\_\_ & 98.0 \\ 1 & \_\_\_\_\_ & 96.0 \\ 1 & \_\_\_\_\_ & 94.1 \\ 1 & \_\_\_\_\_ & 92.1 \\ 1 & \_\_\_\_\_ & 90.2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix}$$

We've printed the top 6 rows of the `poly_data` tibble for your reference here:

```
# A tibble: 6 x 2
       x     y
   <dbl> <dbl>
1 -10     124.
2  -9.90  123.
3  -9.80  132.
4  -9.70  122.
5  -9.60  109.
6  -9.50  112.
```

## 7. (15 points) Model accuracy

Suppose we want to determine how accurate our polynomial model is from section 5. We return `summary(model)` again here for your reference.

```
Call:
lm(formula = y ~ x + I(x^2), data = poly_data)

Residuals:
    Min      1Q  Median      3Q     Max
-23.610  -7.336  -0.086   7.439  33.278

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.85268    1.08822   2.621  0.00944 **
x           -1.95787    0.12503 -15.659  < 2e-16 ***
I(x^2)       1.02483    0.02409  42.539  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.26 on 197 degrees of freedom
Multiple R-squared:  0.9125,    Adjusted R-squared:  0.9116
F-statistic:  1027 on 2 and 197 DF,  p-value: < 2.2e-16
```

Then we perform cross-validation and return the validation metrics with `collect_metrics()`

```
# A tibble: 2 x 6
  .metric .estimator   mean     n std_err .config
  <chr>   <chr>       <dbl> <int>   <dbl> <chr>
1 rmse    standard   10.3      10  0.424  Preprocessor1_Model1
2 rsq     standard    0.913    10  0.0115 Preprocessor1_Model1
```

(a) **(6 points)** Fill in the blanks in the following code to perform the cross validation on the polynomial model.

```
splits <- vfold_cv(poly_data)
model_spec <- _____a_____ %>%  set_engine(engine = "lm")
our_workflow <- workflow() %>% add_model(model_spec) %>%  add_formula(_____b_____)
fitted_models <- fit_resamples(object = our_workflow, resamples = ____c____)
fitted_models %>% collect_metrics()
```

(i) Blank a (choose one):

- ☐ `logistic_reg()`
- ☐ `linear_reg()`
- ☐ `poly_reg()`
- ☐ `lm()`

(ii) Blank b (fill in the blank)

(iii) Blank c (choose one):

- ☐ `"lm"`
- ☐ `splits`
- ☐ `our_workflow`
- ☐ `model_spec`

(b) **(2 points)** $R^2$ for the population was _____ than $R^2$ for the sample. Choose one.

- ☐ higher
- ☐ lower
- ☐ exactly the same
- ☐ not enough information to determine this

(c) **(2 points)** What kind of cross-validation did we perform? Choose one.

- ☐ k-fold
- ☐ boostrapping
- ☐ leave-one out
- ☐ Not enough information to determine this

(d) **(2 points)** How many splits of our data does our code generate?

(e) **(3 points)** Explain why one would prefer cross-validation over simply relying on the $R^2$ value returned by `lm`.

### 8. (12 points) Model reliability

Suppose we plot and fit two models: `y ~ 1`and `y ~ x + I(x^2)`.

```
Call:
lm(formula = y ~ 1, data = poly_data)

Residuals:
    Min      1Q  Median      3Q     Max
-53.443 -28.731  -7.292  22.653  94.217

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    37.36       2.44   15.31   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 34.51 on 199 degrees of freedom
```

```
Call:
lm(formula = y ~ x + I(x^2), data = poly_data)

Residuals:
    Min      1Q  Median      3Q     Max
-23.610  -7.336  -0.086   7.439  33.278

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.85268    1.08822   2.621  0.00944 **
x           -1.95787    0.12503 -15.659  < 2e-16 ***
I(x^2)       1.02483    0.02409  42.539  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.26 on 197 degrees of freedom
Multiple R-squared:  0.9125,    Adjusted R-squared:  0.9116
F-statistic:  1027 on 2 and 197 DF,  p-value: < 2.2e-16
```

(a) **(2 points)** Which model is more accurate? Choose one.

- ☐ `y ~ 1`
- ☐ `y ~ x + I(x^2)`
- ☐ Both models are equally accurate
- ☐ Not enough information to determine this

(b) **(3 points)** Should we always choose the model with the highest accuracy? Explain why or why not.

(c) **(2 points)** Which value in the model summary quantifies the model's reliability?

- ☐ Multiple R-squared
- ☐ Adjusted R-squared
- ☐ Estimate
- ☐ Std. Error
- ☐ Pr(>|t|)

(d) **(3 points)** Suppose we bootstrap a 68% confidence interval for our parameter estimates for `y ~ x + I(x^2)` model. What would happen if we collected a lot more data? Choose one.

- ☐ It would get smaller (narrower)
- ☐ It would get bigger (wider)
- ☐ It would stay the same
- ☐ An error. We cannot bootstrap polynomial models no matter how much data we have or do not have.

(e) **(2 points)** Rather than adding more participants to our existing sample, suppose we decided to repeat our experiment with an entirely new sample of participants. True or false, fitting the same model to these new data would yield approximately the same parameter estimates?

- ☐ True
- ☐ False

## 9. (12 points) Classification

(a) **(2 points)** Which of the following fits a logistic regression model in R? Choose all that apply.

```
# code A
glm(y ~ x, data = data, family = "binomial")

# code B
data %>%
    specify(y ~ x) %>%
    fit()

# code C
logistic_reg %>%
    set_engine("glm") %>%
    fit(y ~ x, data = data)
```

- ☐ code A
- ☐ code B
- ☐ code C
- ☐ not enough information to determine this

(b) **(2 points)** What is the link function for logistic regression?

- ☐ logistic function
- ☐ polynomial expansion
- ☐ log transformation
- ☐ inverse trasformation
- ☐ linear classifyer

(c) **(2 points)** Which of the following parsnip specifications could specify and fit a classification model? Choose all that apply.

- ☐ `linear_reg() %>% set_engine("lm")`
- ☐ `logistic_reg() %>% set_engine("glm")`
- ☐ `linear_reg() %>% set_engine("classification")`
- ☐ `logistic_reg() %>% set_engine("lm")`

(d) **(2 points)** Which of the following expresses the link function for the `glm` we fit?

- ☐ $f(a) = \frac{1}{1+e^{-a}}$
- ☐ $\sum_{i=i}^{n}(d_i - m_i)^2$
- ☐ $y = \sum_{i=1}^{n} w_i x_i$
- ☐ $R^2 = 100 \times (1 - \frac{SSE_{model}}{SSE_{reference}})$

(e) **(2 points)** What is the difference between regression and classification?

- ☐ Regression predicts a continuous output, classification discrete
- ☐ Classification predicts a continuous output, regression discrete.
- ☐ Regression is a supervised learning problem and classification is unsupervised
- ☐ Regression is linear and classification is nonlinear

(f) **(2 points)** What accuracy metric is best applied to classification models?

- ☐ $R^2$
- ☐ RMSE - root mean squared error
- ☐ Percent correct
- ☐ Adjusted $R^2$