Citadel Assessment

15 Questions - **60 Minutes**

Question 15 / 15 **00:16:13**

1

Which of the following statements about correlation and causation are true?:

> I. A study is designed in which 200 plants are randomly assigned to two groups receiving different levels of water treatment. If growth is statistically significantly greater for one group, then we can say that water level is likely to affect growth of this plant species (i.e. establish probable causation)
> II. Correlation is a measure that describes the magnitude and direction of the relationship between two variables
> III. Pearson's correlation coefficient is more robust to outliers than Spearman's correlation coefficient is because it does not use ranks

A. ○ I only

B. ○ I and II only

C. ○ I and III only

D. ○ II and III only

E. ○ I, II, and III

2

Consider k flash cards such that each card has inscribed on it one of N possible symbols. For any given card, each symbol is equally likely to appear on that card. If $N >= k$, what is the probability that two or more cards have the same symbol?

A. ○ 1

B. ○ $1 - \frac{N!}{(N-k)!N^k}$

C. ○ $\frac{N!}{(N-k)!N^k}$

D. ○ $1 - \frac{(N-k)!}{N^k}$

E. ○ $\frac{(N-k)!}{N^k}$

3

Which of the following quantities can NOT be used to optimize a machine learning algorithm?

A. ○ Cross-entropy

B. ○ Mean squared error

C. ○ Dollar value (e.g. in fraud detection, the money lost for every fraudulent event)

D. ○ Accuracy (i.e. proportion of predictions made which are correct)

E. ○ All of the above are legitimate quantities to optimize for in an objective function

4

In Python, you have a `pandas` dataframe `df` with the columns `car_id`, `distance_1`, `distance_2`, `distance_3`. How would you add a new `avg_distance` column to this dataframe that contains the average distance for each `car_id`?

A. ○ `df['avg_distance'] = df[['distance_1', 'distance_2', 'distance_3']].mean(axis=0)`

B. ○ `df['avg_distance'] = df[['distance_1', 'distance_2', 'distance_3']].mean(axis=1)`

C. ○ `df['avg_distance'] = df[['distance_1', 'distance_2', 'distance_3']].avg(axis=0)`

D. ○ `df['avg_distance'] = df[['distance_1', 'distance_2', 'distance_3']].avg(axis=1)`

E. ○ `df['avg_distance'] = df.sum(['distance_1', 'distance_2', 'distance_3']) / 3`

5

Which of the following machine learning algorithms is NOT sensitive to the initial variables used in the optimization algorithm?

A. ○ Hidden Markov models

B. ○ Artificial neural networks

C. ○ Random forests

D. ○ Support vector machines

E. ○ k - nearest neighbors

6

You have two fair coins and one coin with heads on both sides. You pick a coin at random and toss it twice. If it reads heads both times, what is the probability it also reads heads after a third toss?

A. ○ 1 / 6

B. ○ 1 / 3

C. ○ 1 / 2

D. ○ 2 / 3

E. ○ 5 / 6

7

Customers arrive at a shop according to a Poisson process with rate parameter $\lambda = 1$. Suppose we observe customer arrivals from time $t = 0$ to time $t = T$, and find that only one customer arrived during this period. What is the probability that this single customer arrived between time $t = 0$ and time $t = T/4$?

A. ○ $Te^{-T/4}/4$

B. ○ $1 - e^{-T/4}$

C. ○ 1 / 4

D. ○ $e^{-T/4}$

E. ○ None of the above

8

You are trying to assess transit usage trends in a region that exhibits significant tourist traffic. It seems that your data may be heteroskedastic, as none of your predictors are producing good standard errors. Which of the following techniques would be LEAST logical as an immediate next step in mitigating this problem?

A. ○ Removing seasonality from the transit data

B. ○ Differencing between time steps in the time series data

C. ○ Adding data that is lagged by 12 months as an additional covariate

D. ○ Adding data about the number of airline flights into and out of the region as an instrumental variable

E. ○ Logarithmically transforming the predictor variables

9

Consider a function $f(x, y)$ of two variables x and y. Which of the following statements is ALWAYS true? Here, max_k and min_k refer to the maximum over k and the minimum over k, respectively.

A. ○ $max_x min_y f(x, y) = min_y max_x f(x, y)$

B. ○ $max_x min_y f(x, y) <= min_y max_x f(x, y)$

C. ○ $max_x min_y f(x, y) >= min_y max_x f(x, y)$

D. ○ $max_x min_y f(x, y) < min_y max_x f(x, y)$

E. ○ None of the above, because the answer depends on the specific functional form of f

10

Suppose you are tasked with building a classifier from a small set of observations. Which of the following algorithms would be the MOST appropriate?

A. ○ Random forests

B. ○ Support vector machines

C. ○ Naïve Bayes classifiers

D. ○ k - nearest neighbors

E. ○ Decision trees

11

Let $X_1, X_2, ..., X_N$ denote a set of independent and identically distributed random variables drawn from a Pareto distribution with $0 < \alpha <= 1$ and $x_m > 0$. (A Pareto distribution has a probability distribution function given by $p(x) = \frac{\alpha x_m^\alpha}{x^{\alpha+1}}$ for $x >= x_m$.) In the limit as N tends towards infinity, which of the following statements about the sample average of the $X_i, 1 <= i <= N$, is true?

A. ○ The sample average converges almost surely to the expected value

B. ○ The sample average does not converge

C. ○ The sample average converges with probability 1 to the expected value

D. ○ The sample average is zero

E. ○ None of the above

12

In Python, if you had to iteratively read over two files line-by-line, which of the following would be the BEST way to accomplish this task?

A. ○ Use `with open()` to open the two files as `f1` and `f2` , then use `readline()` and a `for` loop to iteratively read lines from each file

B. ○ Use `open()` to open the two files as `f1` and `f2` , then use `readline()` and a `for` loop to iteratively read lines from each file

C. ○ Use `with open()` to open the two files as `f1` and `f2` , then use `zip()` to iterate over the two files together

D. ○ Use `open()` to open the two files as `f1` and `f2` , then use `zip()` to iterate over the two files together

E. ○ Implement a file `seek()` function and call the function for the two files simultaneously

13

An alternative to k - means clustering is k - medoids clustering. This algorithm chooses actual data points as centers, as opposed to choosing centroids (the mean of data points in a cluster) as centers. Which of the following BEST describes why the k - mediods algorithm is often used over the k - means algorithm?

I. The k - medoids algorithm runs faster than the k - means algorithm does
II. The k - medoids algorithm is more robust to outliers than the k - means algorithm is
III. It is easier to choose the value of k in the k - mediods algorithm than in the k - means algorithm

A. ○ I only

B. ○ II only

C. ○ III only

D. ○ I and II only

E. ○ II and III only

14

Which of the following statements about the convergence properties of the perceptron algorithm is true?

A. ○ It does not always converge even if the training data is linearly separable

B. ○ It may sometimes converge even if the training data is not linearly separable

C. ○ It always converges if the training data is linearly separable, but it will always fail if the training data is not linearly separable

D. ○ No general rule can be stated about the convergence properties of the perceptron algorithm without specifying whether the learning rate is adaptive

E. ○ None of the above

15

In Python, you have a `pandas` dataframe `df` with the columns `student_id` , `exam_id` , and `grade` . You want to drop the grades for exams 1 and 3 and keep the grades for exams 2 and 4. Which of the following is the BEST way to accomplish this task?

A. ○ Execute `df[df['exam_id'] == 2]` and `df[df['exam_id'] == 4]` to filter for the grades from exams 2 and 4, then append the 2nd dataframe to the 1st

B. ○ Execute `df[df['exam_id'] == 2]` and `df[df['exam_id'] == 4]` to filter for the grades from exams 2 and 4, then convert to sets and take the union

C. ○ Execute `df[df['exam_id'] == 2]` and `df[df['exam_id'] == 4]` to filter for the grades from exams 2 and 4, then convert to sets and take the intersection

D. ○ `df.filter('exam_id' == 1).filter('exam_id' == 2)`

E. ○ `df[df['exam_id'] in [2,4]]`

Complete