SimpleBet Data Science and Data Engineering Test

Name: _____

Required Time to Complete: _____

***Instructions*: Data Science candidates will be evaluated more heavily on statistics/machine learning-related questions, and data engineering candidates will be evaluated more heavily on data architecture and efficiency questions.**

1. Circle all that apply. Let X and Y be categorical random variables, both with domain {-1,0,1} with uniform probability. Corr(X, Y) = ?

   a. $\frac{Cov(X,Y)}{sqrt(Var(X)Var(Y))}$

   b. $\frac{Cov(X,Y)}{sqrt((E(X^2)-E(X)^2)(E(Y^2)-E(Y)^2))}$

   c. $\frac{Var(X)-Var(Y)-E(XY)}{sqrt((E(X^2)-E(X)^2)(E(Y^2)-E(Y)^2))}$

   d. $\frac{E(XY)-E(X)E(Y)}{sqrt((E(X^2)-E(X)^2)(E(Y^2)-E(Y)^2))}$

   e. 0

   f. -1

   g. 1

   h. $\frac{Pr(XY=1)-Pr(XY=-1)}{sqrt((E(X^2)-E(X)^2)(E(Y^2)-E(Y)^2))}$

2. If you selected more than one option above, please prove below how they are all equivalent.

3. Let's continue thinking about random variables X and Y. Recall that X and Y are categorical random variables, both with domain {-1,0,1} with uniform probability.

   We observe 5 data points, with the following realizations for r.v.'s X and Y -
   Data Set for X: [0, 1, -1, -1, -1]
   Data Set for Y: [0, -1, 1, 1, 0]

   What is Pearson's correlation coefficient based on this sample?

   Under the null hypothesis that X and Y are uncorrelated, what is the probability of observing a data set at least as correlated (negatively or positively) as the above data set?

   Does this mean we should reject the null hypothesis that X and Y are uncorrelated? Why or why not?

4. Suppose we are trying to combine various features/signals into probabilities, based on past historical data. Compare and contrast the pros and cons of 3 supervised learning approaches to this problem (ex: logistic regression, random forests, ANNs, etc.)

Approach #1:	Approach #2:	Approach #3:
Pros		
Cons		

5. CODING CHALLENGE. You will use KMeans to cluster NBA players based on how similar they are in playing style. Please don't spend more than 15 minutes crafting the features used for clustering. This exercise should not take you more than 30 minutes total.

   The business goal is to use the player clustering you generate to create a signal on how unbalanced a team lineup is (for example, too aggressive or too defensive). You are being assessed on your ability to write production-worthy code, your coding style/efficiency, and to a much lesser extent, your ability to feature engineer to meet a business goal.

   Requirements:

   - You must code in Python, and attach a single ".py" file along with your answers to this quiz.
   - Use the following URL to grab a subset of NBA players to cluster: https://stats.nba.com/stats/commonallplayers?LeagueId=00&Season=2016-17&IsOnlyCurrentSeason=0
   - Use the following URL to grab summary statistics for NBA players: https://stats.nba.com/stats/playercareerstats?PerMode=PerGame&PlayerID=X where "X" is the relevant player id grabbed from the first URL.
   - Your Python code must
     - Use K-means for clustering from scikit-learn
     - Grab and massage data from the above URLs to feed into K-Means
     - Automatically choose the right parameters for K-means when producing the final clusters
     - When executing the python file, it should print the final player clusters in the following format {0: [{player_id: 24132, first_name: "Kobe", last_name: "Bryant"}, {…}, {…}, …], 1: […], 2: […], …}

6. Now that you've worked with NBA data, draw a basic schematic on how you would organize NBA data (such as players, teams, games, etc.) into SQL tables. Please consider performance of read/write workloads. Indicate what tables you would create, what columns they would have, what primary keys you'd have, and what indexes you'd create.

7. The following database query is taking over 20 minutes to execute:

   select g.A,g.B,g.C,….,g.Z,, r.W, t.F from gamestb g inner join teamstb t on t.A = g.A left join rankingstb r on r.A = g.A where g.F < 1 order by g.P;

   Circle the TOP TWO things you would prioritize investigating in the data layer, in order to speed the query up?

   a. Investigate reducing the number of columns in each table.
   b. Investigate running on a machine with more RAM.
   c. Investigate if the columns the query joins on have indexes.
   d. Investigate if the database engine is choosing the wrong query plan.
   e. Investigate moving relevant data up the memory hierarchy.

   Please explain your prioritization in detail.

   If the database engine chooses the wrong query plan, explain strategies to ensure the database engine chooses the right query plan.