Name: Sanad Masannat CCID: sanad Date: 5/11/2021

Instructions: Each question is worth the given amount of points, and the whole quiz is worth nine points in total. Answer each question to the best of your ability. Read all instructions carefully. Submit your quiz to eClass before Sunday at 11am. You may submit your quiz as a pdf, a docx, or as a zip file of images. **It is your responsibility to ensure the TAs and instructor can read your answers**, if you’re concerned about that please type your answers when possible.

**Multiple Choice (0.5 points each)**. Circle the answer closest to the one you would give.

**Q1 (0.5 points).** Which of these is the best overfitting definition (mentioned in the regression and decision trees lecture)?

(A.) An overfit model is a model with poor predictive performance

(B.) An overfit model is too focused on irrelevant elements of the training data

(C.) An overfit model performs above some “fitness” function

**(D.)** An overfit model predicts a dog when given a cat

**Q2 (0.5 points).** Below is a set of comparisons between Flow and Drama Management player models. Circle the most specific and accurate comparison.

(A.) Both are player-modelling techniques

(B.) Both require a large amount of tagging of in-game player actions

(C.) Both require designers to implement finite state machines or B trees

**(D.)** Both involve altering player experience to more closely match an ideal

**Q3 (0.5 points)**. Which of the following is untrue of Mixture Models?

(A.) They allow for a final representation where each datapoint “belongs” to multiple clusters

(B.) They can be applied to the same datasets one could apply K means clustering to

**(C.)** They typically are the fastest clustering approach (in terms of determining a final model)

(D.) None of the above are untrue

**Q4 (0.5 points)**. In the case study with Dr. Alexander Zook we saw an example of using a player’s Elo rating (skill rating) as part of the input to a linear (and later logistic) regression to predict match quality (represented as average win rate). Let’s say that a designer decided to include Bartle’s player types as well, a single type for each player in a match as part of the input to their model, despite the fact that these will likely not be all that helpful for this problem. What would be a way to implement this to minimize any loss of accuracy?

(A.) No changes needed; a logistic regression can take Bartle’s player types as input

**(B.)** Each Bartle Player Type should be converted to some arbitrary number (e.g., Killer = 0, Achiever = 1, Socializer = 2, Explorer = 3)

(C.) They should change to use something like a Decision Tree or Random Forest

(D.) It would not be possible to use Bartle’s Player Types for their problem

**Q5 (2.5 points).** Answer the below parts to the best of your ability.

**Q5.A. (0.5 points**). Create a dataset for clustering. Your dataset should have 8 datapoints. Each of these datapoints should have the variables/features: logins, items, and friends. Each of the variables can vary from 0-25 (inclusive**). You should not have the same value twice across any of the datapoints or variables** (this means you’ll have to use every value once except one)**.**

One valid answer could be (your answer should not be within 9 edits/changes of this):

1: 1,2,3

2: 4,5,6

3: 7,8,9

4: 10,11,12

5: 13,14,15

6: 17,18,19

7: 20,21,22

8: 23,24,25

**1: 3, 23, 13**

**2: 6, 20, 10**

**3: 9, 17, 7**

**4: 12, 14, 4**

**5: 15, 11, 1**

**6: 18, 8, 16**

**7: 21, 5, 19**

**8: 24, 2, 22**

**Q5.B. (0.5 point)** What would be the best clustering method for your data from **Q5.A**? Why?

**I would use Single Linkage clustering; this is because there is a more linear pattern within the data sets rather than blobs of data. There is no clear mean/median centre point to be able to use. It is also a bit tough to have average items and friends (one can’t have a fractional item or friend).**

**Q5.C. (1.5 point)** Give valid final clusters for K=3 using K means clustering for your answer to **Q5.A**. You do not have to give the centroids/centers of the clusters, but you may if you like. Name each of the clusters A, B, and C (which is which doesn’t matter). So, you could say for example A: (1,2,3) if datapoints 1, 2, and 3 would all be in one cluster together. For a distance function use: abs(logins1-logins2) + abs(items1-items2) + abs(friends1-friends2).

**A(4,5)**

**B(6,7,8)**

**C(1,2,3)**

**Q6 (1 point)** Imagine you have been asked to come up with some player modelling systems by the game designers of a virtual pet game (where players raise digital animals, both real and fictional). What player modelling technique described in class would you use for each situation, and why?   
(A) A model to map from a series of features that describe a player to a prediction of whether they will buy a new outfit for their pet or not (**0.25 points**),

(B) a model that can identify what groups of players exist within the game (**0.25 point**),

(C) a model that maps from a set of features to how much real world money a player will spend in the game (**0.5 points**)?   
Make sure to give brief (~1-2 sentence) justifications for each answer.

1. **Decision Trees. This is because we can split data up using comparisons and because the outcome of this model would not fit a linear/logistic model due to binary outcome which a decision tree can handle better,**
2. **Mixture Models. This is because many players would have different playstyles or types of playstyle which could be an in-between of 2 different playstyles. Players can also not embody as singular playstyle wholly, so Mixture Models are the best in my opinion.**
3. **Linear Regression. This is because we can easily map a function using specific numerical values/features(time spent playing, previous money spent, number of pets owned etc.) and make various calculations with those numeric variables to get a decent model that allows us to get an approximate amount of money one could spend.**

**Q7 (1 point)**. Describe a situation where it would be better to use KNN instead of a decision tree.

**When we have a lot more data with more variables to take into account alongside data with no patterns, we would rather avoid decision trees due to being slower in nature and as there wouldn’t be a clear pattern it would be tougher to split data up into different branches. Whereas with KNN, we can use a nearby centre point to group data together which would not happen with decision trees due to nature of data. If we have a lot of training data, decision trees can cause overfitting from the training data while KNNs can avoid this.**

**Q8 (2.5 point).** Answer the below parts to the best of your ability.

**Q8.A. (0.5 points)** Create the data for a decision tree. Your 7 datapoints will have the variables/features (level, difficulty, skill) and a final variable/feature that you are attempting to predict (churned). Level can have values between 1-5 (inclusive), difficulty can have values between 1-3 (inclusive) and skill can have values between 1-3 (inclusive).

**Fill in the remainder of this table to create your 7 datapoints:**

|  |  |  |  |
| --- | --- | --- | --- |
| **level [1,5]** | **difficulty [1,3]** | **skill [1,3]** | **Churned** |
| **1** | 2 | 3 | T |
| **5** | **1** | **1** | T |
| 2 | **2** | 3 | T |
| **1** | 2 | **3** | T |
| **5** | **2** | 3 | F |
| 1 | **1** | **3** | F |
| **5** | 3 | **1** | F |

For example, your answer could be (do not have an answer within 6 edits of this:

|  |  |  |  |
| --- | --- | --- | --- |
| **level** | **difficulty** | **skill** | **Churned** |
| **1** | 1 | 1 | T |
| **5** | **1** | **1** | T |
| 1 | **2** | 1 | T |
| **5** | 1 | **3** | T |
| **5** | **2** | 1 | F |
| 1 | **1** | **3** | F |
| **5** | 1 | **1** | F |

**Q8.B. (2 points)** Give the final decision tree predicting Churned for your answer to **Q8.A.** For a splitting metric use: ( **min**(datapoints with False Churned, datapoints with True Churned) divided by all datapoints at this node), with the goal being to minimize this value. For example, if there was 1 datapoint with Churned True and 2 datapoints with Churned False at a node the metric value would be 1/3, and if there was 2 datapoints with Churned True and 1 datapointwith Churned False the metric value would still be 1/3. You may use less than, greater than, less than or equal to, and greater than or equal to for the splits/choice nodes.

Assume no threshold, but if your tree reaches a depth greater than 3 you can just take the majority (or a random choice if tied) at that point (i.e. if you have two choice nodes on one branch). You can either draw your tree or specify itwith text.

For example:   
root = level>1,   
level>1 –(True)->skill<3

level>1 –(False)->True

skill<3-(True)->True

skill<3-(False)->False

This would specify an (incorrect) tree where a choice node (level>1) was the root, which split to a decision node value that returned True when the choice node was False andto another choice node (skill<3) when True. Or represented visually:

Diagram

Description automatically generated

**root = level>2,   
level>2 –(True)-> Difficulty>1**

**level>2 –(False)-> True**

**Difficulty>1–(True)-> True**

**Difficulty>1–(False)-> False**

**EXTRA CREDIT (0.5 points)** One of the major reasons Bartle’s Player Types haven’t been used in a major industry game is the heavy authoring burden of having to tag each player action with one or more player type (Killer, Achiever, Explorer, Socializer). Describe a strategy to automatically tag player actions that requires *significantly less designer effort than just tagging every action by hand*. Given an input datapoint representing a player action your approach should output 1 or more player types that player action represents or relates to. Give an example of a specific datapoint and what method(s)/model(s) you would use to tag it (**0.25 points, or 0.5 points if the method you listed doesn’t require any training data**). In addition, explain what the training data for these method(s) would look like and how you would produce it, if the method(s) you listed required any training data (**0.25 points if methods you listed require any training data**).