CSCI 585: Final exam

Assignment description

Please read the following carefully, before starting the test:

• the exam is open books/notes/devices/minds - feel free to look up whatever you want, from wherever (but not whomever)!

• WHAT TO ANSWER:

- you need to answer the **first 5** questions (Q1..Q5), 5 points each
- after that, you can **pick any 5**, out of the remaining 10 questions (Q6..Q15), 2 points each; you CAN answer more questions if you want (6,7,8,9 or all 10)
- we will score everything you answer, add the scores (including partial ones), **CAP** them at 35 [>35 becomes 35]. How cool!

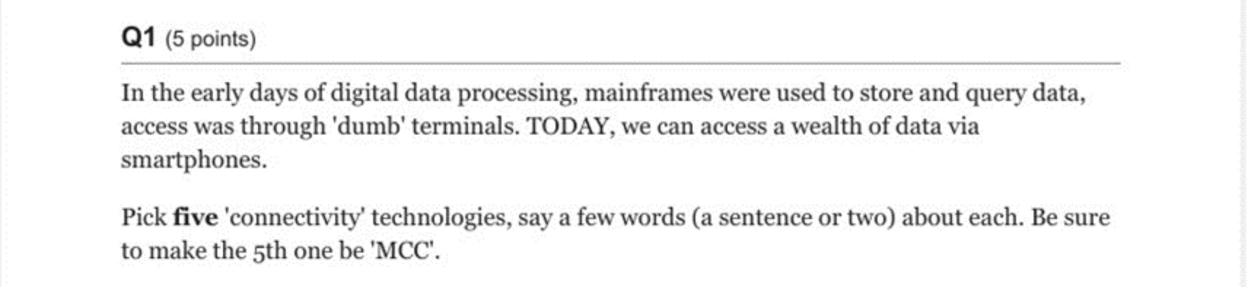
• there are no 'trick' questions, or ones with long calculations or formulae, and there's certainly nothing to memorize [it's all OPEN, duh :)] It doesn't mean the questions are trivial! There are open-ended questions (which means there is more than one right answer), but they are not subjective ones (which means they are not about your opinion/viewpoint). Please do answer carefully: answer just WHAT IS ASKED, otherwise you won't get points (eg. if a question is -ABOUT- column fragmentation, don't DESCRIBE/DEFINE column fragmentation!). It's the quality (of your answer) that counts, not quantity (verbosity), or extraneous details...

• please do NOT cheat - this means NOT communicating with anyone via any device/medium/channel - you will get a 0, and be reported to SJACS, if you are found to have cheated; ANY attempt to get help from others in any form is a VIOLATION, as per https://policy.usc.edu/scampus-part-b/, sections 11.11 through 11.14 [read it, if you are not familiar with it]

• when the time is up (90 minutes), stop your work, then spend the rest of time (30 minutes) on submission [students with DSP accommodations - your exam duration will be as per DSP determination] - **submitting past the deadline comes with a penalty**, because it is not fair to others if you go over when they don't; note that you need to submit each answer separately (not all of them as a single PDF), this is a Crowdmark requirement

Fun fact: 'data' occurs 24 times (25, including in this line!), at least once in each question :)

Good luck! Hope you enjoy answering the questions, hope you find them to be easy+fun+stimulating.



Q1 (5 points)

In the early days of digital data processing, mainframes were used to store and query data, access was through 'dumb' terminals. TODAY, we can access a wealth of data via smartphones.

Pick **five** 'connectivity' technologies, say a few words (a sentence or two) about each. Be sure to make the 5th one be 'MCC'.

Here are possible answers:

* ODBC

* JDBC

* DAO, RDO etc.

* COM, COM+

* CORBA

* cgi-bin scripts

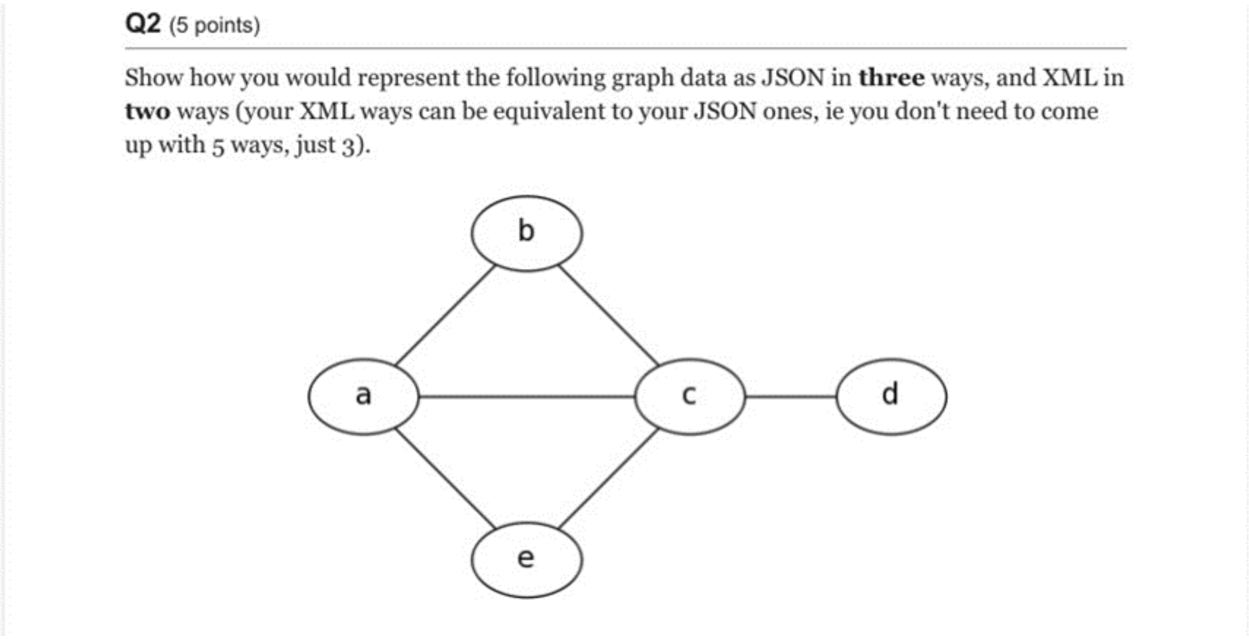
* CFML

* RMI

* ...

* MCC: microservices, containers, clouds - to date, this is the most flexible connectivity option; a frontend app (on a smartphone or website) connects with multiple, distributed microservices, that run in containers deployed in clouds.

Details: Each valued answer will be worth 1 point. The answer should be included MCC, otherwise most 4 points.



Q2 (5 points)

Show how you would represent the following graph data as JSON in **three** ways, and XML in **two** ways (your XML ways can be equivalent to your JSON ones, ie you don't need to come up with 5 ways, just 3).

Here are possible ways:

```
{

 “graph”: {

          [

   ‘a’: [‘b’, ‘c’, ‘e],

   ‘b’: [‘a’, ‘c’]

   ...

          ]

  }
}

{
   “verts”: [‘a’, ‘b’, ‘c’, ‘d’, ‘e’],
   “edges”: [[‘a’, ‘b’], [‘a’, ‘c’], [‘a’, ‘e’]...]
}

{
   “edges”: [[‘a’, ‘b’], [‘a’, ‘c’], [‘a’, ‘e’]...]
}
```

Their XML translations are straightforward, eg.

```
<graph>
  <vertices>
   <vertex> <name>a</name> <connectedTo>b,c,e</connectedTo>
   ...
```

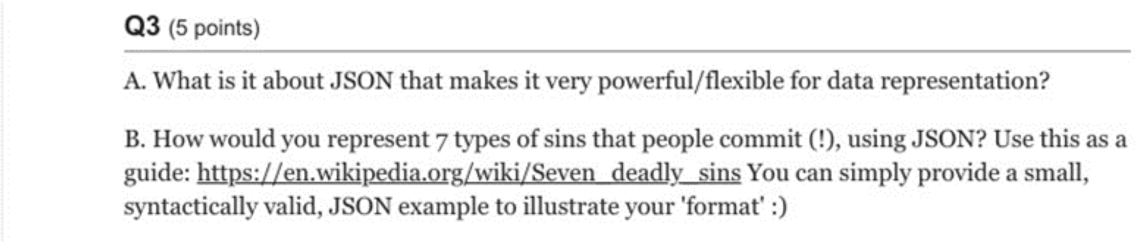
```
 </vertices>

</graph>
```

There could be alternate ways to specify the above.

Details: If 3 representations are all perfect than give 5 points.

Any format error, deduct 1 point. i.e. missing keys, or “]” “[”, “:”.

Any missing data content deduct 1 point. Missing any of “a,b,c,d,e” or missing connection relationship.



Q3 (5 points)

A. What is it about JSON that makes it very powerful/flexible for data representation?

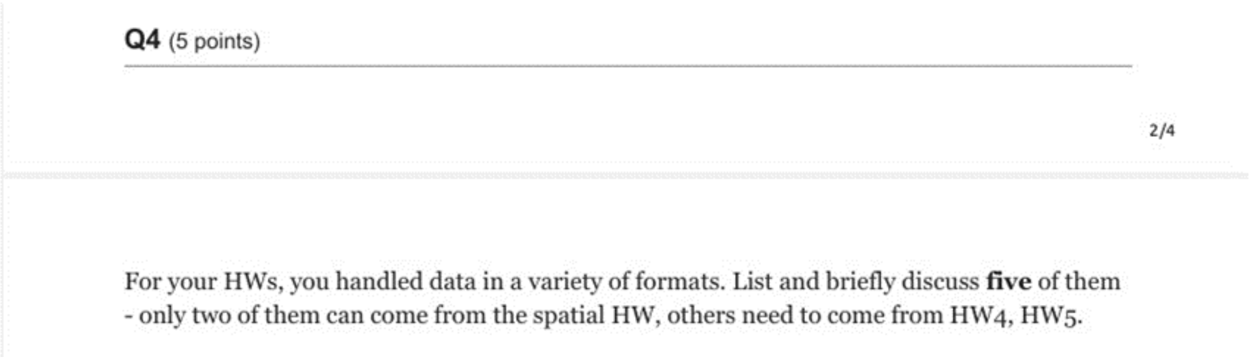
B. How would you represent 7 types of sins that people commit (!), using JSON? Use this as a guide: https://en.wikipedia.org/wiki/Seven_deadly_sins You can simply provide a small, syntactically valid, JSON example to illustrate your 'format' :)

JSON can be endlessly nested, using objects (key:value pairs) and arrays (lists) - a value in key:value could itself be an array, and an array element could be an object, objects can have objects as values, arrays can have arrays as elements [ie. 4 combinations of ‘array’ and ‘object’]. (2 points)

```
{
 “deadlySins”: [

        {

   “greed”: [“wanted much more money”...],

   “envy”: [“hated neighbor’s big house”...]

   ...

        }

 ]

}
```

The idea is to list the sin types as arrays, and for each sin, an array of them. Other representations are OK as long as they are valid JSON and also express the data correctly (7 sin types, each a set of values).

(3points).



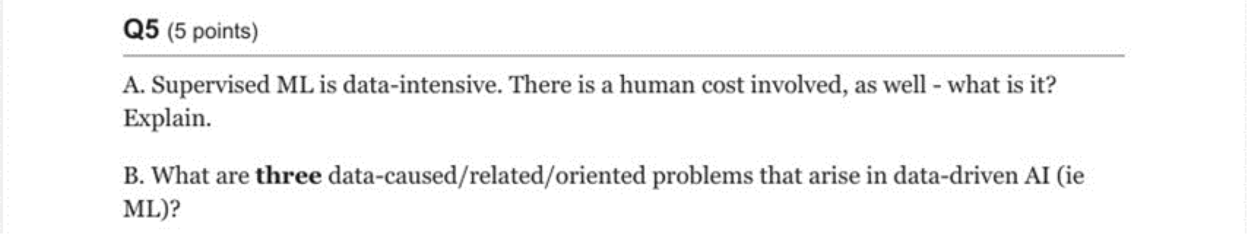
Q4 (5 points)

2/4

For your HWs, you handled data in a variety of formats. List and briefly discuss **five** of them - only two of them can come from the spatial HW, others need to come from HW4, HW5.

Here are possible answers: KML, Shapefile, JSON, csv, ARFF, hd5, a folder/directory of images...

(each of the format worth 1 point)



Q5 (5 points)

A. Supervised ML is data-intensive. There is a human cost involved, as well - what is it? Explain.

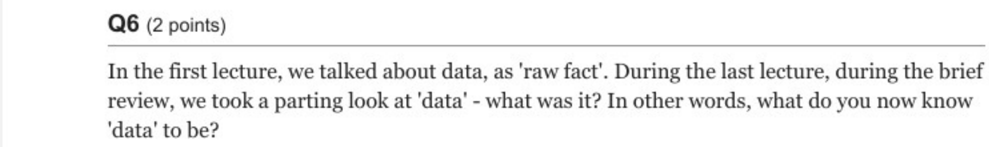
B. What are **three** data-caused/related/oriented problems that arise in data-driven AI (ie ML)?

The human cost is that of **labeling/annotating/grouping** the data. Before an NN can be trained, it needs annotated data that contains class labels (eg for a CNN, these would be images and their class labels) - this needs to be done by humans who know what images correspond to what labels; at the least, we need to collect images and place them in different folders that correspond to labels, eg. cats/, dogs/.

(2points)

Data-related problems include bias, foolability (eg by subtly altering pixels in an input image), and lack of explainability.

(each of them worth 1 point)



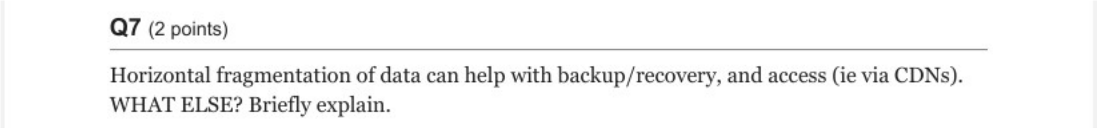
Q6 (2 points)

In the first lecture, we talked about data, as 'raw fact'. During the last lecture, during the brief review, we took a parting look at 'data' - what was it? In other words, what do you now know 'data' to be?

2 points for giving an alternate understanding of data including -

1. Data can be seen as **VALUES,**
2. **for CATEGORIES** (ie. columns, descriptors, characteristics...) that we use to describe an **entity.** Other similar answers are acceptable.

Grader’s discretion is sought as this could have many answers



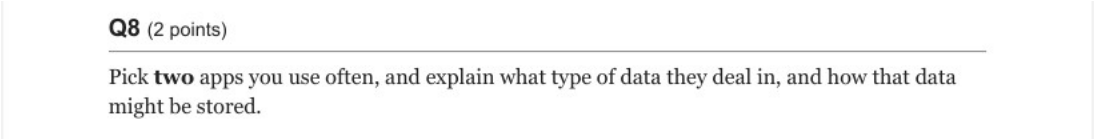
Q7 (2 points)

Horizontal fragmentation of data can help with backup/recovery, and access (ie via CDNs). WHAT ELSE? Briefly explain.

Points –

1. 2 points for stating parallel operations: Horizontal fragmentation can be used for **parallel operations on the fragment, for massive speedup** –
2. If this is not explained but student gives mapreduce as an example: award 1 point

Eg – the MapReduce algorithm is the classic example of this (for a fragmented graph, BSP would be the eqvt).



Q8 (2 points)

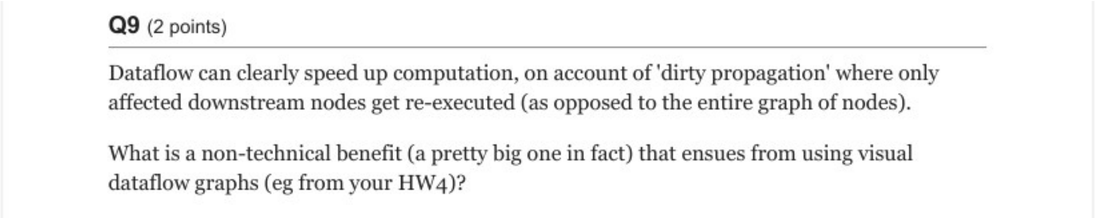
Pick **two** apps you use often, and explain what type of data they deal in, and how that data might be stored.

1. 1 point for explanation of one app

MANY answers are possible, of course :) Eg.

Waze: map data stored as vectors and rasters; user data could be horizontally fragmented documents, ride data could be k:v pairs in main memory, etc.

LinkedIn: resume data would be documents, as would employer job listings, articles could be documents as well [all documents would reside in horizontally fragmented document stores].



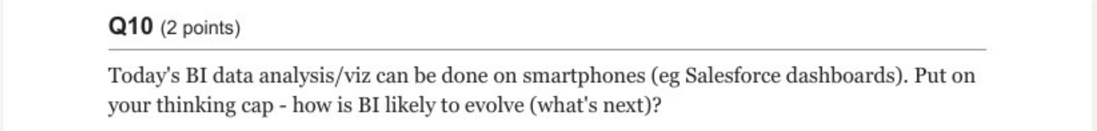
Q9 (2 points)

Dataflow can clearly speed up computation, on account of 'dirty propagation' where only affected downstream nodes get re-executed (as opposed to the entire graph of nodes).

What is a non-technical benefit (a pretty big one in fact) that ensues from using visual dataflow graphs (eg from your HW4)?

1. 2 points for explaining this. [Easy to visualize, debug or understand]

   The non-technical benefit is that a dataflow graph is **self-documenting** - it is easy to examine the nodes, their parameters, and interconnections (ie the graph) and understand exactly what the graph computes; this also lets us be able to know what to modify and where, to affect the output.

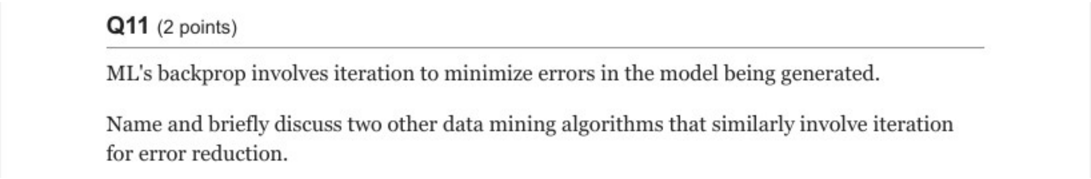


Q10 (2 points)

Today's BI data analysis/viz can be done on smartphones (eg Salesforce dashboards). Put on your thinking cap - how is BI likely to evolve (what's next)?

1. 2 points based on grader's discretion. There is no way to tell what is "novel" and what has been achieved.

BI could occur via VR, and AR - data could be visualized and interacted with, spatially (eg being surrounded by it). NLP can be used to make analysis possible by simply speaking using plain English for ex.

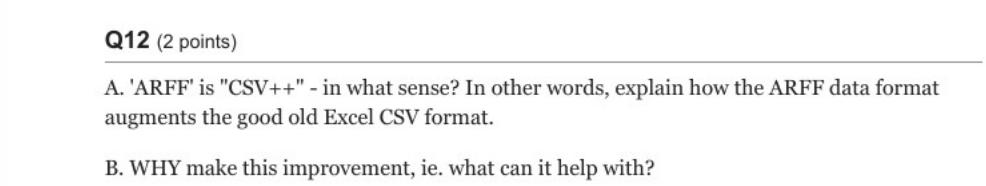


Q11 (2 points)

ML's backprop involves iteration to minimize errors in the model being generated.

Name and briefly discuss two other data mining algorithms that similarly involve iteration for error reduction.

1. 1 point for each correct algorithm instance.
   Clustering, linear regression can both be computed iteratively. Decision trees can be improved via iteration, eg using gradient-boosting.



Q12 (2 points)

A. 'ARFF' is "CSV++" - in what sense? In other words, explain how the ARFF data format augments the good old Excel CSV format.

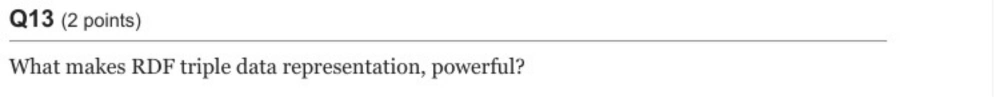
B. WHY make this improvement, ie. what can it help with?

1. 1 point :

ARFF adds data types and comments, to CSV-like comma-separated data - so it's a superset of CSV.

2. 1 point:

It can help enforce proper data entry, ie. the input for a column (eg via a UI) can be checked to make sure it matches the expected type (since we would know the type on account of it being declared earlier).

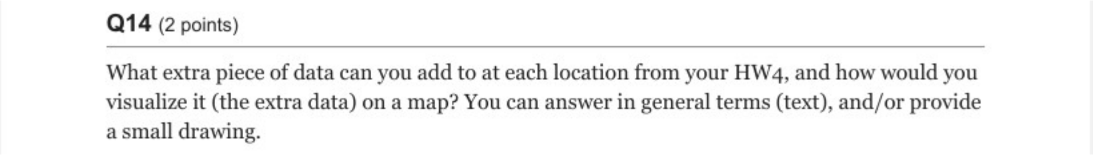


Q13 (2 points)

What makes RDF triple data representation, powerful?

2 points if student communicates that triplets are not restrictive

Triples can be linked up, without bounds, to create arbitrarily-complex graphs - subjects can function as predicates for other triples, predicates can conversely function as subjects, identical subjects or predicates can be 'fused' (merged).



Q14 (2 points)

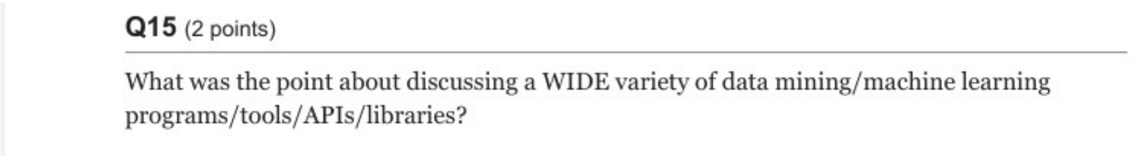
What extra piece of data can you add to at each location from your HW4, and how would you visualize it (the extra data) on a map? You can answer in general terms (text), and/or provide a small drawing.

1 points if the attribute is valid

+1 point for visualization description or drawing

An attribute called ‘popularity’ could be added, or, ‘noiseLevel’, or ‘area’, etc. Such a (numerical) attribute can be depicted using circle radius, when the locations are represented as discs; or we could use fill-color, or different symbols.

- Rating of place could be another attribute



Q15 (2 points)

What was the point about discussing a WIDE variety of data mining/machine learning programs/tools/APIs/libraries?

Marks : 2 marks if the student conveys any one of these –

1. The point is this: data mining/ML has **matured** to the point of being able to be carried out by **low-code or no-code setups**, or
2. if coding is a necessity, via already-available libraries, APIs and frameworks - there is decreasing **need to code anything up “from scratch”.**