RYERSON UNIVERSITY

CIND110
DATA ORGANIZATION FOR DATA ANALYSTS

ASSIGNMENT 2
XML, Information Retrieval, Data Mining, and NoSQL

SECTION: DK0
SUMBITTED BY: ANN SAM
STUDENT NUMBER: 501160843

---

1 Section-A:
1.1 XML

1. Write an XPath expression to find all authors along with their corresponding books.

```
db:open("A3Data", "A3Data.xml")
//(author | title)
```

24 Results, 820 b

```
<author>Gambardella, Matthew</author>
<title>XML Developer's Guide</title>
<author>Ralls, Kim</author>
<title>Midnight Rain</title>
<author>Corets, Eva</author>
<title>Maeve Ascendant</title>
<author>Corets, Eva</author>
<title>Oberon's Legacy</title>
<author>Corets, Eva</author>
<title>The Sundered Grail</title>
<author>Randall, Cynthia</author>
<title>Lover Birds</title>
<author>Thurman, Paula</author>
<title>Splish Splash</title>
<author>Knorr, Stefan</author>
<title>Creepy Crawlies</title>
<author>Kress, Peter</author>
<title>Paradox Lost</title>
<author>O'Brien, Tim</author>
<title>Microsoft .NET: The Programming Bible</title>
<author>O'Brien, Tim</author>
<title>MSXML3: A Comprehensive Guide</title>
<author>Galos, Mike</author>
<title>Visual Studio 7: A Comprehensive Guide</title>
```

2. Write an XPath expression to find the prices of all the books and their genre.

```
db:open("A3Data", "A3Data.xml")
//(price | genre)
```

24 Results, 553 b

```
<genre>Computer</genre>
<price>44.95</price>
<genre>Fantasy</genre>
<price>5.95</price>
<genre>Fantasy</genre>
<price>5.95</price>
<genre>Fantasy</genre>
<price>5.95</price>
<genre>Fantasy</genre>
<price>5.95</price>
<genre>Romance</genre>
<price>4.95</price>
<genre>Romance</genre>
<price>4.95</price>
<genre>Horror</genre>
<price>4.95</price>
<genre>Science Fiction</genre>
<price>6.95</price>
<genre>Computer</genre>
<price>36.95</price>
<genre>Computer</genre>
<price>36.95</price>
<genre>Computer</genre>
<price>49.95</price>
```

3. Write an XPath expression to find the title, price, and the description in the text of the last book in the catalog.

```
db:open("A3Data", "A3Data.xml")
/catalog/book[last()]//(title | price | description)
```

3 Results, 289 b

```
<title>Visual Studio 7: A Comprehensive Guide</title>
<price>49.95</price>
<description>Microsoft Visual Studio 7 is explored in depth,
      looking at how Visual Basic, Visual C++, C#, and ASP+ are
      integrated into a comprehensive development
      environment.</description>
```

4. Write an XPath expression to find the authors and titles of the books which cost more than 40 dollars, along with the respective prices.

```
db:open("A3Data", "A3Data.xml")
/catalog/book[price > 40]//(author | title | price)
```

6 Results, 204 b

```
<author>Gambardella, Matthew</author>
<title>XML Developer's Guide</title>
<price>44.95</price>
<author>Galos, Mike</author>
<title>Visual Studio 7: A Comprehensive Guide</title>
<price>49.95</price>
```

5. Write an XPath expression to find the authors and prices of the books belonging to Computer genre.

```
db:open("A3Data", "A3Data.xml")
/catalog/book[genre = 'Computer']//(author | price)
```

8 Results, 217 b

```
<author>Gambardella, Matthew</author>
<price>44.95</price>
<author>O'Brien, Tim</author>
<price>36.95</price>
<author>O'Brien, Tim</author>
<price>36.95</price>
<author>Galos, Mike</author>
<price>49.95</price>
```

6. Write an XQuery (FLWOR) script to find the titles of the books arranged in ascending order of price, of which the price are more than 30 dollars.

```
for $x in db:open('A3Data')//book
where $x/price > 30
order by $x/price
return $x/title
```

4 Results, 191 b

```
<title>Microsoft .NET: The Programming Bible</title>
<title>MSXML3: A Comprehensive Guide</title>
<title>XML Developer's Guide</title>
<title>Visual Studio 7: A Comprehensive Guide</title>
```

7. Write an XQuery (FLWOR) script to provide only the descriptions of the books which cost less than 5 dollars.

```
for $x in db:open('A3Data')//book
where $x/price < 5
return $x/description
```

3 Results, 356 b

```
<description>When Carla meets Paul at an ornithology
      conference, tempers fly as feathers get ruffled.</description>
<description>A deep sea diver finds true love twenty
      thousand leagues beneath the sea.</description>
<description>An anthology of horror stories about roaches,
      centipedes, scorpions  and other insects.</description>
```

8. Write an XQuery (FLWOR) script which gives the various genre along with the text of the title of the books in them.

```
for $x in db:open('A3Data')//book
order by $x/genre
return $x/genre | $x/title
```

24 Results, 738 b

```
<title>XML Developer's Guide</title>
<genre>Computer</genre>
<title>Microsoft .NET: The Programming Bible</title>
<genre>Computer</genre>
<title>MSXML3: A Comprehensive Guide</title>
<genre>Computer</genre>
<title>Visual Studio 7: A Comprehensive Guide</title>
<genre>Computer</genre>
<title>Midnight Rain</title>
<genre>Fantasy</genre>
<title>Maeve Ascendant</title>
<genre>Fantasy</genre>
<title>Oberon's Legacy</title>
<genre>Fantasy</genre>
<title>The Sundered Grail</title>
<genre>Fantasy</genre>
<title>Creepy Crawlies</title>
<genre>Horror</genre>
<title>Lover Birds</title>
<genre>Romance</genre>
<title>Splish Splash</title>
<genre>Romance</genre>
<title>Paradox Lost</title>
<genre>Science Fiction</genre>
```

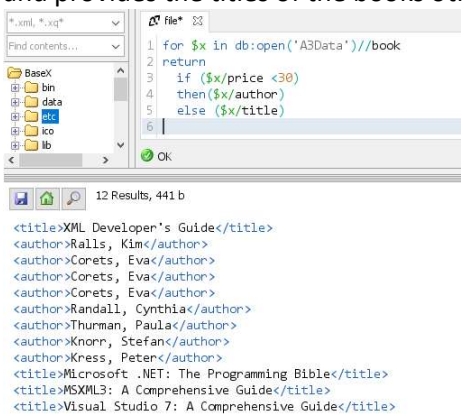
9. Write an XQuery (FLWOR) script which gives the description text showing that the books belongs to Fantasy genre.

```
for $x in db:open('A3Data')//book
where $x/genre = 'Fantasy'
return $x/description | $x/genre
```

8 Results, 779 b

```
<genre>Fantasy</genre>
<description>A former architect battles corporate zombies,
      an evil sorceress, and her own childhood to become queen
      of the world.</description>
<genre>Fantasy</genre>
<description>After the collapse of a nanotechnology
      society in England, the young survivors lay the
      foundation for a new society.</description>
<genre>Fantasy</genre>
<description>In post-apocalypse England, the mysterious
      agent known only as Oberon helps to create a new life
      for the inhabitants of London. Sequel to Maeve
      Ascendant.</description>
<genre>Fantasy</genre>
<description>The two daughters of Maeve, half-sisters,
      battle one another for control of England. Sequel to
      Oberon's Legacy.</description>
```

10. Write an XQuery (FLWOR) script which gives the list of authors whose books cost less than 30 dollars and provides the titles of the books otherwise.



1.2 Information Retrieval (IR)

See *SamErb.Ann.rmd* and *SamErb.Ann.html* files.

---

2 Section-B
2.1 Data Mining

1. Use the K-means algorithm to cluster this dataset. You can initiate the calculations by assuming K=2 and assume that the records with RIDs 103 and 104 are used as the initial cluster centroids.

Using the Euclidean Distance formula:

$$d(x,\ y)\ =\ \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

The distance/proximity was calculated between the data points:

Pairwise Euclidean distance between all pairs of datapoints						
ID	d(ID = 101,j)	d(ID = 102,j)	d(ID = 103,j)	d(ID = 104,j)	d(ID = 105,j)	d(ID = 106,j)
101		28.28	22.36	5.00	5.00	32.02
102	28.28		10.00	32.02	25.00	5.00
103	22.36	10.00		26.93	20.62	11.18
104	5.00	32.02	26.93		7.07	36.06
105	5.00	25.00	20.62	7.07		29.15
106	32.02	5.00	11.18	36.06	29.15	

Example calculations:

$$d(101, 102) = \sqrt{(30-50)^2 + (5-25)^2} = 28.28$$
$$d(101, 103) = \sqrt{(30-50)^2 + (5-15)^2} = 22.36$$
$$d(101, 104) = \sqrt{(30-25)^2 + (5-5)^2} = 5.00$$

Looking at the distance calculations at each initial centroid (103 and 104), we can find the new position of the centroid by calculating the average position of all the points in each cluster. As we have identified the first cluster around datapoints 104, 101, and 105, we can now calculate the average position as follows:

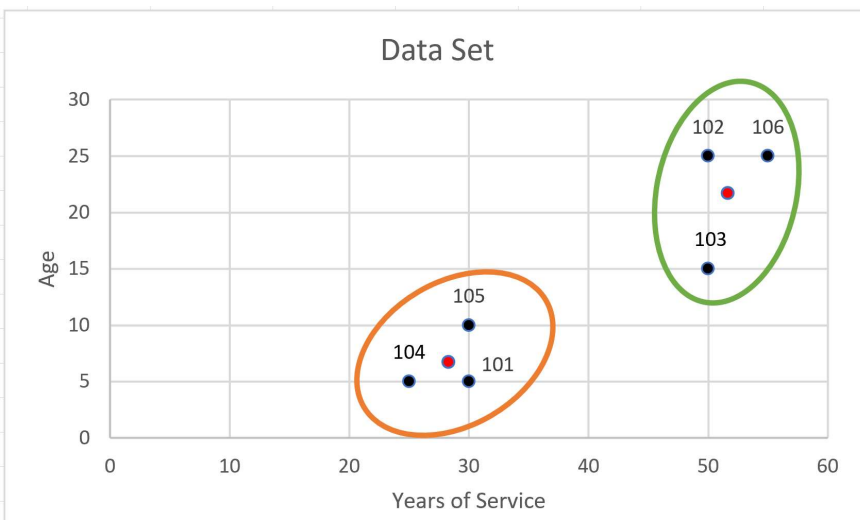
$$centroid_{C1} = \left(\frac{30+25+30}{3}, \frac{5+5+1}{3}\right) = (28.33, 6.67)$$

The same calculation can be done for the second cluster around datapoints 103, 102, and 106:

$$centroid_{C2} = \left(\frac{50+50+55}{3}, \frac{25+15+25}{3}\right) = (51.67, 21.67)$$

We can visually see the two clusters, K=2 when you plot the dataset:

Dataset for K-means Clustering			
ID	age	years	Cluster ID
101	30	5	1
102	50	25	2
103	50	15	2
104	25	5	1
105	30	10	1
106	55	25	2



- *The average position of all the points of a cluster (centroid) are plotted in red*

2. Provide a brief description on the difference between describing discovered knowledge using clustering and describing it using classification.

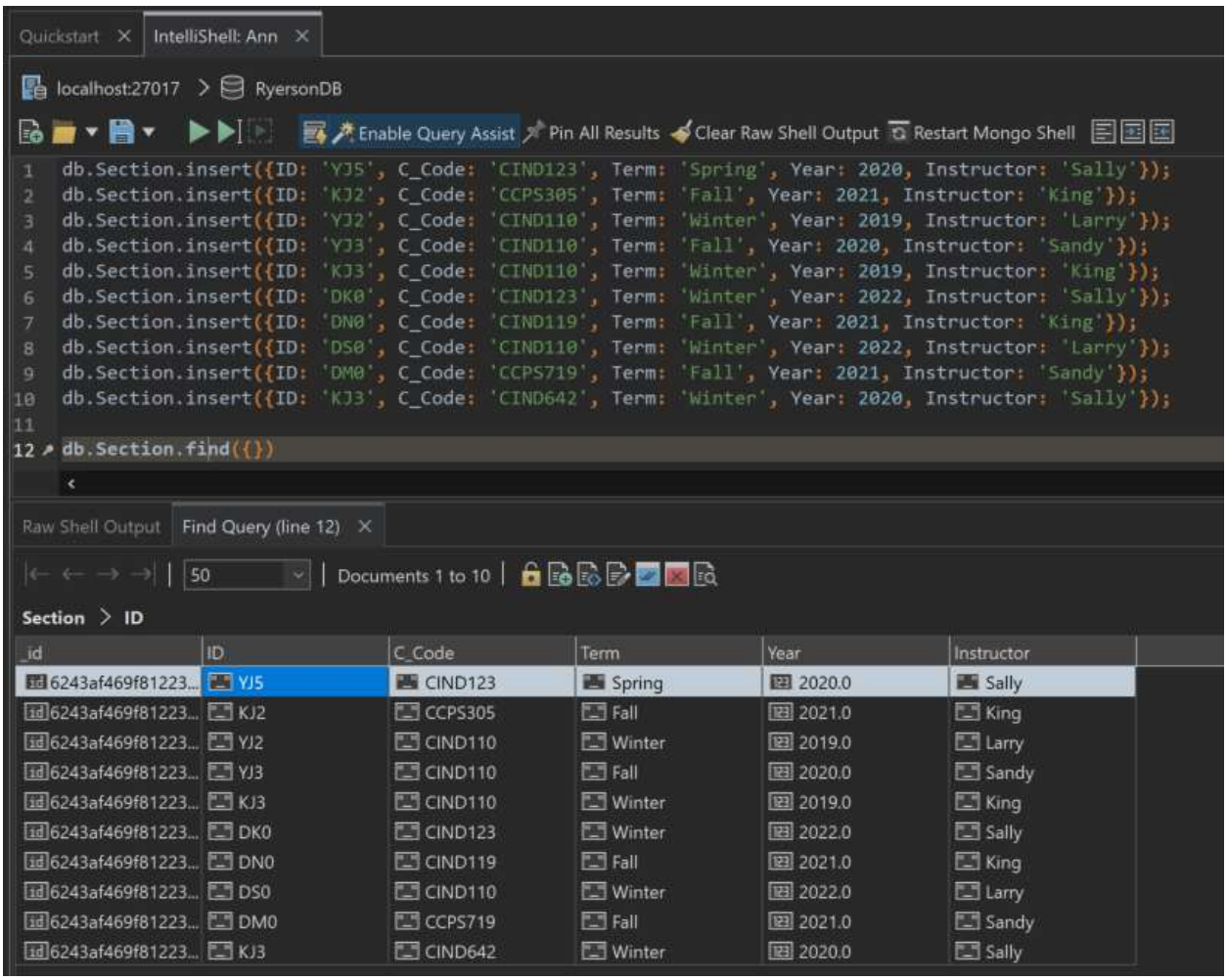
Given the small two-dimensional dataset and lack of predefined classes or labels of the data points, we were able to group the data into two clear clusters. Without any additional data we see that there is a relationship between age and years of service: the younger the age, the less years of service and the older the age, the more years of service.

In terms of classification, any new data points we receive, we can then use relationship found using K-means to determine which class/cluster the new datapoint belongs to. If we were to attempt to classify

this dataset before using the K-means algorithm, it would be much harder to determine a relationship of any between the small dataset. One data analyst could easily classify the data differently from another, while K-means clustering allows a more streamline approach.

2.2 NoSQL

1. **Create the corresponding MongoDB database and Collection with the values.**



2. **In the following, there are five questions to retrieve the output by NoSQL Queries from the Collection created. Write down the corresponding MongoDB codes for each question and provide the output as well.**

   a. **Find the course codes that instructor King taught in each term:**

```
db.Section.find({Instructor: 'King'}, {C_Code: 1, Term: 1, Instructor: 1} );
```

```
// Find the course codes that instructor King taught in each term //
db.Section.find({Instructor: 'King'}, {C_Code: 1, Term: 1, Instructor: 1} );
```

Raw Shell Output | Find Query (line 15)

50 | Documents 1 to 3

Section > C_Code

_id	C_Code	Term	Instructor
6243af469f81223...	CCPS305	Fall	King
6243af469f81223...	CIND110	Winter	King
6243af469f81223...	CIND119	Fall	King

   b. **Find out the courses and their instructors after year 2020 onward:**

```
db.Section.find({Year: {$gt:2020}}, {C_Code:1, Instructor:1} );
```

```
// Find out the courses and their instrutors after year 2020 onward //
db.Section.find({Year: {$gt:2020}}, {C_Code:1, Instructor:1} );
```

Raw Shell Output | Find Query (line 18)

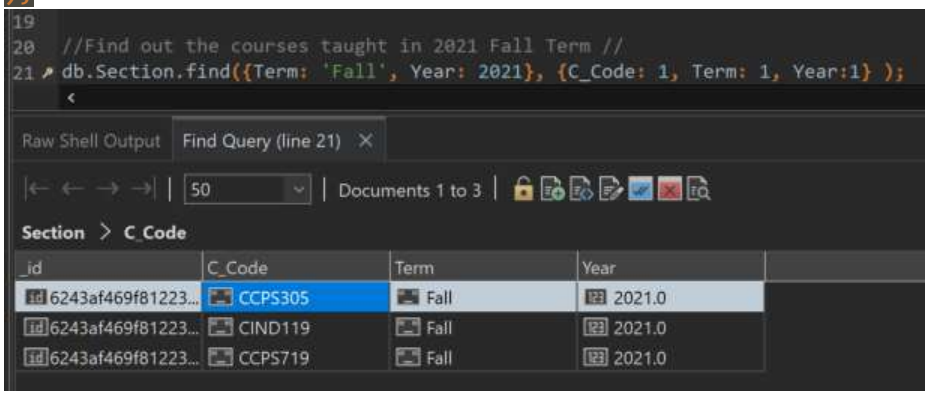
50 | Documents 1 to 5

Section > C_Code

_id	C_Code	Instructor
6243af469f81223...	CCPS305	King
6243af469f81223...	CIND123	Sally
6243af469f81223...	CIND119	King
6243af469f81223...	CIND110	Larry
6243af469f81223...	CCPS719	Sandy

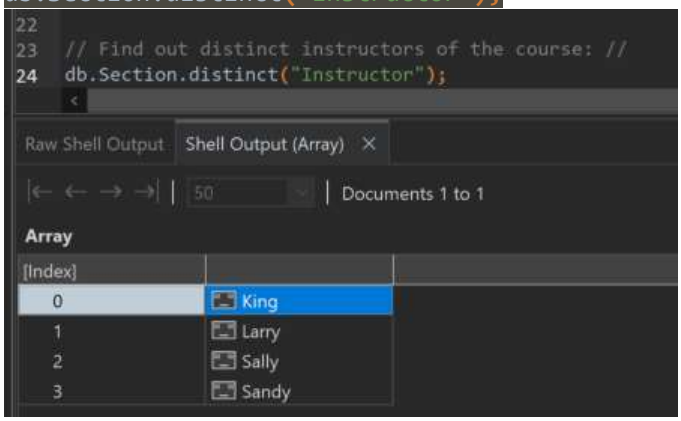
c. **Find out the course taught in 2021 Fall Term:**

```
db.Section.find({Term: 'Fall', Year: 2021}, {C_Code: 1, Term: 1, Year:1}
);
```



d. **Find out distinct instructors of the course:**

```
db.Section.distinct("Instructor");
```



e. **Find distinct courses taught in the program by grouping them as per their codes:**

```
db.Section.distinct("C_Code").sort();
```