COMP4332 Big Data Mining and Management
RMBI4310 Advanced Data Mining for Risk Management and Business Intelligence
(Spring Semester 2018)
Homework 2
Deadline: 4 May, 2018 1:30pm
(Please hand in the written part during lecture and the soft copy (i.e., the code) via Canvas.)
Full Mark: 100 Marks

Instructions:

1. You can use a coupon to waive any question you want and obtain full marks for this question.
2. You can waive at most one question in each assignment.
3. You can also answer the question you will waive. We will also mark it but will give full marks to this question.
4. The coupon is non-transferrable. That is, the coupon with a unique ID can be used only by the student who obtained it in class.
5. Please staple the coupon to the submitted written assignment.
6. Please write down the question no. you want to waive on the coupon.
7. No matter whether you use the coupon for a question about coding which requires a soft copy submission, just staple the coupon to the written part. The written part and the soft copy (i.e., the code) will be marked together.
8. You are required to use Python 3.6.3 (or above), MySQL 5.7 (or above), MongoDB 3.4.10 (or above) and Keras 2.1.1 (or above) (built on TensorFlow 1.5.0rc0 (or above)) in this assignment.
9. Please follow the submission guideline which could be found at the end of this assignment.
10. Please do the assignment by yourself so that you could learn something from the course. ☺ The assignment is not very difficult to be done if you revise the course materials well! ☺

Q1 [20 Marks] (Soft Copy)

[You are required to submit this question as a soft copy. Please see the submission instruction at the end of this assignment.]

Assume that a MySQL database server is running at a machine (e.g., your laptop and our tutor's server). Please use package "MySQL Connector" of Python for this question.

You are given a file called "student.txt" denoting a list of students' scores. The following shows an **example** of this file.

student.txt
4 1111,Raymond,80,70,95 2222,Peter,49,90,76 3333,Mary,23,100,20 4444,John,91,82,55

The first row contains a number (e.g., "4") denoting the total number of students. Each remaining row denotes the score information of a student. Each remaining row is represented by "<sid>,<sname>,<score1>,<score2>,<score3>" where <sid> denotes the ID of a student, <sname> denotes the name of a student, <score1> denotes the chemistry subject score of this student, <score2> denotes the history subject score of this student and <score3> denotes the computer subject score of this student. Note that each name does not contain character "," for the sake of simplicity (i.e., "WONG, Raymond" could not be found in this file since "WONG, Raymond" has character ",").

We are given another file called "config-MySQL.txt" storing the log-in information about the MySQL database server. The following shows an example of this file.

config-MySQL.txt
comp4332 datamining 127.0.0.1 hw2

The first row corresponds to the user name of the MySQL server. The second row corresponds to the password of the corresponding user. The third row corresponds to the IP address of the MySQL server. The fourth row corresponds to the database name of the MySQL server.

In this question, you are required to write a Python code called "q1.py" to do the following.

1. Write a function called "obtainStudentList" without any input argument but with a function return feature which returns a 2-dimensional array/list in order to read the list of students' scores from "student.txt". For example, if the array is L, then L[0][0] = "1111", L[0][1] = "Raymond", L[0][2] = "80", L[0][3] = "70" and L[0][4] = "95". Similarly, we have L[1][0] = "2222" and so on.
2. Write a function called "storeToMySQL" with an input argument of the 2-dimensional array (described in Point 1) but without any function output in order to perform the following.
    a. Read "config-MySQL.txt" to obtain the log-in information and store the information into 4 variables
    b. Create a table called "student" in the MySQL server with the following requirement (using the SQL statement in Python). After that, please print to the console with "Creating Table is Successful!".

Attribute Name	**Attribute Requirement**
sid	a string with a fixed length of 4 (which is a primary key)
sname	a string with a maximum length of 200
chemistry	an integer
history	an integer
computer	an integer

    c. Insert the information of each student stored in the 2-dimensional array into table "student" stored at the MySQL server (using the SQL statement in Python). After that, please print to the console with "Inserting Data is Successful!". The following shows the content of this table based on the given example.

sid	**sname**	**chemistry**	**history**	**computer**
1111	Raymond	80	70	95
2222	Peter	49	90	76
3333	Mary	23	100	20
4444	John	91	82	55

d. Delete all records in table "student" where each of these records has the computer score strictly smaller than 50 (using the SQL statement in Python). After that, please print to the console with "Deleting Records is Successful!" (even if no records has the computer score strictly smaller than 50). The following shows the content of this table based on the given example.

sid	sname	chemistry	history	computer
1111	Raymond	80	70	95
2222	Peter	49	90	76
4444	John	91	82	55

e. Obtain a list of records in table "student" in ascending order of "sname" (after the deletion operation) where each of these records has the chemistry score strictly greater than 50 (using the SQL statement in Python) and store this list in a 2-dimensional array/list (like the array/list described in Point 1). After that, please print it to the console with "Obtaining Records is Successful!" (even if no records are in the list). In the given example, the list contains 2 records only (i.e., the first record with "sid" = "4444" and the second record with "sid" = "1111").

f. Write the list to the output file called "output-MySQL.txt" based on the ordering stored in this list. After that, please print to the console with "Writing Records is Successful!" (even if no records are written) (In case that there are no records, we just need to write "0" in the first line of the output file). The following shows the output of "output-MySQL.txt" based on the given example.

output-MySQL.txt
2 4444,John,91,82,55 1111,Raymond,80,70,95

g. Drop table "student" (using the SQL statement in Python). After that, please print to the console with "Dropping Table is Successful!".

3. Call the above 2 functions in the above ordering in the main Python code (by passing some input/output arguments among these 2 functions) so that after we execute the whole Python code, "output-MySQL.txt" is generated.

Q2 [20 Marks] (Soft Copy)
[You are required to submit this question as a soft copy. Please see the submission instruction at the end of this assignment.]

Assume that a MongoDB database server is running at a machine (e.g., your laptop and our tutor's server). Please use package "PyMongo" of Python for this question.

You are given a file called "link.txt" storing a URL link of an HTML file. The following shows an **example** of this file.

link.txt
```
http://comp4332.com/hw2/q2-dataset-1.html
```

This HTML file stores a list of students' scores. The following shows an **example** of this HTML file.

HTML
```
<!DOCTYPE html>

<html>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>HW2-Q2-Dataset-1</title>
</head>

<body>

<h1>Students' Scores</h1>

<table width="900" border="1">
 <tr>
  <th scope="col">sid</th>
  <th scope="col">sname</th>
  <th scope="col">chemistry</th>
  <th scope="col">history</th>
  <th scope="col">computer</th>
 </tr>
 <tr>
  <td>1111</td>
  <td>Raymond</td>
  <td>80</td>
  <td>70</td>
  <td>95</td>
 </tr>
 <tr>
  <td>2222</td>
  <td>Peter</td>
  <td>49</td>
  <td>90</td>
  <td>76</td>
 </tr>
 <tr>
```

```
    <td>3333</td>
    <td>Mary</td>
    <td>23</td>
    <td>100</td>
    <td>20</td>
  </tr>
  <tr>
    <td>4444</td>
    <td>John</td>
    <td>91</td>
    <td>82</td>
    <td>55</td>
  </tr>
</table>
</body>
</html>
```

Since the score information of students is shown in the HTML file which is in a presentable manner, the meaning of each component/part in the above HTML file is straightforward. For example, based on the above HTML file, we know that there is a student with "sid" = "1111", "sname" = "Raymond", "chemistry" = "80", "history" = "70" and "computer" = "95".

Note that the above is an **example** of the HTML file. The other HTML file also follows the same format with the same set of attribute/field names (e.g., "sname" and "chemistry") but with different content (e.g., different values and different number of students).

We are given another file called "config-MongoDB.txt" storing the log-in information about the MongoDB database server. The following shows an **example** of this file.

config-MongoDB.txt
mongodb://localhost:27017 hw2

There are two rows in this file. The first row corresponds to the parameters used in MongoDB written in Python (which includes the "MongoDB" server (i.e., "mongodb"), the IP address of the MongoDB server (e.g., "localhost") and the port number of the MongoDB server (i.e., "27017")). The second row corresponds to the database name of the MongoDB server.

In this question, you are required to write a Python code called "q2.py" to do the following.

1. Write a function called "obtainStudentList" either without any input argument or with 2 input arguments called "self" and "response" but with a function return feature which returns an array/list of (Python's) dictionaries in order to perform a data crawling process on the HTML file in the URL specified in "link.html" (by using Python's package "scrapy" where the project name of "scrapy" is set to "hw2" (where "testCrawl" is used in our lecture notes as a project name) and the (unqiue) "name" variable used in "scrapy" in this Python code is set to "q2") and read the list of students' scores from the HTML file. For example, if the array is L, then L[0] is equal to the following dictionary.

   {'sid': '1111', 'sname': 'Raymond', 'chemistry': '80', 'history': '70', 'computer': '95'}

Similarly, L[1] is equal to the following dictionary.

{'sid': '2222', 'sname': 'Peter', 'chemistry': '49', 'history': '90', 'computer': '76'}
Note that all fields of each dictionary in the array store "strings".

2. Write a function called "storeToMongoDB" either with an input argument of the array of dictionaries (described in Point 1 of this question (i.e., Q2)) or with 2 input arguments called "self" and the array of dictionaries but without any function output in order to perform the following.
   a. Read "config-MongoDB.txt", obtain the log-in information and store the information into 2 variables
   b. Insert the information of each student stored in the array of dictionaries into a new collection called "student" stored at the MongoDB server where the field names used in MongoDB should be "sid" (storing a string), "sname" (storing a string), "chemistry" (storing an integer), "history" (storing an integer) and "computer" (storing an integer) (using the NoSQL statement in Python). After that, please print to the console with "Inserting Data is Successful!". The following shows the content of this collection based on the given example.

Collection student

{
sid: "1111",
sname: "Raymond",
chemistry: 80,
history: 70,
computer: 95
}
{
sid: "2222",
sname: "Peter",
chemistry: 49,
history: 90,
computer: 76
}
{
sid: "3333",
sname: "Mary",
chemistry: 23,
history: 100,
computer: 20
}
{
sid: "4444",
sname: "John",
chemistry: 91,
history: 82,
computer: 55
}

   c. Delete all records in collection "student" where each of these documents has the computer score strictly smaller than 50 (using the NoSQL statement in Python). After that, please print to the console with "Deleting Documents is Successful!" (even if no documents has the

computer score strictly smaller than 50). The following shows the content of this collection based on the given example.

Collection student

```
{
        sid: "1111",
        sname: "Raymond",
        chemistry: 80,
        history: 70,
        computer: 95
}
{
        sid: "2222",
        sname: "Peter",
        chemistry: 49,
        history: 90,
        computer: 76
}
{
        sid: "4444",
        sname: "John",
        chemistry: 91,
        history: 82,
        computer: 55
}
```

d. Obtain a list of documents in collection "student" in ascending order of "sname" (after the deletion operation) where each of these documents has the chemistry score strictly greater than 50 (using the NoSQL statement in Python) and store this list in an array/list of dictionaries (like the array/list described in Point 1 of this question). After that, please print it to the console with "Obtaining Documents is Successful!" (even if no documents are in the list). In the given example, the list contains 2 documents only (i.e., the first document with "sid" = "4444" and the second document with "sid" = "1111").

e. Write the list to the output file called "output-MongoDB.txt" based on the ordering stored in this list. After that, please print to the console with "Writing Documents is Successful!" (even if no documents are written). (In case that there are no documents, we just need to write "0" in the first line of the output file). The following shows the output of "output-MongoDB.txt" based on the given example.

output-MongoDB.txt

```
2
4444,John,91,82,55
1111,Raymond,80,70,95
```

f. Drop collection "student" (using the NoSQL statement in Python). After that, please print to the console with "Dropping Collection is Successful!".

3. Call the above 2 functions in the above ordering in the main Python code (by passing some input/output arguments among these 2 functions) so that after we execute the whole Python code, "output-MongoDB.txt" is generated.

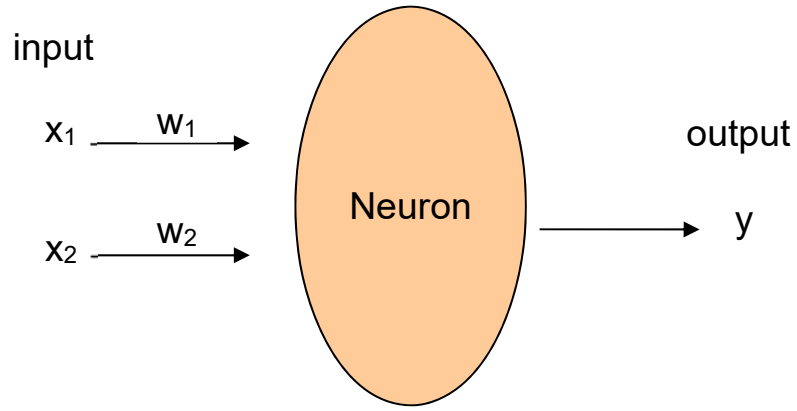
Q3 [20 Marks] (Written Copy)
[You are required to submit this question as a written copy. Please submit it during lecture.]

(a) Consider the following table with three attributes where "No. of Computers" and "No. of Phones" are input attributes and "Buy_NintendoLabo" is the target attribute. Each record corresponds to a customer.

No. of Computers	No. of Phones	Buy_NintendoLabo
2	0	No
0	2	No
4	2	Yes
2	4	Yes

(i) Rewrite the above table such that values "Yes" and "No" in attribute "Buy_NintendoLabo" are mapped to values 1 and 0, respectively.

(ii) Consider a neural network containing a single neuron where x_1 = "No. of Computers", x_2 = "No. of Phones" and y = "Buy_NintendoLabo".



Initially, we set the values of w_1, w_2 and b to be 0.3 where b is a bias value in the neuron.

Suppose that the learning rate is denoted by α. Let $\alpha = 0.6$.
Suppose that we adopt the sigmoid function as an activation function.
Please try to train the neural network with five instances by the following inputs in the given sequence.

1. $(x_1, x_2) = (2, 0)$
2. $(x_1, x_2) = (0, 2)$
3. $(x_1, x_2) = (4, 2)$
4. $(x_1, x_2) = (2, 4)$
5. $(x_1, x_2) = (2, 0)$

What are the final values of w_1, w_2 and b after these five instances are processed?

(b) What is the major disadvantage of the traditional neural network model compared with the recurrent neural network model?

Q4 [20 Marks] (Soft Copy)
[You are required to submit this question as a soft copy. Please see the submission instruction at the end of this assignment.]

You are given a file called "training.csv" denoting a list of records with 2 input attributes and 1 target attribute for training. The following shows an **example** of this file.

training.csv
5,5,Yes
8,3,Yes
9,6,Yes
6,8,Yes
2,3,No
1,2,No
3,3,No
2,1,No

Each row is represented by "<attribute1>,<attribute2>,<target>" where <attribute1> denotes the first input attribute of a record, <attribute2> denotes the second input attribute of a record and <target> denotes the target attribute of a record.

You are also given a file called "new.csv" denoting a list of records with 2 input attributes for prediction. The following shows an **example** of this file.

new.csv
2,4
7,6
8,5
3,2

In this question, you are required to write a Python code called "q4.py" to do the following.

1. Write a function called "trainData" without any input argument but with a function return feature which returns a neural network model with the following requirement.
    a. Read the records in "training.csv" using function "loadtxt" of the package "numpy"
    b. Transform (or map) "Yes" to 1 and "No" to 0 in the target attribute of all records
    c. Perform a training process on these records using package "Keras" of Python where the parameters of this training are shown as follows.
        i. Seed Setting
            A. Set the seed to "4332" in function "random.seed" of the package "numpy"
        ii. Neural Network Model Parameter
            A. The neural network has the following 3 layers.
                - The first layer (i.e., the input layer) contains the total number of input attributes
                - The second layer contains 4 neurons using the rectifier function as an activation function
                - The third layer (i.e., the output layer) contains only 1 neuron using the sigmoid function as an activation function
                - For every two adjacent layers, the neurons in one layer are fully connected with other neurons in the other layer.

B. Optimization method = "adam"
C. Error function = "binary cross entropy"

iii. Training Parameter
A. No. of epochs = 1500
B. Batch size = 4

iv. Evaluation
A. Measurement = "accuracy"
B. Parameter “validation_split” = 0.2

2. Write a function called “predictData” with an input argument of the neural network model (described in Point 1) but without any function output in order to perform the following.
   a. Read the records in “new.csv” using function “loadtxt” of the package “numpy”
   b. Predict the probability that the target attribute of each record in “new.csv” is “Yes” and the corresponding predicted categorical value of this target attribute. That is, if the probability is at least 0.5, we set the predicted categorical value as “Yes”. Otherwise, we set the predicted categorical value as “No”.
   c. Write the predicted probabilities (shown/rounded up to 10 decimal places) and the predicated categorical values of all records to the output file called “output-NN.csv”. The following shows the example of this output file.

output-NN.csv
0.4402543306,No 0.9776688814,Yes 0.9857090712,Yes 0.4912259281,No

   Each row is represented by “<probability>,<categorical>” where <probability> denotes the probability that the target attribute of a record in “new.csv” is “Yes” and <categorical> is the corresponding categorical value of this target attribute.
3. Call the above 2 functions in the above ordering in the main Python code (by passing some input/output arguments among these 2 functions) so that after we execute the whole Python code, “output-NN.csv” is generated.

Q5 [20 Marks] (Written Copy)
[You are required to submit this question as a written copy. Please submit it during lecture.]

We are given two data points with 2 different timestamps.
At the timestamp t = 1, we have a data point (x_1, x_2, y) where (x_1, x_2) = (0.3, 0.6) and y = 0.2.
At the timestamp t = 2, we have a data point (x_1, x_2, y) where (x_1, x_2) = (0.1, 1.0) and y = 0.4.
Here, x_1 and x_2 are 2 input variables. y is the output variable.

(a) Consider the traditional LSTM model. Initially, we have the following internal weight vectors and bias variables as follows.

$$W_f = \begin{pmatrix} 0.8 \\ 0.4 \\ 0.1 \end{pmatrix} \quad b_f = 0.2$$

$$W_i = \begin{pmatrix} 0.9 \\ 0.8 \\ 0.7 \end{pmatrix} \quad b_i = 0.5$$

$$W_a = \begin{pmatrix} 0.4 \\ 0.2 \\ 0.1 \end{pmatrix} \quad b_a = 0.3$$

$$W_o = \begin{pmatrix} 0.6 \\ 0.4 \\ 0.1 \end{pmatrix} \quad b_o = 0.2$$

In the model, we have the following status variables. For each t = 1, 2, ….

1. forget gate variable f_t
2. input gate variable i_t
3. input activation variable a_t
4. internal state variable s_t
5. output gate variable o_t
6. final output variable y_t

Suppose that $y_0 = 0$ and $s_0 = 0$.

Consider the input forward propagation step only.

(i) What are the values of the above status variables when t = 1 and when t = 2? Please show each answer up to 4 decimal places.

(ii) What are the errors of the final output variables when t = 1 and when t = 2? Please show each answer up to 4 decimal places.

(b) Consider the GRU model. Initially, we have the following internal weight vectors and bias variables as follows.

$$W_r = \begin{pmatrix} 0.3 \\ 0.2 \\ 0.1 \end{pmatrix} \quad b_r = 0.5$$

$$W_a = \begin{pmatrix} 0.4 \\ 0.3 \\ 0.1 \end{pmatrix} \quad b_a = 0.1$$

$$W_u = \begin{pmatrix} 0.4 \\ 0.2 \\ 0.1 \end{pmatrix} \quad b_u = 0.1$$

In the model, we have the following status variables. For each t = 1, 2, ….

1. reset gate variable r_t
2. input activation variable a_t
3. update gate variable u_t
4. final output variable y_t

Suppose that $y_0 = 0$.

Consider the input forward propagation step only.

(i) What are the values of the above status variables when t = 1 and when t = 2? Please show each answer up to 4 decimal places.

(ii) What are the errors of the final output variables when t = 1 and when t = 2? Please show each answer up to 4 decimal places.

(c) What are the advantages and the disadvantages of the GRU model compared with the traditional LSTM model?

Submission Guideline:

1. For the written part, please submit it during the lecture.
2. For the soft copy part (i.e., code/script), please submit it via Canvas with the following requirement.
    - Please zip the following files into a single file and submit as a ZIP file. The file name is "XXX.zip" where "XXX" is your student ID. If your student ID is 20123456, then your file name is "20123456.zip".
        i. The soft copy for Q1 (i.e., "q1.py")
        ii. The soft copy for Q2 (i.e., "q2.py")
        iii. The soft copy for Q4 (i.e., "q4.py")
        iv. readme file (readme.txt) which contains
            1. your information (i.e., student ID and student name)
            2. file list
            3. file description
            4. which directory that "q2.py" should be put (e.g., "<working directory>/hw2/hw2/spiders" where <working directory> is the working directory.)
            5. method of execution (e.g., "python q1.py" and "scrapy crawl q2")
            6. known bugs of your programs/scripts (if any)