Data- X Homework 04: **Due Date: 23rd April, 2018**

Created by:
Part1 and Part 2: Sana Iqbal

Part 1: Entropy

1. The dataset shown below represents bank customers with 3 features and the label corresponding to each customer identifies whether they've defaulted or not.

   Description:

   a. **HasJob:** Binary value, equal to 0 when a person has no job and 1 otherwise.
   b. **HasFamily:** Binary value, equal to 0 when a person has no family and 1 otherwise.
   c. IsAbove30years: Binary value, equal to 0 when a person's age is 30 or below and 1 otherwise.
   d. Defaulter is also a binary valued label which is equal to 1 if a person is a defaulter and 0 otherwise.

   Use this dataset to **identify the best feature to do the first split** in a binary decision tree, so as to maximize the information gain in the next split. Show your calculations.

HasJob	HasFamily	IsAbove30years	Defaulter
1	1	1	0
1	1	1	0
1	0	1	0
0	1	0	0
0	0	1	1
0	1	0	1
1	0	1	1
1	0	1	1

2. Given a signal of three symbols **S=(A, B, C)** and P(A)=0.7, P(B)=0.2, P(C)=0.1, What is the entropy of S? What does it mean according to the *Source coding Theorem*?

Part 2: Natural Language Processing:

1. What is the difference between a Bag Of words Model in NLP and a Word2vec Model, discuss advantages of one over the other? (in 200 words)
2. What is a word vector? What is a word Embedding? On what factors does the word embedding of a word depend (explain it from a NLP perspective)? (in 100 words)
3. What is a corpus in NLP? How is the vocabulary of a model different from the corpus? (in 50 words)
4. Train a word2vec model on the corpus consisting of the text in the novel *Pride and Prejudice*.
    a. DATA: Here is the csv file of the list of sentences in the novel. Here is the Wikipedia description of the novel.
    b. PREPROCESS and MODEL: Preprocess the sentences to generate a data set with cleaned sentences (you can use the same cleaning procedure as shown in class). Use the cleaned sentences with word tokens as a corpus for training your Word2vec model, using the gensim package (you can use any hyperparameter setting).
    c. VISUALIZATION: Once you train the model, visualize the PCA decomposition of the word vectors.
    d. EVALUATION of the Trained Model: Come up with five intrinsic evaluations of your trained model using methods like *most_similar( ), similarity( ), doesnt_match( )*, etc. For example one evaluation can be - *model.similarity('elizabeth','girl')*.

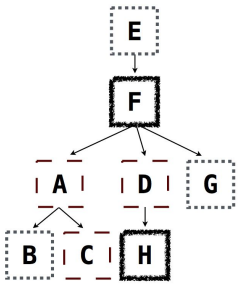
Note: Since the corpus is very small, be aware that your model vocabulary is very limited. You can increase the number of training iterations to improve embedding quality.

Part 3: **Simple SQL Queries**
(*Report your queries and what they return.)*

Let's play with Dogs.. & SQL! :)

First create a table called parents. It has two columns: 'parent' and 'child'. The first column indicates the parent of the child in the second column. We will use a new form of `CREATE TABLE` expression to produce this table.

```
CREATE TABLE parents (
     parent VARCHAR(20),
     child VARCHAR(20));
INSERT INTO parents (parent, child)
  VALUES ("abraham",     "barack")      UNION
  VALUES ("abraham",    "clinton")   UNION
  VALUES ("delano",     "herbert")   UNION
  VALUES  ("fillmore",   "abraham")   UNION
  VALUES ("fillmore",   "delano")    UNION
  VALUES ("fillmore",   "grover")    UNION
  VALUES ("eisenhower", "fillmore");
```



DOG FAMILY TREE

A: abraham
B: barack
C: clinton
D: delano
E: eisenhower
F: fillmore
G: grover

1. Simple SELECTS (on the parents table)

1. SELECT all records in the table.
2. SELECT child and parent, where abraham is the parent.
3. SELECT all children that have an 'e' in their name (hint: use LIKE and '%e%').
4. SELECT all unique parents (use SELECT DISTINCT) and order them by name, descending order (i.e. fillmore first)
5. SELECT all dogs that are siblings (one-to-one relations). Only show a sibling pair once. To do this you need to select two times from the parents table.

2. JOINS

Create a new table called dogs, which indicates the fur type of every dog. In the dog family tree image above:

- long haired dogs = dashed box
- curly haired dogs = black fluffy box
- short haired dogs = dotted box

You can create the table by running (this is an alternative way to create a table that is a

little bit more concise than the method shown during the lecture. Here SQL will figure out the datatypes of the columns):

```
CREATE TABLE dogs AS
  SELECT "abraham" AS name, "long" AS fur UNION
  SELECT "barack",          "short"       UNION
  SELECT "clinton",         "long"        UNION
  SELECT "delano",          "long"        UNION
  SELECT "eisenhower",      "short"       UNION
  SELECT "fillmore",        "curly"       UNION
  SELECT "grover",          "short"       UNION
  SELECT "herbert",         "curly";
```

1. COUNT the number of short haired dogs
2. JOIN tables parents and dogs and SELECT the parents of curly dogs.
2. JOIN tables parents and dogs, and SELECT the parents and children that have the same fur type. Only show them once.

3. Aggregate functions, numerical logic and grouping

Create a new table with many different animals. The table includes the animal's kind, number of legs and weight. Create it by running:

```
create table animals as
 select "dog" as kind, 4 as legs, 20 as weight union
 select "cat" , 4 , 10 union
 select "ferret" , 4 , 10 union
 select "parrot" , 2 , 6 union
 select "penguin" , 2 , 10 union
select "t-rex" , 2 , 12000;
```

1. SELECT the animal with the minimum weight. Display kind and min_weight.
2. Use the aggregate function AVG to display a table with the average number of legs and the average weight.
3. SELECT the animal kind(s) that have more than two legs, but weighs less than 20. Display kind, weight, legs.
4. SELECT the average weight for all the animals with 2 legs and the animals with 4 legs (by using GROUP BY).

SUBMISSION:

Submit **a single pdf file (***without any code, you can use images or screen shots of graphs and results or any hand calculations* **)** and **a link to your github repository** containing all the code.

Your pdf file should contain:

For Part 1 :

1. Report your results, calculations and answers for questions in part-1.

For Part 2:

1. Report your answers for questions 1, 2 and 3 in part 2.
2. For question 4 of part 2:
    a. Write a short summary describing your preprocessing and modeling technique. (300 words only)
    b. Report the vocabulary count, embedding size, number of training iterations in your modelling.
    c. Report and explain your observation from the visualization of word vectors.
    d. Report any five intrinsic evaluation results that you performed using your model.

For Part 3:

1. Report your **queries as well as answers** for questions 1, 2 and 3 in part 3.

Good luck!