UC San Diego Extension

Cloud Services for Machine Learning

Summer 2020

Homework#3

Date Given: July 13, 2020 Due Date: July 19, 2020

==

Classification Using GCP: **There are 2 problems in this assignment.**

==

The file “HW03 Jobs Data.csv” contains tabular student’s job data. There are 4 predictor variables.

- Height in inches
- Major
- Points (on a scale from 1 to 100)
- Weight in lbs

There is one response variable.

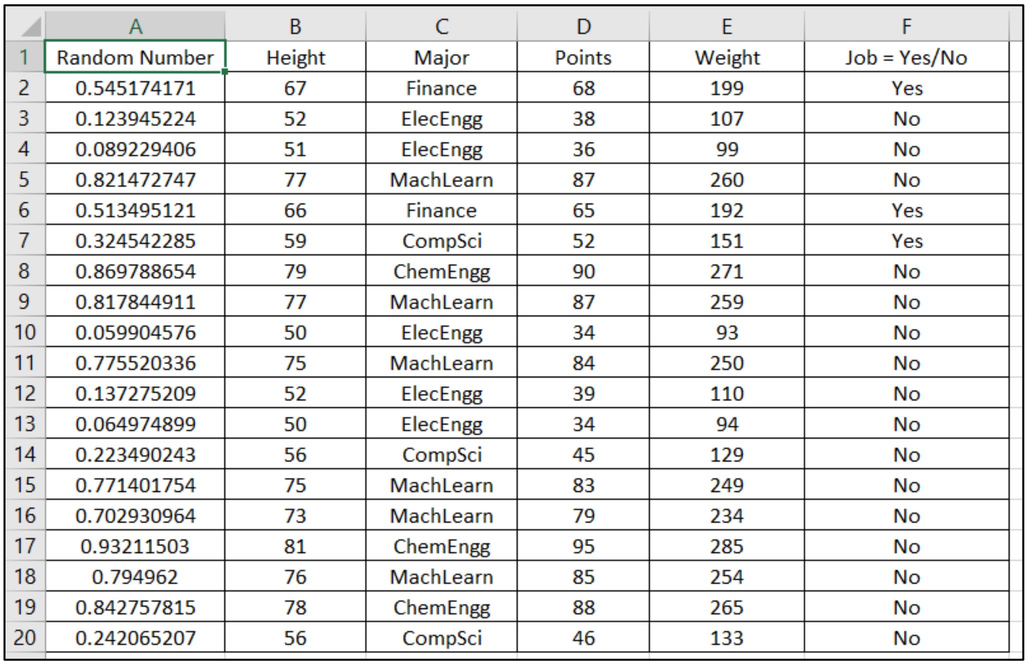
- Job (Categorical): Yes or No

This data was artificially generated using a script which ran on Google’s Spreadsheet application “Sheets”. The source code of the Sheets script is displayed at the end (Appendix) of this assignment.

The relationship between ‘Job’ and ‘Major’ + ‘Points’ is as follows.

If (Major = CompSci or Medicine or Finance) AND Points > 50
Job = Yes

This relationship between ‘Job’ and ‘’Major’ + ’Points’ is non-linear. The ‘Height’ and the ‘Weight’ data are purely noise. There are 1,500 observations. The following table displays the first 20 observations.



	A	B	C	D	E	F
1	Random Number	Height	Major	Points	Weight	Job = Yes/No
2	0.545174171	67	Finance	68	199	Yes
3	0.123945224	52	ElecEngg	38	107	No
4	0.089229406	51	ElecEngg	36	99	No
5	0.821472747	77	MachLearn	87	260	No
6	0.513495121	66	Finance	65	192	Yes
7	0.324542285	59	CompSci	52	151	Yes
8	0.869788654	79	ChemEngg	90	271	No
9	0.817844911	77	MachLearn	87	259	No
10	0.059904576	50	ElecEngg	34	93	No
11	0.775520336	75	MachLearn	84	250	No
12	0.137275209	52	ElecEngg	39	110	No
13	0.064974899	50	ElecEngg	34	94	No
14	0.223490243	56	CompSci	45	129	No
15	0.771401754	75	MachLearn	83	249	No
16	0.702930964	73	MachLearn	79	234	No
17	0.93211503	81	ChemEngg	95	285	No
18	0.794962	76	MachLearn	85	254	No
19	0.842757815	78	ChemEngg	88	265	No
20	0.242065207	56	CompSci	46	133	No

Problem#1

Build a **classification** Machine Learning model (Neural Network) using Google Cloud Platform (GCP) with the data in the ‘HW03 Jobs Data.csv’ file. Ignore the first column ‘Random Number’ while building the model.

This model does not know the TRUE relationship between ‘Job’ and ‘Major’ + ‘Points’. It must learn this relationship only by analyzing the data. To challenge the ML model, noise data of ‘Height’ and ‘Weight’ columns have been added to the dataset.

The procedure to build a **classification** model on GCP is as follows (same as regression model).

1. GCP/Storage
   a. Create a Bucket in GCP
   b. Region: us-central1(Iowa)
   c. Upload Data file in that bucket
2. GCP/Table/Dataset
   a. Import data in a GCP Dataset from the bucket: Takes time
3. GCP/Table/Model
   a. Train the Model
      i. Select Target Variable + Budget: Takes time
   b. Evaluate the Model
   c. Test & Use: Deploy the Model: Takes time
      i. Prediction

Evaluate the model after the model is trained on GCP. Which variable is the most important for prediction? Copy the ‘Feature Importance’ plot generated by GCP in your answer document. Does your model built on GCP able to capture the TRUE relationship between ‘Job’ and ‘Major’ + ’Points’ variables?

Using the **classification** model, you have built on GCP, **classify** the job status of the following 4 students. Compute the probability of the response variable ‘Job’. Assume the cut-off probability as 0.5. If the probability is greater than 0.5, classification of the ‘Job’ variable is ‘Yes’.

	Height in inches	Major	Points	Weight in lbs	Logical Value of Job variable	Probability computed by GCP Job = Yes	Probability computed by GCP Job = No	**Classification** Yes/No
1	75	CompSci	85	220	Yes	?	?	?
2	82	CompSci	49	200	50/50	?	?	?
3	62	MachLearn	81	151	No	?	?	?
4	67	Finance	51	95	50/50	?	?	?

Problem#2: Can we build a kNN (k Nearest Neighbor) model for this dataset used in Problem#1?
If no, why not?

==

Building a classification model on GCP will cost a certain amount. Please check the GCP charges on your account before and after you complete this assignment. Make sure you do not deplete the $300 credit you have on your account.

Appendix:

The Google's "Sheets" script that generated 1,500 observations is displayed here. This script is also uploaded on Canvas. Feel free to run this script to generate new dataset.

The following procedure is needed to run this script.

- Visit the URL: google.com/sheets
- Menu command: Tools/Script Editor
- Copy the script here
- Menu command: File/Save
- Run the script: First permission is needed from the Google account holder. Data will be generated.
- Download the data from the "Sheets" spreadsheet to a .csv file.

===

```
function addData()
{
  var sheet = SpreadsheetApp.getActiveSheet();

   /// First row: Generate the Heading Data
   ///
  var newRow = [];
  newRow.push("Random Number");
  newRow.push("Height");
  newRow.push("Major");
  newRow.push("Points");
  newRow.push("Weight");
  newRow.push("Job = Yes/No");
  sheet.appendRow(newRow);

   /// New genearate the 1,500 rows of data
   ///
  for (var i = 0 ; i < 1500 ; i++)
  {
    /// Generate a random Number between 0 and 1 - Column 0 /////////
    ///
    var newRow = [];
    var randNumber = Math.random();
    newRow.push(randNumber);

    /// Height - Column 1 /////////////////////////////
    /// minimum height = 48 inches (4 feet * 12 = 48 inches)
    /// maximum hiight = 84 inches (7 feet * 12 = 84 inches)
    ///
    var minHeight = 48
    var maxHeight = 84
    var number = Math.floor(randNumber*(maxHeight - minHeight) + minHeight);
    newRow.push(number);

    /// Major - Column 2 ///////////////////////////
    ///
    var columnB = ["ElecEngg", "CompSci","Medicine", "Finance",
"MachLearn","ChemEngg"];
    var pickColumnB = Math.floor(randNumber*6);
    newRow.push(columnB[pickColumnB]);

    /// Points - Column 3 /////////////////////////////
    /// minimum Points = 30
    /// maximum Points = 100
    ///
    var minPoints = 30
    var maxPoints = 100
    var number = Math.floor(randNumber*(maxPoints - minPoints) + minPoints);
    newRow.push(number);

    /// Weight - Column 4 /////////////////////////////
    /// minimum Weight = 80
    /// maximum Weight = 300
    ///
    var minWeight = 80
    var maxWeight = 300
    var number = Math.floor(randNumber*(maxWeight - minWeight) + minWeight);
    newRow.push(number);

    /// If Points(Column 3) > 50 AND Major(Column 2) = 'CompSci' OR
'Medicine' OR 'Finance' ////
    /// Column(5): Job = Yes, Else Job = No /////////////
    ///
    if (newRow[3] > 50 && /CompSci|Medicine|Finance/i.test(newRow[2])) {
      newRow.push("Yes")
    } else {
      newRow.push("No") }
    ///////////////////////////////////////////

    sheet.appendRow(newRow);

  }
}
```