Manufacturing Data Science 製造數據科學

Assignment 3

Due Date: Nov. 11, 2022

Please solve the following questions and justify your answer. Show all your analysis result including equation/calculation or Python code in your report. **Upload your "zip" file including MSWord/PDF report and Python code with 檔名: MDS_Assignment3_ID_Name.zip" to NTU COOL** by due. The late submission is not allowed.

1. **(40%)** Decision Tree Algorithms

Use Python to solve the problem. The SECOM (Semiconductor Manufacturing) dataset, consists of manufacturing operation data and the semiconductor quality data. It contains 1567 observations taken from a wafer fabrication production line. Each observation is a vector of 590 sensor measurements plus a label of pass/fail test. Also, there are only 104 fail cases which are labeled as positive (encoded as 1: bad), whereas much larger amount of examples pass the test and are labeled as negative (encoded as -1: good). The dataset can be collected from UCI machine learning repository.

Data Source: McCann & Johnston (2008), https://archive.ics.uci.edu/ml/datasets/SECOM.

The data is attached in the file **MDS_Assignment3_SECOM.xlsx**.

Hint: you may refer https://rpubs.com/jeff_datascience/Semiconductor_Manufacturing

(a) **(5%)** Construct a data science framework and show the data summary

(b) **(5%)** What is the problem about the dataset? Any identical column? Any redundant column? Any missing value? How to handle these issues?

(c) **(5%)** After data preprocessing, based on the **prepared dataset**, use the classification and regression tree (CART) to analyze the prepared dataset. Show the classification results by 10-fold cross validation with several metrics (eg. accuracy, area under ROC curve (AUC), and F1-score), and also list the hyperparameters you adjust.

(d) **(5%)** Suggest a method to address the data imbalance issue. Build a new balanced dataset. (hint: undersampling or oversampling)

(e) **(5%)** Based on the **balanced dataset**, use the classification and regression tree (CART) to analyze the balanced dataset. Show the classification results by 10-fold cross validation with several metrics (eg. accuracy, area under ROC curve (AUC), and F1-score), and also list the hyperparameters you adjust.

(f) **(5%)** Give a comparison between (c) and (e). Any suggestion or insight?

(g) **(5%)** Use "Random Forest" to solve both prepared dataset and balanced dataset, respectively. Give a comparison and provide your insight.

(h) **(5%)** Use "Gradient Boosting Decision Tree (GBDT)" to solve both prepared dataset and balanced dataset, respectively. Give a comparison and provide your insight.

2. (40%) Feature Selection

在 Kaggle 開放數據中包含了一個挖礦製程的浮選廠數據(a flotation plant in a mining process，https://www.kaggle.com/edumagalhaes/quality-prediction-in-a-mining-process)，一共包含 24 個特徵。硬件傳感器，如溫度、pH 值、流量、密度和所有連續過程變量，每 20 秒收集一次數據。其他某些特徵每小時採樣一次。品質特徵，如二氧化矽含量百分比、鐵礦石含量百分比等，是通過實驗室分析進行的品質測量。每 15 分鐘在現場收集一次鐵礦漿樣本。這些樣本被送到實驗室進行分析，每兩小時提供一次品質分析結果。據上述描述，將因子「% Silica Concentrate 含二氧化矽濃度百分比」作為應變量(y)且其他所有因子（除日期和 % Iron Concentrate 含鐵濃度百分比外）均為自變量(x)，如何找出影響 y 的重要因子呢？若能建構預測模型，提前幾個小時來預測二氧化矽濃度百分比，這將幫助工程師以預測和優化的方式來減少可能進入後製程的鐵百分比。試著參考網路資源學習並撰寫程式，使用此數據回答下列問題。

The data is attached in the file **MDS_Assignment3_MiningProcess.zip**.

(a) 試使用線性迴歸以最小平方法估計迴歸係數，並說明重要變數(例如排序 p-value 或 t 統計量)。
(b) 試使用逐步迴歸找出重要變數。
(c) 試比較(a)與(b)的結果是否一致？有何不同？
(d) 試使用脊迴歸挑選重要變數。
(e) 試使用套索迴歸或適應性套索迴歸挑選重要變數。
(f) 試比較(d)與(e)的結果是否一致？有何不同？
(g) 在特徵中那些特徵彼此之間高相關？若以線性迴歸預測，請問是否有共線性的問題？
(h) 是否可用含鐵濃度百分比來建模預測含二氧化矽濃度百分比？為什麼？可能有什麼潛在問題？如何解決？

3. (20%) Deep Learning

Use Python to build up backpropagation network (BPN) (or convolutional neural network, CNN) for "**handwritten digit recognition**". Data set (**MDS_Assignment3_DRtraining.xlsx**) is collected from the Semeion Research Center of Sciences of Communication. It is available in text form and contains 1593 handwritten digits from 80 persons. The images are 16 × 16 pixels square box and in black and white format. (teacher took 15 samples for validation and you only see 1578 samples)

Each record represents a handwritten digit, originally scanned with a resolution of 256 grays scale. Each pixel of each original scanned image was first stretched, and after scaled between 0 and 1 (setting to 0 every pixel whose value was under a fixed threshold value 127 of the grey scale (127 included), and setting to 1 each pixel whose original value in the grey scale was over 127).

We name variable "Pixel001" to "Pixel256". After the array of pixels there is information what

digit the image depicts, i.e., “Target0” to “Target9”. We use 256 nodes in input layer representing “Pixel” binary variable, and 10 nodes for output layer representing “Target” binary variable, i.e., 0 to 9.

The pixel format is shown as following figure.

Pixel															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176
177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208
209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256

The digit “3”, for example, is shown as following figure.

1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1
0	0	1	1	1	1	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0

Any question, you can google it (keyword: digit recognition) or refer to the following linkage.
https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/
If you would like to use Tensorflow, Keras, numpy, and pillow, you may refer to
https://medium.com/analytics-vidhya/deep-learning-project-handwritten-digit-recognition-using-python-26da7ed11d1c

(a) **(15%)** For BPN (or CNN if you prefer), try to investigate the effects of changing "**PARAMETERS**" such as learning rates, momentum, # of hidden/convolutional layers, dropout rate, etc. Show the numerical results and "**DIAGRAM**" from different perspectives (e.g., MSE/accuracy, F1-score, convergence time, error of training data, error of testing data, etc.). Please show all your work in detail, in particular, you "MAY" need to design your **experiments with different parameters** systematically.

(b) **(5%)** Please predict the digit No.1579 to No.1593 (data source: **MDS_Assignment3_DRpredict.xlsx**) using your **best** established BPN/CNN model in (a) and fill out the following table.

No.	Digit Number (0-9) you predict
1579	?
1580	?
1581	?
1582	?
1583	?
1584	?
1585	?
1586	?
1587	?
1588	?
1589	?
1590	?
1591	?
1592	?
1593	?

Note

1. Show all your work in detail. **Innovative** idea is encouraged.
2. If your answer refers to any external source, please "must" give an academic citation. Any "plagiarism" is not allowed.