GTx: ISYE6501x - Homework 6

Muh Alif Ahsanul Islam

June 25, 2019

Question 13.2

In this problem you, can simulate a simplified airport security system at a busy airport. Passengers arrive according to a Poisson distribution with λ1 = 5 per minute (i.e., mean inter-arrival rate μ1 = 0.2 minutes) to the ID/boarding-pass check queue, where there are several servers who each have exponential service time with mean rate μ2 = 0.75 minutes. After that, the passengers are assigned to the shortest of the several personal-check queues, where they go through the personal scanner (time is uniformly distributed between 0.5 minutes and 1 minute).

Use the Arena software (PC users) or Python with SimPy (PC or Mac users) to build a simulation of the system, and then vary the number of ID/boarding-pass checkers and personal-check queues to determine how many are needed to keep average wait times below 15 minutes. [If you're using SimPy, or if you have access to a non-student version of Arena, you can use λ1 = 50 to simulate a busier airport.]

Answer:

I used SimPy in Python to solve this problem.

The results are:

No	Passenger arrival (λ1)	Number of boarding pass checker service	Number of personal check queue	Average waiting time (min)
1	5	5	5	1.93
2	5	4	4	5.3
3	5	3	3	154.3
4	5	4	3	145.1
5	5	3	4	147.5
6	5	3	2	338.4
7	5	3	1	531.5
8	5	5	4	3.09
9	5	5	3	150.29
10	5	5	2	337.23
11	5	5	1	529.3
12	5	4	5	4.7
13	5	3	5	146.3
14	5	2	5	344
15	5	1	5	530.92

The best combination for λ1=5 that still satisfy the requirement (15 minutes average waiting time) is 4 boarding pass check queues and 4 personal check queues.

For busier airport:

No	Passenger arrival (λ1)	Number of boarding pass checker service	Number of personal check queue	Average waiting time (min)
1	50	50	50	1.5
2	50	40	40	1.79
3	50	30	30	148.4
4	50	40	30	145.7
5	50	40	35	47.95
6	50	40	38	2.37
7	50	40	37	9.5
8	50	40	36	30.23
9	50	41	36	31.7
10	50	42	36	31.7
11	50	43	36	30.65
12	50	47	36	30.86
13	50	47	37	12.98
14	50	46	37	12.68
15	50	50	37	11.45
16	50	40	37	11.46
17	50	30	37	143.2
18	50	35	37	47.94
19	50	36	37	31.98
20	50	38	37	14.05
21	50	37	37	12.57

The best combination for λ1=50 that still satisfy the requirement (15 minutes average waiting time) is 38 boarding pass check queues and 37 personal check queues.

To make better decision, it is important to know the cost of adding boarding pass check and personal check queues so we can select the combination with lowest cost and still satisfy 15 minutes average waiting time. When doing optimization it is also important to consider the robustness of our prescriptive solution.

Question 14.1

The breast cancer data set breast-cancer-wisconsin.data.txt from has missing values.

1. Use the mean/mode imputation method to impute values for the missing data.

2. Use regression to impute values for the missing data.

3. Use regression with perturbation to impute values for the missing data.

4. (Optional) Compare the results and quality of classification models (e.g., SVM, KNN) build using

   (1) The data sets from questions 1, 2, and 3;

   (2) The data that remains after data points with missing values are removed; and

   (3) The data set when a binary variable is introduced to indicate missing values.

The code for this questions is in 14_1.R file.

Data with missing data are below:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
24	1057013	8	4	5	1	2	?	7	3	1	4
41	1096800	6	6	6	9	6	?	7	8	1	2
140	1183246	1	1	1	1	1	?	2	1	1	2
146	1184840	1	1	3	1	2	?	2	1	1	2
159	1193683	1	1	2	1	3	?	1	1	1	2
165	1197510	5	1	1	1	2	?	3	1	1	2
236	1241232	3	1	4	1	2	?	3	1	1	2
250	169356	3	1	1	1	2	?	3	1	1	2
276	432809	3	1	3	1	2	?	2	1	1	2
293	563649	8	8	8	1	2	?	6	10	1	4
295	606140	1	1	1	1	2	?	2	1	1	2
298	61634	5	4	3	1	2	?	2	3	1	2
316	704168	4	6	5	6	7	?	4	9	1	2
322	733639	3	1	1	1	2	?	3	1	1	2
412	1238464	1	1	1	1	1	?	2	1	1	2
618	1057067	1	1	1	1	1	?	1	1	1	2

There are 2.2% missing value. This is small proportion so I think it is okay to do imputation.

1. Mean/mode imputation
   a. Mean imputation: cancer_df$V7[is.na(cancer_df$V7)] <- mean(cancer_df$V7, na.rm = TRUE)

```
> cancer_df[missing_rows, ]
         V1 V2 V3 V4 V5 V6       V7 V8 V9 V10 V11
24  1057013  8  4  5  1  2 3.544656  7  3   1   4
41  1096800  6  6  6  9  6 3.544656  7  8   1   2
140 1183246  1  1  1  1  1 3.544656  2  1   1   2
146 1184840  1  1  3  1  2 3.544656  2  1   1   2
159 1193683  1  1  2  1  3 3.544656  1  1   1   2
165 1197510  5  1  1  1  2 3.544656  3  1   1   2
236 1241232  3  1  4  1  2 3.544656  3  1   1   2
250  169356  3  1  1  1  2 3.544656  3  1   1   2
276  432809  3  1  3  1  2 3.544656  2  1   1   2
293  563649  8  8  8  1  2 3.544656  6 10   1   4
295  606140  1  1  1  1  2 3.544656  2  1   1   2
298   61634  5  4  3  1  2 3.544656  2  3   1   2
316  704168  4  6  5  6  7 3.544656  4  9   1   2
322  733639  3  1  1  1  2 3.544656  3  1   1   2
412 1238464  1  1  1  1  1 3.544656  2  1   1   2
618 1057067  1  1  1  1  1 3.544656  1  1   1   2
```

b. Mode imputation: cancer_df$V7[is.na(cancer_df$V7)] <- Mode(cancer_df$V7)

```
> cancer_df[missing_rows, ]
         V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11
24  1057013  8  4  5  1  2  1  7  3   1   4
41  1096800  6  6  6  9  6  1  7  8   1   2
140 1183246  1  1  1  1  1  1  2  1   1   2
146 1184840  1  1  3  1  2  1  2  1   1   2
159 1193683  1  1  2  1  3  1  1  1   1   2
165 1197510  5  1  1  1  2  1  3  1   1   2
236 1241232  3  1  4  1  2  1  3  1   1   2
250  169356  3  1  1  1  2  1  3  1   1   2
276  432809  3  1  3  1  2  1  2  1   1   2
293  563649  8  8  8  1  2  1  6 10   1   4
295  606140  1  1  1  1  2  1  2  1   1   2
298   61634  5  4  3  1  2  1  2  3   1   2
316  704168  4  6  5  6  7  1  4  9   1   2
322  733639  3  1  1  1  2  1  3  1   1   2
412 1238464  1  1  1  1  1  1  2  1   1   2
618 1057067  1  1  1  1  1  1  1  1   1   2
```

2. Regression imputation

```
> cancer_df[missing_rows, ]
         V1 V2 V3 V4 V5 V6        V7 V8 V9 V10 V11
24  1057013  8  4  5  1  2 5.3660666  7  3   1   4
41  1096800  6  6  6  9  6 8.2259101  7  8   1   2
140 1183246  1  1  1  1  1 0.8892805  2  1   1   2
146 1184840  1  1  3  1  2 1.6605574  2  1   1   2
159 1193683  1  1  2  1  3 1.0899300  1  1   1   2
165 1197510  5  1  1  1  2 2.2208736  3  1   1   2
236 1241232  3  1  4  1  2 2.7818889  3  1   1   2
250  169356  3  1  1  1  2 1.7605617  3  1   1   2
276  432809  3  1  3  1  2 2.1208694  2  1   1   2
293  563649  8  8  8  1  2 5.8459477  6 10   1   4
295  606140  1  1  1  1  2 0.9796727  2  1   1   2
298   61634  5  4  3  1  2 2.3918282  2  3   1   2
316  704168  4  6  5  6  7 5.5419942  4  9   1   2
322  733639  3  1  1  1  2 1.7605617  3  1   1   2
412 1238464  1  1  1  1  1 0.8892805  2  1   1   2
618 1057067  1  1  1  1  1 0.5687034  1  1   1   2
```

3. Regression with perturbation imputation

```
> cancer_df[missing_rows, ]
         V1 V2 V3 V4 V5 V6         V7 V8 V9 V10 V11
24  1057013  8  4  5  1  2  9.4723599  7  3   1   4
41  1096800  6  6  6  9  6 12.9062208  7  8   1   2
140 1183246  1  1  1  1  1  0.7659279  2  1   1   2
146 1184840  1  1  3  1  2 -1.8340744  2  1   1   2
159 1193683  1  1  2  1  3  0.2579615  1  1   1   2
165 1197510  5  1  1  1  2  7.9175138  3  1   1   2
236 1241232  3  1  4  1  2  1.4380937  3  1   1   2
250  169356  3  1  1  1  2  1.0969233  3  1   1   2
276  432809  3  1  3  1  2  0.6342375  2  1   1   2
293  563649  8  8  8  1  2  3.9966592  6 10   1   4
295  606140  1  1  1  1  2 -1.8721776  2  1   1   2
298   61634  5  4  3  1  2  2.9922076  2  3   1   2
316  704168  4  6  5  6  7  4.0593774  4  9   1   2
322  733639  3  1  1  1  2  5.8739521  3  1   1   2
412 1238464  1  1  1  1  1 -2.7916506  2  1   1   2
618 1057067  1  1  1  1  1 -1.2108064  1  1   1   2
```

4. I am using SVM to make classification models. The results are shown below.

Imputation technique	Accuracy (%)
Mean	97.86
Mode	99.29
Regression	97.86
Regression with perturbation	97.86
Remove missing value	97.79
Add new column to indicate missing value	96.32

From table above, the result doesn't change significantly when we change the imputation method (or removal). This means the imputation method is correct or maybe that particular attribute is not really that important.

Question 15.1

Describe a situation or problem from your job, everyday life, current events, etc., for which optimization would be appropriate. What data would you need?

I work as data analyst in an analytics division of an IT company. We help clients to optimize their power plant.

Problem: increase power generation efficiency in a coal fired power plant

Method to approach the problem:

1. Gather operational data for 3 years. Data needed: power generated, coal supplied, temperature and pressure of components inside the plant.
2. Build model with "efficiency" as target because we want to maximize this value.
3. Build other models that needs to be considered during optimization. For example we have to obey regulation that says we must keep emission below some level, so we have to make emission model.
4. Find best operational parameter (like what type of coal to use, when to send the coal, etc.) to maximize efficiency such that constraints are satisfied.