Data Analysis and Predication Using Hospital Medical Records Dataset

Fengmei Liu 497666

Yuan Wei 500135

Table of Contents

Problem and Background

Long waiting time in hospital is a common problem for many hospitals in Australia, it is a big concern for many hospitals since it could have a very negative impact on their customer services, their reputation as well as its profits. Thus, classifying patients' waiting time in which time range in hospital would be practical and useful, our problem proposal is to classify patient waiting time according to the duration of their waiting time, 5 categories are proposed, including "very short", "short", "medium", "long", "very long". In order to do so, the team decided to use the hospital dataset, which contains more than 30000 patients' medical records from January of 2009 to December 2009 throughou the whole year. The dataset is from the emergency department of the hospital and its patients are children who are aged from 0 to 15, all the data samples in the dataset are structured with 17 different attributes in total, including MRN (medical record number), presentation visit number, triage priority which is given by professional nurses, age, arrival date when the patient arrives the hospital, Dr seen date when the patient sees the doctor, Depart actual date when the patient leaves the hospital, depart status code and depart status description, department dest code, department dest description, TimeDiff Arrival-Actual Depart in minutes, TimeDiff TreatDrNr-Act. Depart in minutes, presenting complaint code, presenting complaint description, diagnosis code and diagnosis description.

Attributes	Data Types
MRN	Numerical: Integer
Presentation Visit Number	Numerical: Integer
Triage Priority	Nominal
Age (yrs)	Numerical: Integer
Arrival Date	Numerical: DateTime
Dr Seen Date	Numerical: DateTime
Depart Actual Date	Numerical: DateTime
Depart Status Code	Nominal
Departure Status Desc.	Nominal
Depart. Dest. Code	Nominal
Depart. Dest. Desc.	Nominal
TimeDiff Arrival-Actual Depart (mins)	Numerical: Integer
TimeDiff TreatDrNr-Act. Depart (mins)	Numerical: Integer
Presenting Complaint Code	Nominal
Presenting Complaint Desc.	Nominal
Diag Code	Nominal
Diagnosis Desc.	Nominal

Table1. Dataset Attributes and Data Types

Classifying patients' waiting time has lots of benefits. Firstly, it could help patients to better manage their time. For instance, if a patient knows that he or she must wait for several hours, the patients could make use of his or her waiting time to do other more important things. Meanwhile, it is also beneficial for the hospital. For example, it could improve their customer services and help them to manage their resources more efficiently.

Data Cleaning

Data cleaning is an important step during data preparation, it is about how to deal with missing or wrong variables in the dataset, including incomplete data, inconsistent data as well as noisy data. The reason why we need to do data cleaning is because that Dg could help improve the overall quality of the dataset and enhance the rate of successful data mining. As for the data cleaning, it normally deals with three types of dirty data, including missing data, inconsistent data as well as noisy data.

Incomplete Data

In general, there are four main approaches to deal with missing data, including cold-deck imputation which replaces missing values with mean or median of that attribute, hot-deck imputation which tries to find the most similar cases to the case which has a missing value and replaces the missing value from the most similar case, statistical imputation which uses functions such as linear regression and decision tree to fill in the missing value with most likely value, and predictive imputation which makes use of other attributes to predict the missing value.

As for the hospital data set, there are 223 missing values for four attributes in total, 1 variable missing for TimeDiff TreatDrNr-Act. Depart (mins), 19 variables missing for Presenting Complaint Code and Presenting Complaint Desc and 184 missing value for Dr Seen Data. For TimeDiff TreatDrNr-Act. Depart (mins) which type is numerical, the team used cold-deck imputation to replace the missing value, which means that getting the average value for the time-off attribute, then replace the missing value with the average value. The reason we decided to do so is because that the average of TimeDiff TreatDrNr-Act. Depart (mins) involves each value in the dataset and can represent the central tendency of the attribute, so replacing it would be more reasonable and accurate than other values like median, maximum and minimum.

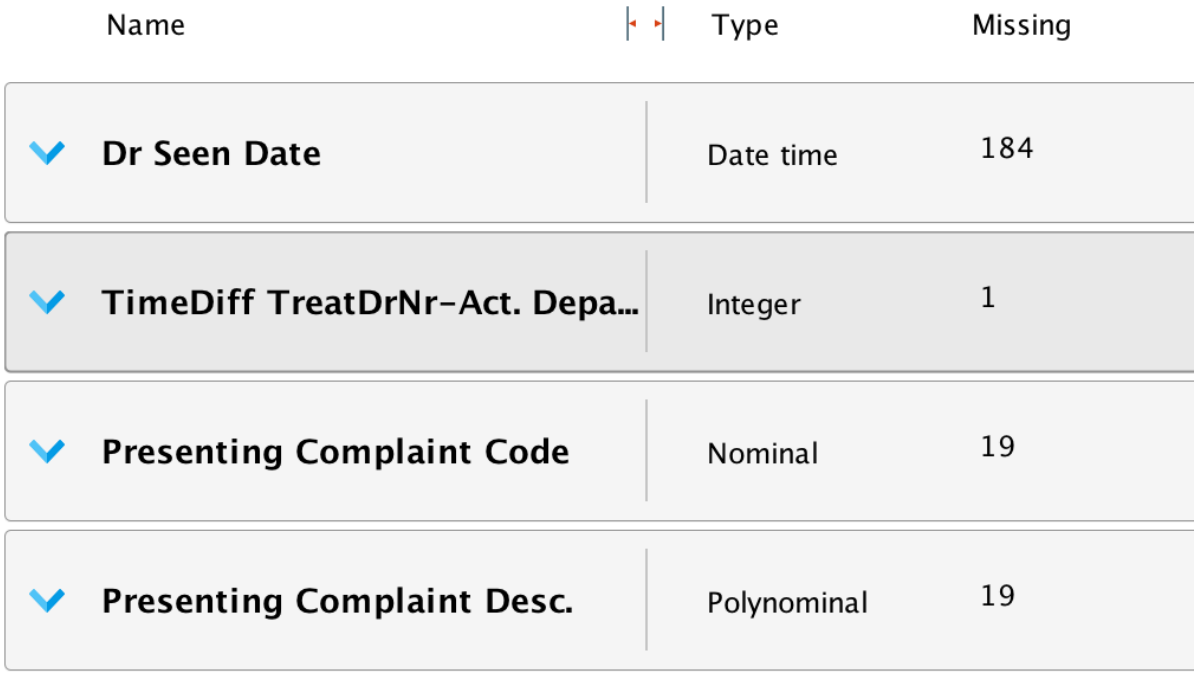


Figure 1. Missing Data

Regarding complaint code and Presenting Complaint Desc, there are 38 values missing in the whole dataset in total. As for them, the team used hot-deck imputation to substitute the missing values. We analyzed the dataset and tried to find the most similar sample which are like the samples with missing values based on triage priority, age and Diag Code, then replace the missing values using the values from the most similar samples. For instance, if the age is 4, priority is 3 and Diag Code is Z53.2, the complaint code and complaint Desc in the complete data samples are 9000 and injury. We replaced the missing values using 9000 and injury too.

For the Dr Seen Date, there are 184 missing values. There are two ways to deal with them. Firstly, Dr Seen Date could be calculated by Depart Actual Date and TimeDiff TreatDrNr-Act. Depart (mins) in datetime format. Secondly, one simple way is to remove these values from the dataset. The team decided to use the first calculation method to get its date instead of just deleting these data from the whole dataset, the reason is because that removing date might also lead to the loss of the important information.

Inconsistent Data

For the inconsistent data, what we found in the hospital dataset is that around 5 percent of the values for the TimeDiff TreatDrNr-Act. Depart (mins) attribute are inconsistent with our calculation result. For the TimeDiff TreatDrNr-Act. Depart (mins) is the time difference between Depart Actual Date and Dr Seen Date in minutes. After doing the calculation in RapidMiner, we found that some values in the dataset do not match with the values we calculated. As for these inconsistent values, we decided to replace them with the values we calculated. The reason we chose to do this is because that it is hard to decide the reason for the problem, it could be the Depart Actual Date which was wrongly recorded or Dr Seen Date which was wrongly recorded. Thus, correcting the inconsistent values in TimeDiff TreatDrNr-Act using the calculation would be more efficient and reasonable.

Dr Seen Date	Depart Actual Date	TimeDiff TreatDrNr-Act...	New_TimeDiff TreatDrNr-Act. ...	diff ↑
May 16, 2010 5:38:00 PM AEST	May 17, 2010 12:37:00 AM AEST	1440	419	-1021
Dec 9, 2010 1:15:00 AM AEDT	Dec 9, 2010 1:15:00 AM AEDT	891	0	-891
Mar 24, 2009 5:33:00 AM AEDT	Mar 24, 2009 5:33:00 AM AEDT	658	0	-658
Jul 28, 2009 1:16:00 AM AEST	Jul 28, 2009 3:00:00 AM AEST	435	104	-331
Jun 12, 2009 6:51:00 AM AEST	Jun 12, 2009 6:51:00 AM AEST	329	0	-329
Oct 1, 2010 4:01:00 AM AEST	Oct 1, 2010 4:01:00 AM AEST	318	0	-318
Jan 23, 2010 10:11:00 PM AEDT	Jan 23, 2010 11:00:00 PM AEDT	360	49	-311
Dec 3, 2009 4:28:00 AM AEDT	Dec 3, 2009 4:29:00 AM AEDT	304	1	-303
Nov 16, 2009 9:19:00 PM AEDT	Nov 16, 2009 9:19:00 PM AEDT	289	0	-289
Aug 16, 2009 8:32:00 PM AEST	Aug 16, 2009 11:13:00 PM AEST	449	161	-288
Aug 24, 2009 5:39:00 AM AEST	Aug 24, 2009 10:20:00 PM AEST	1289	1001	-288
Nov 16, 2009 8:18:00 PM AEDT	Nov 16, 2009 9:00:00 PM AEDT	330	42	-288
Dec 30, 2009 1:30:00 PM AEDT	Dec 30, 2009 3:15:00 PM AEDT	390	105	-285
Jun 4, 2010 10:24:00 PM AEST	Jun 4, 2010 10:53:00 PM AEST	313	29	-284
Mar 9, 2009 12:44:00 AM AEDT	Mar 9, 2009 1:36:00 AM AEDT	334	52	-282
Nov 27, 2009 1:13:00 AM AEDT	Nov 27, 2009 1:13:00 AM AEDT	278	0	-278

Figure 2. Inconsistent Data

Noisy Data

As for the noisy data, its goal is to detect errors, outliners and noise in the dataset, which is also helpful for improving the quality of the dataset. Approaches which could be used to deal with noisy data include binning, regression, clustering, combined with computer and human inspection.

As for binning, it has two main types: equal distance binning which means that each bin has the same range and equal depth partitioning method which means that each bin contains approximately the same samples. For regression, it uses a function to smooth the dataset. Clustering is to put data into groups according to their similarity, then remove outliners. As for the hospital dataset, our problem is to predict patient's waiting time in which class, and the patients' waiting time in the hospital is the time difference between TimeDiff Arrival-Actual Depart (mins) and TimeDiff TreatDrNr-Act. Depart (mins). During our preliminary statistics, we found that the waiting time are smaller than zero after calculation, this is not corrected which should be removed from the dataset. As for outliers, we found that the distribution of the waiting time is very sparse which ranges from 0 to 1000. For our problem, we only care about the waiting time less than four hours, waiting time which is longer than four hours are very long. Thus, as to improve the quality and performance of our model, we removed data samples from our model which waiting time is larger than 250 or smaller than zero. The main reason we chose to remove these data samples from our dataset is that these data samples have little value for our problem and would have a negative effect on our model's performance.

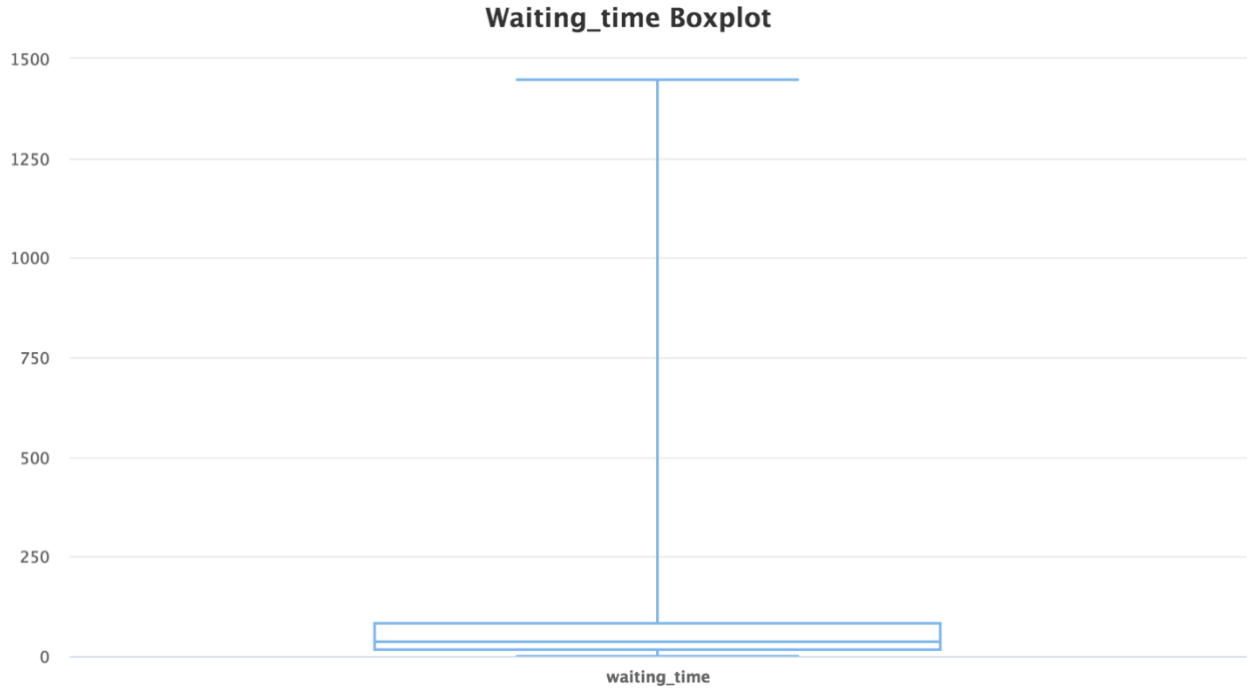


Figure 3. Waiting time Boxplot

Data Integration, Transformation and Reduction

As for the data integration, the dataset only comes from one source, so it is unnecessary to do schema integration. Meanwhile, the entities that the dataset represents are quite clear and readable, it has no conflicts with the reality. In addition, there are no value conflicts and redundant data in the dataset since it has only one data source. Thus, it is not needed to do the data integration for the hospital dataset.

For the data transformation, we used data aggregation to summarize the hospital dataset. Firstly, we aggregated Triage Priority together as to see which priority has more frequencies in the dataset. Meanwhile, we also aggregated the complaint code as to see which compliant code is the most common reason why children go to the hospital. And it will be further discussed in the Explorative data analysis section.

Additionally, as for the data generalization, the dataset is structured, and each attribute is human readable and understandable. Thus, it is unnecessary to generalize the dataset. As for the data normalization which is only for numerical values, the dataset has three numerical features, including TimeDiff TreatDrNr-Act. Depart (mins) and TimeDiff Arrival-Actual Depart (mins) and waiting time. In order to predict patient time in which time range, we need to normalize the waiting time. After visualizing the patient waiting time, which is range from zero to more than 900, the range of the data is huge. In order to predict the patient waiting time in which category, we did data normalization for our model, after normalization, the numerical values range is from 0 to 1.

Row No.	Triage Prio…	Age (yrs)	TimeDiff A…	New_Time…	waiting_time
1	0.500	1	0.322	0.283	0.454
2	0.750	0.533	0.004	0.012	0.048
3	0.500	0.800	0.042	0.024	0.285
4	0.750	1	0.120	0.064	0.618
5	0.750	0.133	0.047	0.016	0.398
6	0.500	0.067	0.084	0.068	0.257
7	0.750	1	0.378	0.346	0.382
8	0.750	1	0.080	0.019	0.671
9	0.500	0	0.040	0.018	0.317
10	0.500	0.600	0.133	0.128	0.157
11	0.750	0.267	0.056	0.015	0.490
12	0.750	0.600	0.072	0.034	0.466
13	0.500	0.867	0.108	0.094	0.233
14	0.750	0.800	0.062	0.027	0.438
15	0.500	0	0.029	0.026	0.141
16	0.500	0.333	0.023	0.021	0.145
17	0.500	0.400	0.055	0.061	0.064
18	0.500	0	0.034	0.032	0.137
19	0.750	0.267	0.035	0.031	0.153

ExampleSet (25,853 examples, 0 special attributes, 5 regular attributes)

Figure 4. Normalization

As for the data reduction, we did dimensionality reduction for the dataset. We found that some attributes are redundant, including departure status desc, depart dest desc, presenting complaint desc and diagnosis desc, we removed these redundant features from the dataset. Meanwhile, there are also two unimportant attributes in the dataset, including MRN and presentation visit number since they are not related to the prediction of patient's waiting time.

Original Features
MRN
Presentation Visit Number
Triage Priority
Age (yrs)
Arrival Date
Dr Seen Date
Depart Actual Date
Depart Status Code
Departure Status Desc.
Depart. Dest. Code
Depart. Dest. Desc.
TimeDiff Arrival-Actual Depart (mins)
TimeDiff TreatDrNr-Act. Depart (mins)
Presenting Complaint Code
Presenting Complaint Desc.
Diag Code
Diagnosis Desc.

Figure 5. Data Reduction

Feature Selection and Design

Feature selection is an important step of successful data mining. For the hospital dataset, there are 17 features in total. Among them, departure status desc, depart dest desc, presenting complaint desc and diagnosis desc are duplicated with depart status code, depart dest code, presenting complaining code and diag code, so we will remove these duplicated attributes from the dataset since they would affect the performance of our model negatively and not bring any important information for predicting patient's waiting time in which time range. Besides these four attributes, attributes MRN and presentation visit number are useless in predicting patient's waiting time, these are just randomly numbers, so we decided to remove them too. For arrival date, Dr Seen Date and Depart actual date, these three attributes are closely related to TimeDiff Arrival-Actual Depart(mins) and TimeDiff TreatDrNr-Act. Depart(mins). For these three attributes, we decided to removed them too since TimeDiff Arrival-Actual Depart(mins) and TimeDiff TreatDrNr-Act.Depart(mins) could subsitute that three attributes and patient's waiting time could be calculated just from that two attributes instead of using arrival date, Dr Seen Date and Depart Actual Date. In addition, as to predict patient's waiting time, we need to create one more attribute for the dataset which is the time difference between TimeDiff Arrival-Actual Depart(mins) and TimeDiff TreatDrNr-Act.Depart(mins). The reason for the creation is because that this attribute would be used as label for our model. Another attribute is also added into the dataset, which is called Arrival Time Category, its type is normal and has four values, including: morning, afternoon, evening and early morning. We divided 24 hours into four sections. Early Morning means time from 00.00am to 06:00 am, morning means time from 06:00am to 12:00pm; afternoon means time from 12:00pm to 06:00pm, then evening means: 06:00pm to 12:00am. The reason we added this new attribute and divided the timespan into four sections is to see which time section has more patients and the hospital is busier than others, this could affect the patients' waiting time. Another attribute we added is called Queue Time, which would be used as the label for our model.

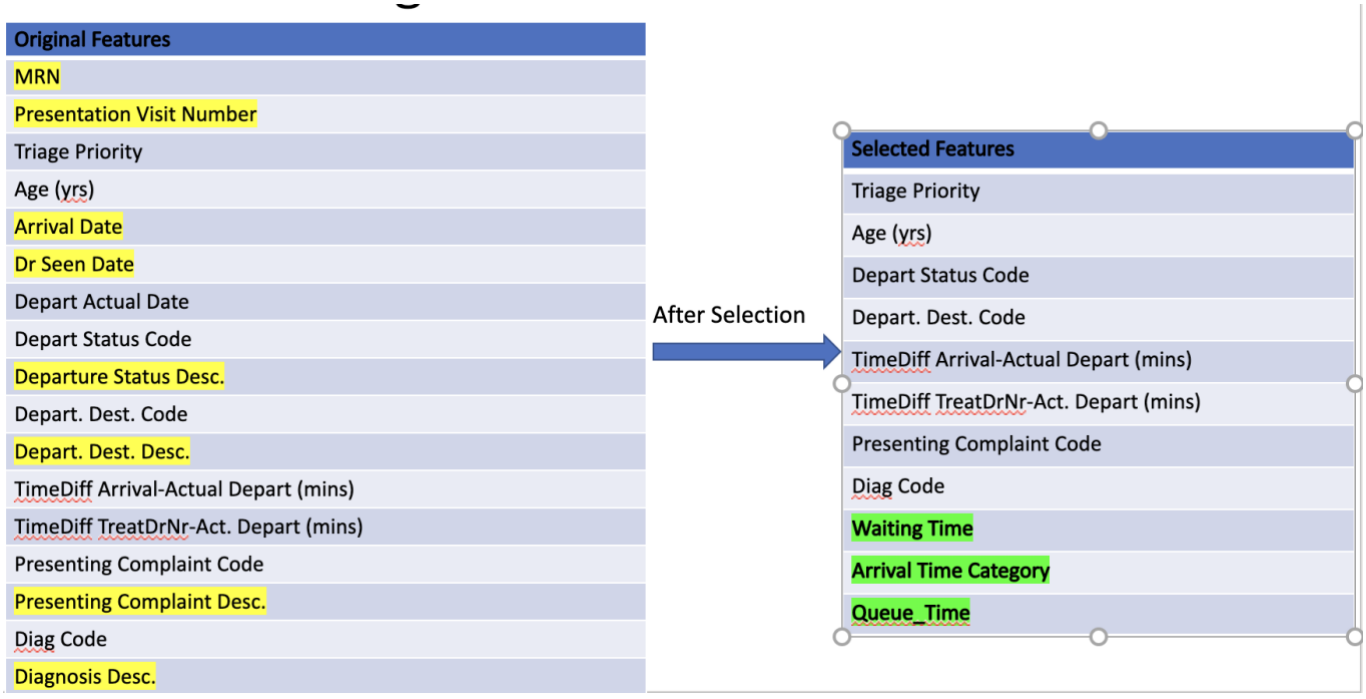


Figure6. Attribute selection

Exploratory Data Analysis

Exploratory data analysis is very important in data analysis since it could help analysists to know their data samples, including data distribution, detecting outliners, identifying data quality problems as well as finding the relationships between different data features. For the hospital dataset, we used different graphs to visualize the dataset. The following are the results with our findings.

Firstly, about priority and the average of waiting time, for priority four, its average waiting time is the largest which is around 90mins. The second largest value for average of waiting time is priority 5 which is around 75 mins. For the priority 1, it has the lowest value for the average of waiting time. It is not very hard to understand. The higher priority of priority, the shorter of waiting time, which mean that it is more urgent in reality. It is quite interesting for priority 3 and priority 4. For priority 3, it has the largest number of patients, but its average waiting time is shorter than priority four. The implication for this finding is the hospital could use these data as references to identify the reason for this.

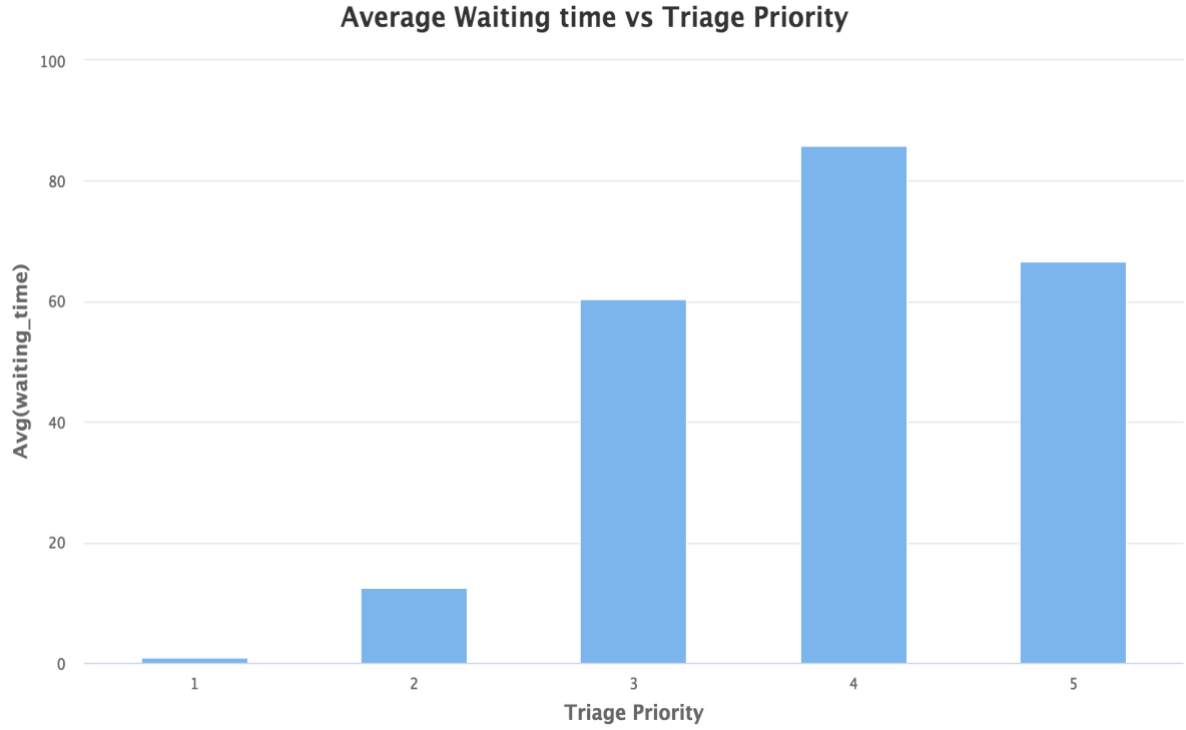


Figure7. Average Waiting Time vs Triage Priority

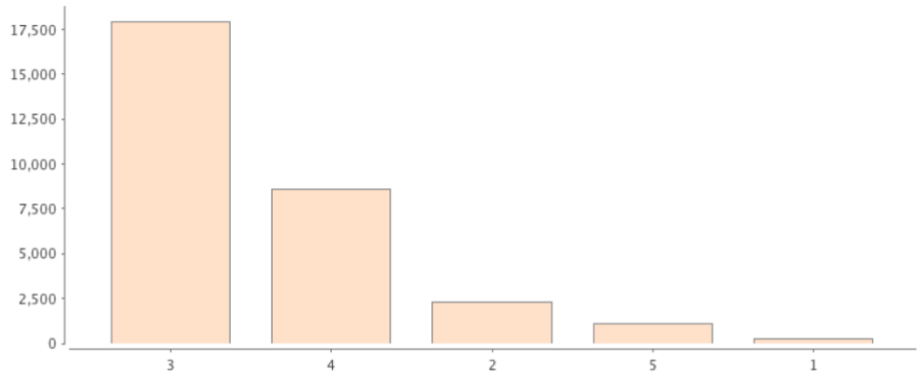
Triage Priority

Summary

Category

Missing: 0.00%
Infinite: 0.00%
ID-ness: 0.02%
Stability: 59.57%
Valid: 40.41%

Top Values

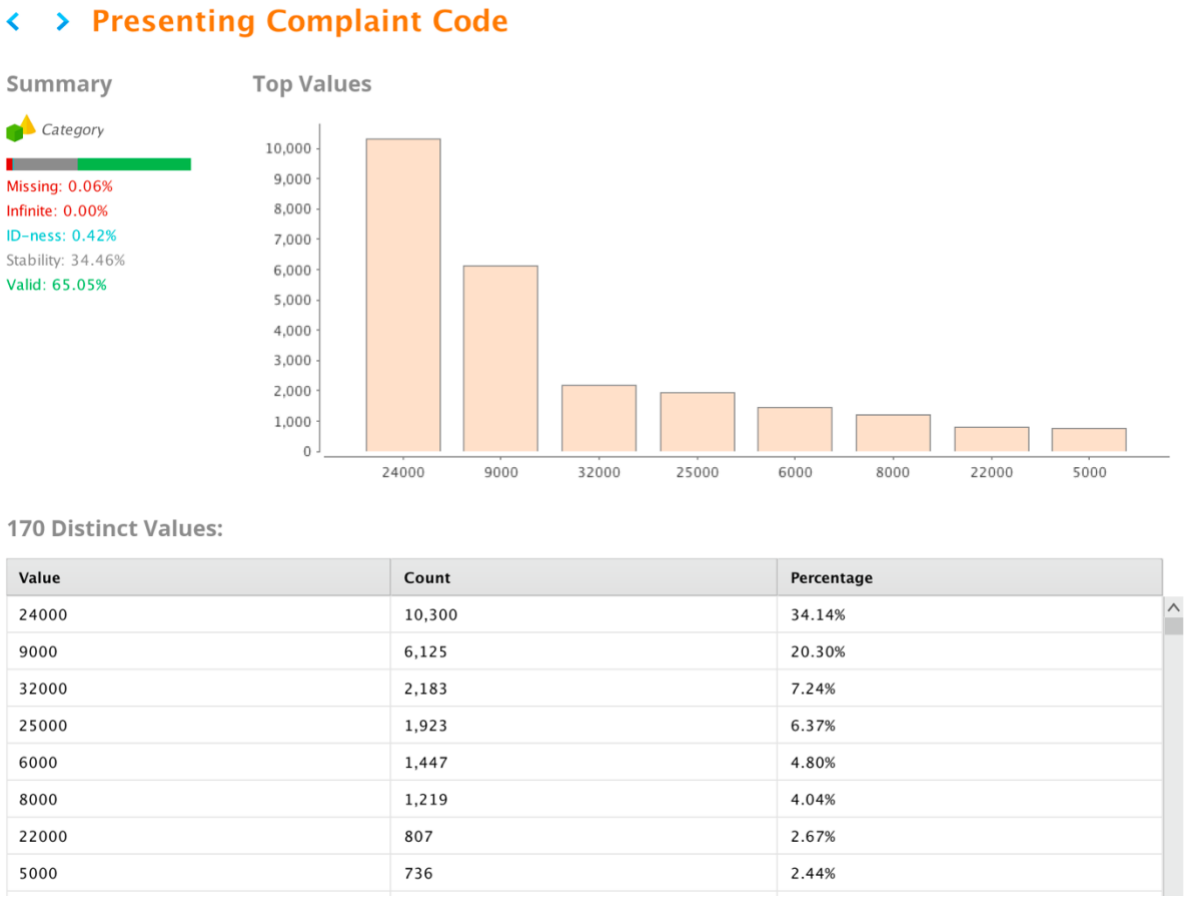


5 Distinct Values:

Value	Count	Percentage
3	17,915	59.34%
4	8,604	28.50%
2	2,306	7.64%
5	1,112	3.68%
1	251	0.83%

Figure8. Triage priority

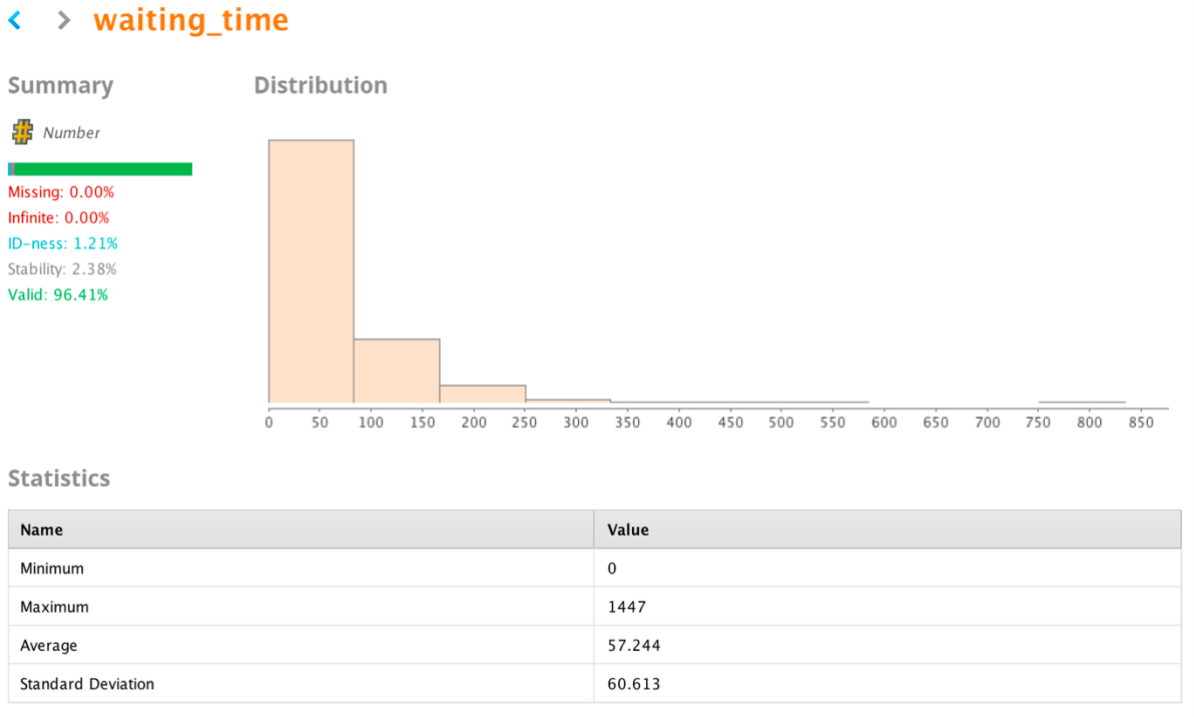
Secondly, we also found from the dataset that the most common reasons why children go to the hospital is because of injury and pediatric problems which is shown as the following, among which 24000 and 9000 are the code for pediatric and injury respectively.



Value	Count	Percentage
24000	10,300	34.14%
9000	6,125	20.30%
32000	2,183	7.24%
25000	1,923	6.37%
6000	1,447	4.80%
8000	1,219	4.04%
22000	807	2.67%
5000	736	2.44%

Figure8. Presenting Complaint Code

Thirdly, regarding the waiting time, we found that the average of waiting time is 57 minutes, the maximum of waiting time is 1447 minutes, and the minimum of waiting time is zero, and the standard derivation is 60. From here, we could conclude that the waiting time in the hospital is around one hour.



Name	Value
Minimum	0
Maximum	1447
Average	57.244
Standard Deviation	60.613

Figure9. Waiting time distribution

Fourthly, we also discovered that the duration of the patients stays in the hospital which is the attribute of TimeDiff Arrival-Actual Depart (mins), the majority of the patients have to stay in the hospital from 1 hour to around 1.5 hours. From the graph, we can see that around 20000 patients wait in the hospital from 60 mins to 150 minutes.

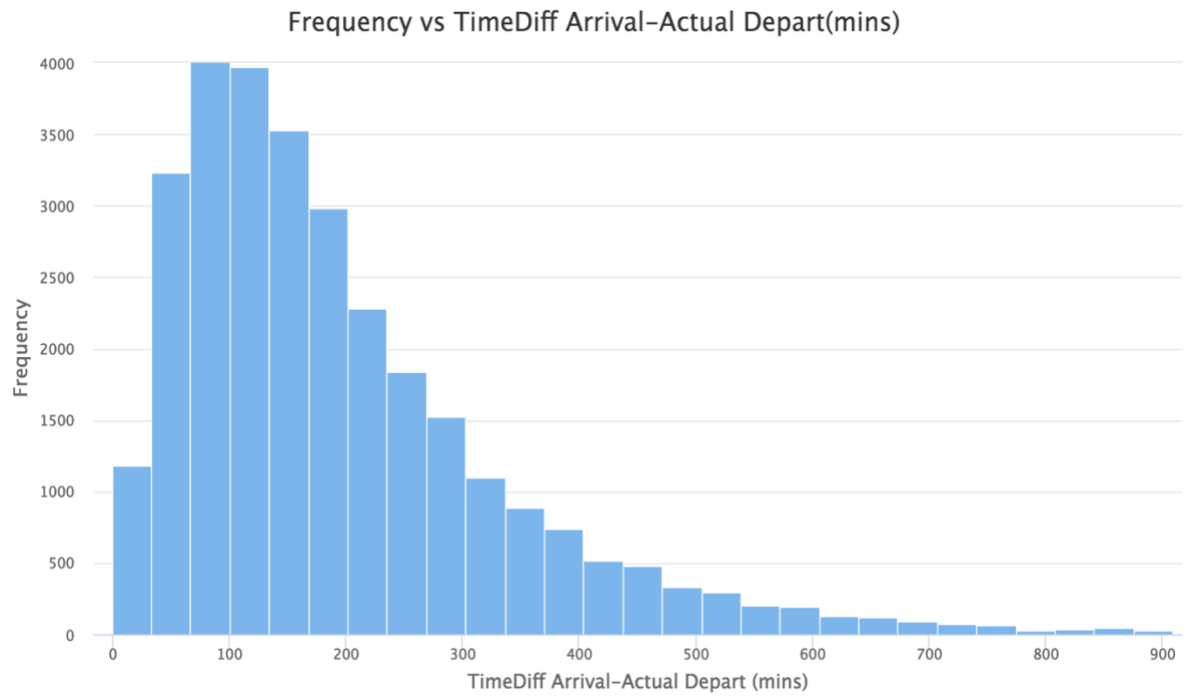


Figure10. TimeDiff Arrival-Actual Depart Histogram

Fifthly, we also found that the infants who are less than 1 year old have largest number of records for the patients which is shown as the following.

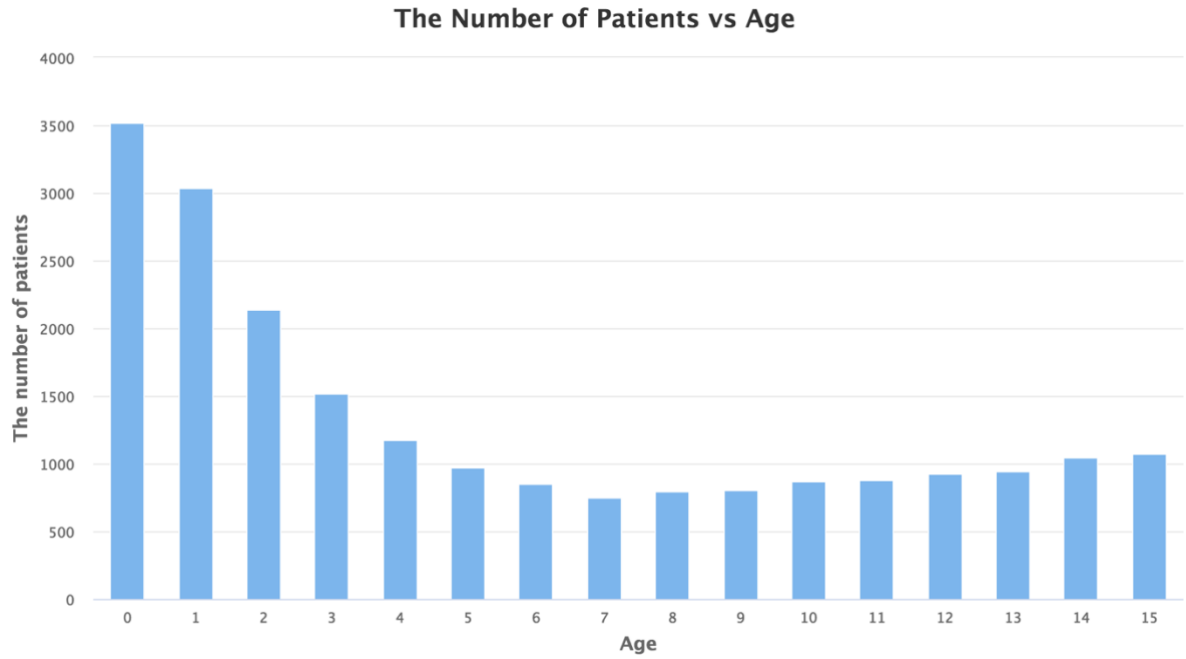


Figure11. The Number of Patients vs Age

Lastly, the distribution of data among the years in each month is like the following. From here we could see that during May, there are more patients visiting the hospital than others which is wintertime here. During summertime November and December, there are less children who visit the hospital.

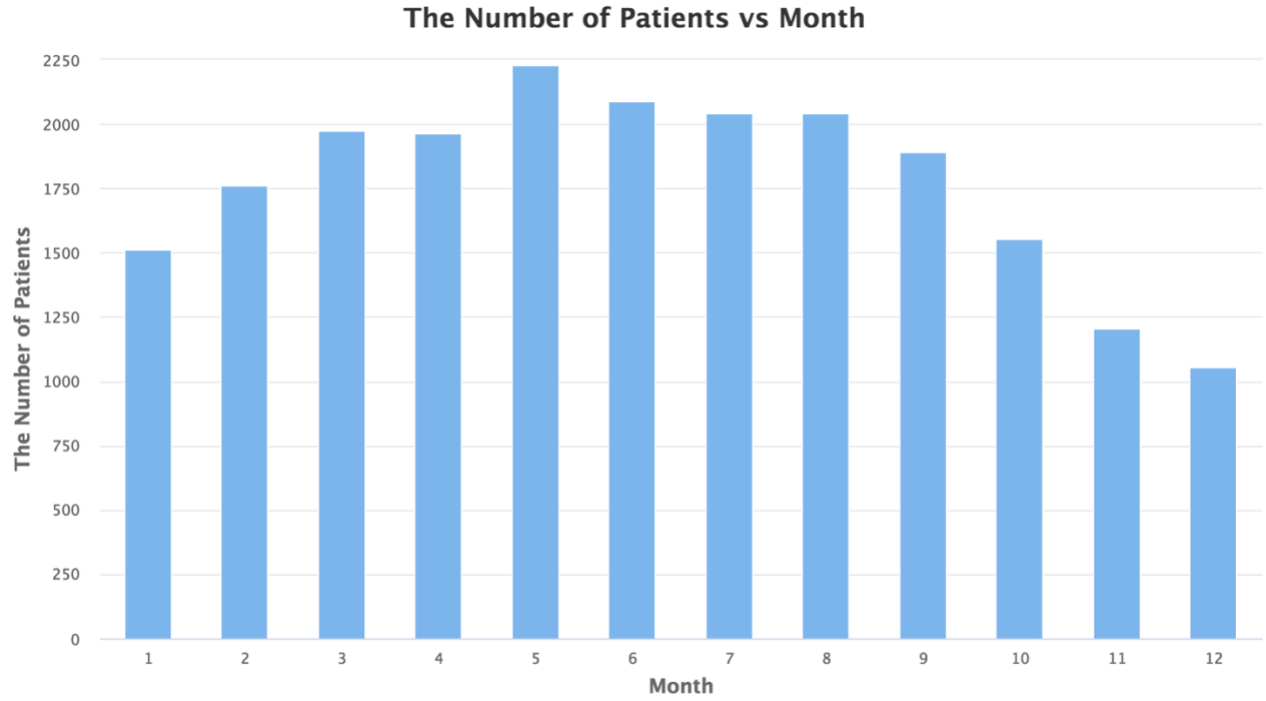
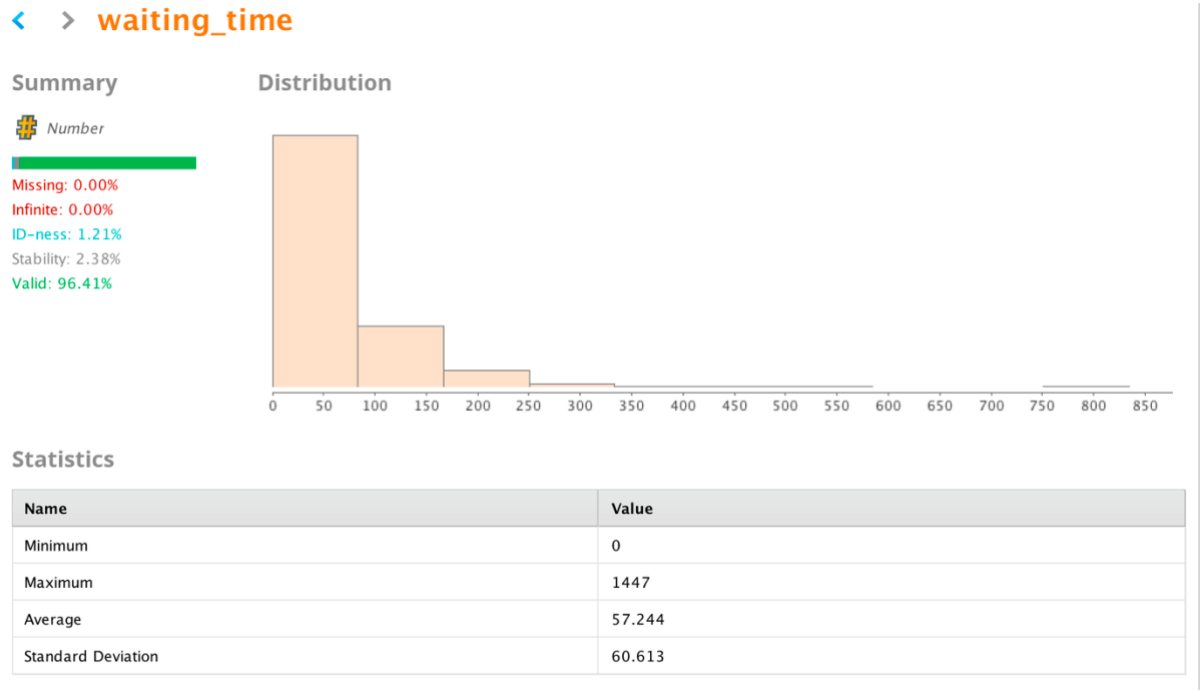


Figure12. The Number of Patients vs Month

Descriptive Data Mining

Descriptive data mining is generally used to produce correlation, cross tabulation, frequency etcetera, its goal is to find the regularities in the data and to detect patterns and structures in the data. It focuses on the summarization and conversion of the data into meaningful information for reporting and monitoring, the common method used is clustering which is to group data into different categories based on the similarity. In our analysis, we used two different clustering algorithms to divide patient waiting time into different groups. The first method we used is K-Means which is a distance-based clustering method, the other method is DBScan which is a density-based clustering method.

For K-mean, before we put the patient waiting time into different groups. During exploratory data analysis, we found that the range of the patient waiting time is very large which ranges from zero to more than 1447 minutes. In date cleaning, we removed the waiting time which is equal zero and waiting time more 250 minutes, as the majority of the waiting time for patients ranges from 30minutes and 250 minutes so we moved these outliners which waiting time is smaller than 30mins and is larger than 250 minutes.



Name	Value
Minimum	0
Maximum	1447
Average	57.244
Standard Deviation	60.613

Figure12. Waiting time distribution

Then we tried to cluster the waiting time into different groups. According to the performance graph, we believed that 60 is the best value we could get since its average within centroids distance is relatively smallest. The reason why we did not choose 70 or 90 is that the avg. Within centroid distance has no big difference in comparison with 60.

Optimize Parameters (Grid) (11 rows, 4 columns)

iteration	Clustering.k	Performance.main_criterion	Avg. within centroid dist...
11	100	Avg. within centroid distance	-0.008
10	90	Avg. within centroid distance	-0.008
9	80	Avg. within centroid distance	-0.009
8	70	Avg. within centroid distance	-0.010
7	60	Avg. within centroid distance	-0.011
6	51	Avg. within centroid distance	-0.012
5	41	Avg. within centroid distance	-0.015
4	31	Avg. within centroid distance	-0.018
3	21	Avg. within centroid distance	-0.023
2	11	Avg. within centroid distance	-0.035
1	1	Avg. within centroid distance	-0.196

Figure12. K-Mean Clustering Performance

Predictive Data Mining

Predictive data mining is to use some variables to predict unknown or future values of other variables, it normally has two types: linear regression and classification. For linear regression, it is to use a line to predict the value of variables based on other variables. For classification, it is to identify to which of a set of categories a new observation belongs on the basis of the training set of data containing observations whose category is labelled in advance. For the hospital dataset, predicting the patients' waiting time, it is a linear regression problem not a classification problem.

In order to obtain the best model of predicting patients' waiting time in which category, we used three different methods, including decision tree, K-NN and neural network. According to the waiting time, we divided the selected dataset into five classifications and created one new attributed called Queue Time which is used as a label in the classifying the patient's waiting time.

Category	Queue Time
Very short	Larger than zero but smaller than 60 minutes
Short	Larger than 60 but smaller than 120
Medium	Larger than 120 but smaller than 180
Long	Larger than 180 but smaller than 240
Very Long	Larger than 240

Table2. Queue Time

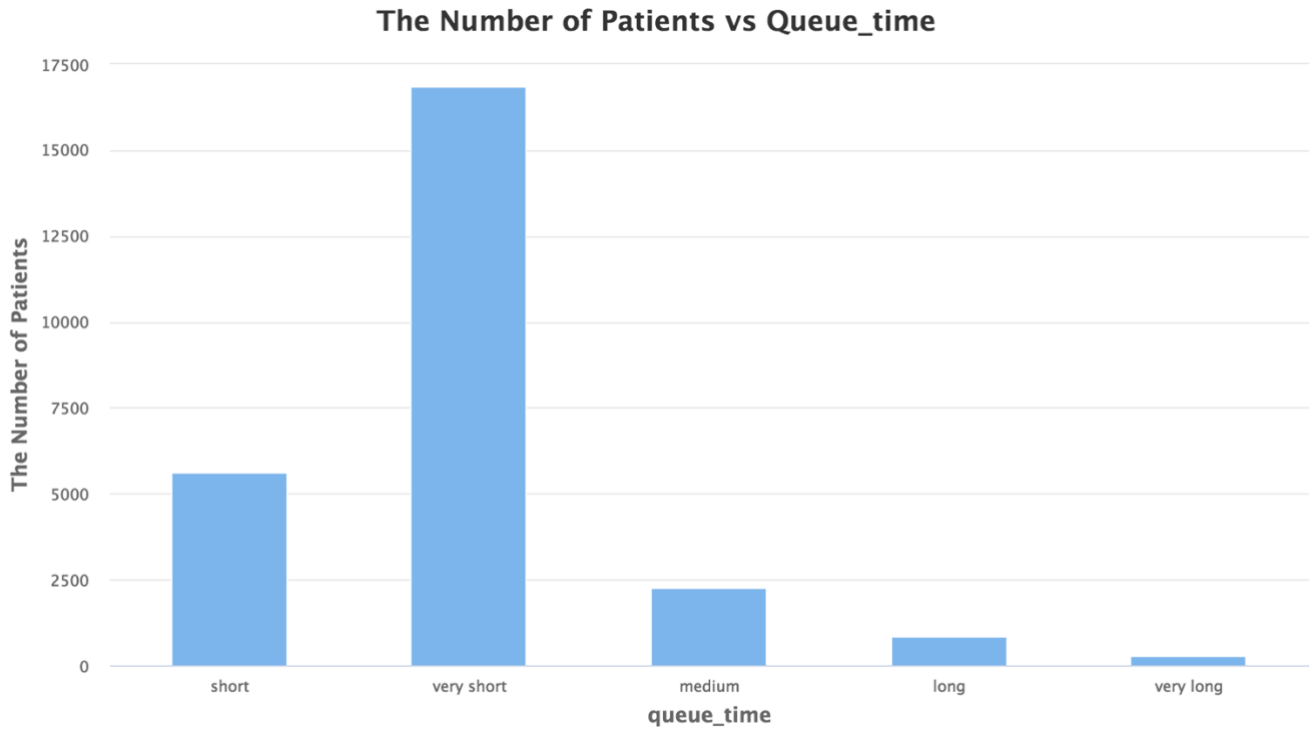


Figure13. The Number of Patients vs Queue Time

waiting_time	queue_time
114	short
13	very short
72	short
155	medium
100	short
65	short
96	short
168	medium
80	short

Figure14. Queue Time

For the decision tree, it uses a graph like a tree and their consequences to present an algorithm, it is a tool for decision support. In the first place, all the training samples are at the root, then training samples will be partitioned according to split attributions, this partition process happens recursively. The algorithm will stop partitioning when there are no further attributes for more partitioning, or no samples left. The reason we chose decision tree is because the classification results are displayed graphically and easily to be explained and interpreted. Thus, it would be convenient to see how long patients would wait in the hospital.

In addition, as for k-nearest neighbors (KNN) algorithm is a supervised machine learning algorithm which could be used for classification as well as regression. Here for the hospital dataset, we used it for classification instead of regression. For KNN, it believes that similar things exist in proximity. The algorithm works as the following: first initiating K which is the number of neighbors you choose, then for each sample in the dataset, calculating the distance between query example and current example from the data, sorting the distance from shortest to largest, then selecting the first K entries from the sorted collection. The reason we chose KNN to classify patient's waiting time is that it is very easy to implement and robust to noisy data since we only needed to care about one parameter K in RapidMiner.

Furthermore, as for the neural network, it is based how human's brain works which consists of numerous neurons. Normally, it has three layers, including input layer, hidden layers and output layer. For the hidden layers, it could be composed of different layers, each neuron in neural network receives several inputs, these inputs might be vectors. Meanwhile, an activation function is applied to the inputs which would produce the output value of the

neuron. For each hidden layer, it receives inputs from its previous layer, then use an activation function to calculate the output of each neuron, and these outputs will become inputs for next layer. For the activation functions, the commonly used functions are ReLu, Maxout and Sigmoid. Nue can be used for regression problems as well as classification. Here we used it to classify patient's waiting time. The main reason why we chose to use it is because that the model does not have to be re-programmed after learning, so it would save lots of time in comparison with other methods.

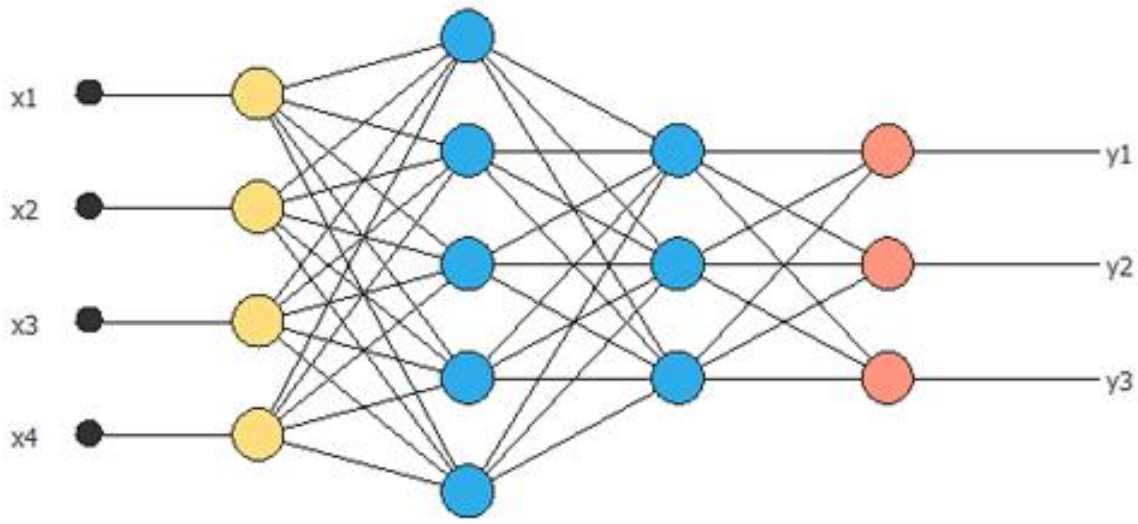


Figure15. Neural networks

As for the hospital dataset, we used the whole dataset instead of sampling a portion of it. For the modelling part, we used cross validation to deal with training and testing data which is also called K-fold sampling, the reason is because that it divides the whole datasets into K folds, then for each fold, it uses split data to divide them into training and testing part. Thus, each data sample in the dataset could be used for training as well as testing which is believed to be better than hold-on sampling methods. The processing of the model looks like this:

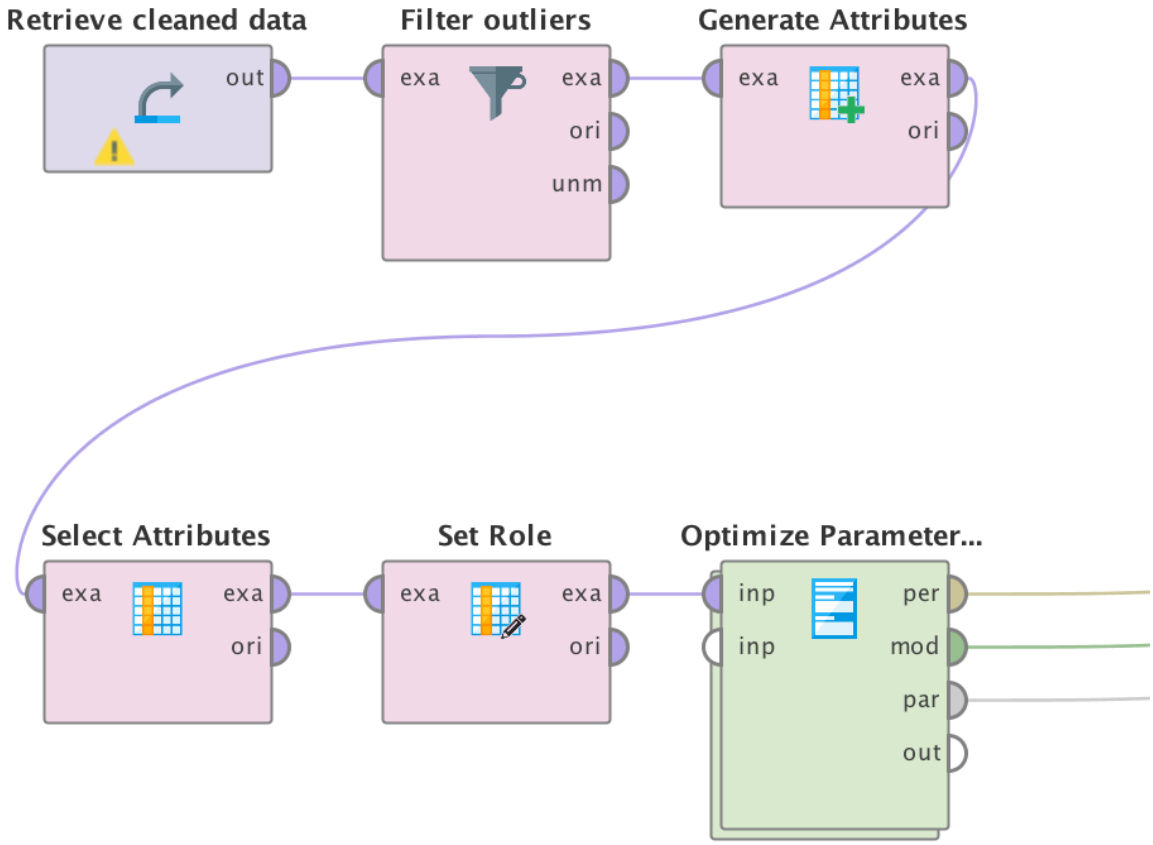


Figure16. The processing of the model

Evaluation and Result Analysis

For all the models, including decision tree, KNN, neural network, the team used 10-fold cross validation for sampling instead of hold-on method. The reason we chose to use cross validation is because it could reduce bias and improve accuracy. The following are the result for the three algorithms we used.

As for the decision tree, we did two experiments. For the first experiment, we divided the waiting time into different sections like from 0 to 15 minutes, from 15 to 30 minutes, from 30 to 45, from 45 to 60 and more than 60. We used loop operation in our data modelling. The best accuracy we got is 73.88 percent and the relative error is 33.42 percent which is very high. The reason why we used accuracy as our main performance criterion is that predicting waiting time is a linear regression problem, so accuracy could be the main indictor for our model.

Category	Queue Time
Very short	Larger than zero but smaller than 15 minutes
Short	Larger than 15 but smaller than 30
Medium	Larger than 30 but smaller than 45
Long	Larger than 45 but smaller than 60
Very Long	Larger than 60

Table3. Queue Time classification One

ParameterSet

```
Parameter set:

Performance:
PerformanceVector [
-----accuracy: 73.88% +/- 0.75% (micro average: 73.88%)
ConfusionMatrix:
True:   very long       very short      medium  long    short
very long:      8818    759     569     554     866
very short:     233     4422    147     41      722
medium: 131     1       1485    145     201
long:   262     0       203     1357    4
short:  463     648     611     193     3018
-----absolute_error: 0.334 +/- 0.007 (micro average: 0.334 +/- 0.347)
-----relative_error: 33.42% +/- 0.75% (micro average: 33.42% +/- 34.72%)
]
Decision Tree.apply_prepruning  = false
Decision Tree.criterion = gini_index
Decision Tree.apply_pruning     = false
Decision Tree.maximal_depth     = 10
Decision Tree.confidence        = 0.5
```

Figure17. D-Tree Parameter Set One

The performance for this is quite low. After visualizing the distribution of waiting time, we found that the density of data in that five categories (“very short”, “short”, “medium”, “long” and “very long”) are not the same and data are not equally distributed. Thus, we tried the second method. we changed the criterion to the following, then we run the decision tree again, the performance improved around 30 percent shown in the following, which is 92.68 percent and relative error is also reduced from 33 percent to 11 percent. The reason why the performance is improved is mainly because that after enlarging the range of the category, the sparsity of the data was reduced.

Category	Queue_Time
Very short	Larger than zero but smaller than 60 minutes
Short	Larger than 60 but smaller than 120
Medium	Larger than 120 but smaller than 180
Long	Larger than 180 but smaller than 240
Very Long	Larger than 240

Table4. Queue Time classification Two

ParameterSet

```
Parameter set:

Performance:
PerformanceVector [
-----accuracy: 92.68% +/- 0.44% (micro average: 92.68%)
ConfusionMatrix:
True:   short   very short      medium  long    very long
short:  4796    195     183     2       70
very short:     733     16637   192     77      91
medium: 89      0       1826    97      38
long:   0       0       48      640     31
very long:      17      17      0       13      61
-----absolute_error: 0.109 +/- 0.003 (micro average: 0.109 +/- 0.236)
-----relative_error: 10.93% +/- 0.35% (micro average: 10.93% +/- 23.65%)
]
Decision Tree.apply_prepruning  = false
Decision Tree.criterion = information_gain
Decision Tree.apply_pruning     = true
Decision Tree.maximal_depth     = 10
Decision Tree.confidence        = 0.5
```

Figure18. D-Tree Parameter Set Two

As for the KNN, its performance is better than decision tree which accuracy is 95.94 percent and the K is 11 which improves a lot in comparison with decision tree. Meanwhile its relative error is much smaller than decision tree. The reason why KNN is better is that it is more robust to noisy data in comparison with decision tree.

ParameterSet

```
Parameter set:

Performance:
PerformanceVector [
-----accuracy: 95.94% +/- 0.53% (micro average: 95.94%)
ConfusionMatrix:
True:   short   very short      medium  long    very long
short:  1066    18      34      0       14
very short:     56      3329    2       3       18
medium: 12      0       428     22      7
long:   0       0       4       138     19
very long:      0       0       0       1       0
-----weighted_mean_recall: 73.80% +/- 2.09% (micro average: 73.81%), weights: 1, 1, 1, 1, 1
ConfusionMatrix:
True:   short   very short      medium  long    very long
short:  1066    18      34      0       14
very short:     56      3329    2       3       18
medium: 12      0       428     22      7
long:   0       0       4       138     19
very long:      0       0       0       1       0
-----weighted_mean_precision: 73.82% +/- 0.98% (micro average: 73.76%), weights: 1, 1, 1, 1, 1
ConfusionMatrix:
True:   short   very short      medium  long    very long
short:  1066    18      34      0       14
very short:     56      3329    2       3       18
medium: 12      0       428     22      7
long:   0       0       4       138     19
very long:      0       0       0       1       0
]
k-NN.k  = 11
```

Figure19. KNN Parameter Set

As for the neural network, the team used traditional three layers of model, including input layer, one hidden layer and output layer. the accuracy is 78.89 percent which is much lower than decision tree and KNN. The reason why its performance is lower is probably because that the dataset is large with more than 30000 datasets and adding more hidden layers or increasing the number of iteration cycles could improve the performance.

PerformanceVector

```
PerformanceVector:
accuracy: 78.89% +/- 6.94% (micro average: 78.89%)
ConfusionMatrix:
True:   short   very short      medium  long    very long
short:  315     56      120     25      7
very short:     244     1628    8       0       11
medium: 5       1       97      58      11
long:   0       0       0       0       0
very long:      0       0       0       0       0
weighted_mean_recall: 39.16% +/- 10.83% (micro average: 39.12%), weights: 1, 1, 1, 1, 1
ConfusionMatrix:
True:   short   very short      medium  long    very long
short:  315     56      120     25      7
very short:     244     1628    8       0       11
medium: 5       1       97      58      11
long:   0       0       0       0       0
very long:      0       0       0       0       0
weighted_mean_precision: 38.23% +/- 11.06% (micro average: 40.54%), weights: 1, 1, 1, 1, 1
ConfusionMatrix:
True:   short   very short      medium  long    very long
short:  315     56      120     25      7
very short:     244     1628    8       0       11
medium: 5       1       97      58      11
long:   0       0       0       0       0
very long:      0       0       0       0       0
absolute_error: 0.341 +/- 0.044 (micro average: 0.341 +/- 0.278)
relative_error: 34.13% +/- 4.42% (micro average: 34.13% +/- 27.79%)
```

Figure20. Neural Network Parameter Set

In conclusion, the team decided to choose KNN as our data product and as our model. The reason is because that its accuracy is much higher in comparison with decision tree and neural network, while its relative error is smaller. Thus, the team will use KNN algorithm in which the parameter of K is equal to 11 to classify patient's waiting time.

Lesson Learned and Contribution

Through this assignment, the team has learned more practical experiences.

Firstly, we learned how to put the knowledge of data mining, machine learning and statistics into practice. For instance, we learned to use machine learning algorithms like Neural network and K-NN to do the prediction. And we also learned or recalled some of our mathematic knowledge like how to calculate probabilities etc. This is a very good practice, we enjoyed it very much.

Secondly, we have learned how to collaborate with each other more efficiently. For instance, before we started this assignment, the team had several meetings to talk about the process and then divided the whole assignment into different section. After these meetings, everyone was on the same page and could concentrate on his or her own part. If we got any problems, we communicated with each other constantly and solved the problems together. Thirdly, preparation and research at the first-hand is very important. Like in the beginning, the team did not do a thorough research for the dataset, we wasted some time when we did data cleaning. These are the most important lessons we have learned.

This is a groups' work, and everyone was actively engaged in the whole process even each one has a different focus. And we worked together and helped each other to complete the assignment, so the team agreed to share the contribution and credits with each other equally even though the contribution table looks different.

Group Member	Contributions
	Yuan Wei concentrated on using RapidMiner to build the prediction models and did data cleaning in Rapid Miner and generated all sorts of reports.
Yuan Wei 500135	Get involved in all the meetings and actively participated in the whole assignment.
	Yuan Wei actively involved in each part of the assignment.
	Helped to review the writing report and verified each statistics report.
Fengmei Liu 497666	Fengmei mainly focused on the report part and she actively engaged in the data mining part using RapidMiner.
	Fengmei helped to check the models in RapidMiner and analyzed reports.
	Fengmei also actively involved in each process of the assignment