MGSC 661 - Final Project Report

A classification and clustering analysis of patient hospital stays

Sophie Courtemanche-Martel

Duncan Wang

Master of Management in Analytics

Desautels Faculty of Management,
McGill University

December 16, 2020

Contents

1 Introduction **1**

1.1 Background . . . 1

1.2 Objective . . . 1

2 Data Description **2**

2.1 Pre-processing . . . 2

2.2 Variable exploration and relationships . . . 3

2.3 Outlier detection and collinearity . . . 3

3 Model Selection and Methodology **4**

3.1 Classification Model . . . 4

3.1.1 Feature selection . . . 4

3.1.2 Model selection . . . 5

3.1.3 Model training . . . 5

3.2 Clustering Model . . . 6

3.2.1 Feature selection . . . 6

3.2.2 Cluster formation using K-Means . . . 6

4 Results and Discussion **7**

4.1 Classification of patients by length of hospital stay . . . 7

4.1.1 Accuracy and error rate . . . 7

4.1.2 Precision and recall of classification models . . . 7

4.1.3 Feature Importance . . . 8

4.2 Clustering of patients by patient-caretaker interactions . . . 8

4.3 Discussion and Managerial implications . . . 10

4.4 Limitations and Future steps . . . 11

4.4.1 Model interpretation . . . 11

4.4.2 Sample bias . . . 11

4.4.3 Feature validity . . . 12

5 Conclusion **12**

A Appendix **14**

A.1 Tables . . . 14

A.2 Graphs . . . 16

1 Introduction

1.1 Background

Over the last few decades, the prevalence of chronic diseases has considerably increased in developed countries and are now the leading causes of death and disability in the United States (US) [1]. Four of the most prominent chronic diseases, namely obstructive pulmonary disease, type 2 diabetes, cancer, and cardiovascular diseases, are closely linked to changing lifestyles and dietary habits, and are placing unprecedented burdens on healthcare systems. These complex and chronic illnesses have contributed most substantially to driving an increase in long-term hospital stays in recent years [2], and it is estimated that managing patient admissions and associated hospitals stays costs the the US over $377.5 billion each year [3]. Long term hospital stays risk overwhelming hospitals with limited capacities and patient resources, and research demonstrates that longer term stays also increase probability of hospital-acquired conditions [4]. As such, there is significant financial and social value in being able to predict the length of a patient's hospital stay, and in better understanding the utilization of hospital resources by patients. Being able to do so may facilitate hospitals in advance-allocating resources to accommodate patients, which in turn can enhance patient care outcomes, reduce the length of unnecessary stays, and produce cost savings which can be reinvested into improving health infrastructures.

Machine learning and data mining techniques have been increasingly used to draw insights from healthcare data, which can aid public health agencies and hospitals in managing human and financial resources. Past studies in the field have examined the prevalence of intensive care admissions, diagnoses, and their complications, and have demonstrated growing potential to deliver impacts in the clinical setting [5]. For the current project, the Medical Information Mart for Intensive Care (MIMIC) database provides a comprehensive stream of patient hospitalization records that have proven useful in past descriptive, predictive, and hypothesis-driven studies [6].

1.2 Objective

The objective of our analysis is to use data extracted from the MIMIC database to build two models, and to generate both predictive and exploratory insights from these models. The first model aims to predict the likelihood of a categorical outcome, being the length of a patient's stay. Based on a patient's characteristics and circumstances of admission, we will classify patients as likely to experience a short, medium, or long term stays. The second model aims to cluster patients based on the number of various patient-caretaker interactions – such as procedures, inputs taken, and drugs prescribed – which can quantify the amount of human or physical resources used by a patient during their stay. In all, we aim to combine two models to be able to explore both characteristics inherent to a patient's background and condition, and attributes associated with a patient's treatment process. By leveraging predictive analytics and by grouping patients by hospital resource consumption, we hope to generate insights that can better inform healthcare systems, allow for the proactive allocation of critical healthcare resources, and overall enhance patient care outcomes.

2 Data Description

The MIMIC III database is an openly available database developed by the Massachusetts Institute of Technology Computational Physiology lab. The database is comprised of anonymized entries for around 60,000 intensive care unit admissions at the Beth Israel Deaconess Medical Center, a teaching hospital of Harvard University in Boston, Massachusetts, containing more than 50,000 stays for adult patients and 8,000 neonatal patients recorded between June 2001 and October 2012.

For the purposes of the analysis, we utilized a subset of attributes from the MIMIC III database compiled by physician Dr. Alexander Scarlat, available on Kaggle [7]. This dataset contains information on patient demographics, admission details, patient condition, as well as information on the average daily number of various patient-caretaker interactions, such as drugs prescribed or number of clinical notes recorded for a patient. In addition, the data includes the total length of a patient stay and whether or the patient passed away while admitted. A complete description of variables can be found in the Appendix (5).

2.1 Pre-processing

Unless specified otherwise, all data preprocessing, visualization, and model building steps were performed in R Studio.

Variable cleaning and transformation Initial data exploration revealed that three categorical variables contained null values. Most categorical variables also contained a number of "unknown" or "unspecified" categories, thus null values and categories with no information were re-classified into a single group. Several categorical classes contained variable-encoding errors, including misspelled words and variations of the same category (i.e. "GI" vs. "Gastrointestinal"), which were likewise encoded into the same group. There were a number of high-cardinality categorical variables which were re-classified into broader categories based on intuition; for instance, for patient ethnicity, "White - European" and "White - Russian" were re-grouped as "White". In particular, there were over 15,000 unique patient diagnosis categories. Thus, to facilitate the feature selection process, we selected a number of the most common admission diagnoses as well as a few diagnoses of interest, and re-grouped the remaining diagnoses as "Other". Patient IDs were uniquely assigned to each admission, and the admission procedure variable was comprised of text descriptions, thus both were subsequently dropped from the analysis.

Target Variable Creation - Length of Stay (LOS) Length of stay was identified as a continuous variable ranging from 0-294 days. To create a categorical target for the classification task, length of stay was categorized into three classes with a comparable number of observations in each class:

1. Short stays: 0-5 days
2. Medium stays: 6-10 days
3. Long stays: greater than 10 days

2.2 Variable exploration and relationships

By reducing high-cardinality categorical variables into fewer classes, we were able to visualize variable categories using histograms and box-plots. These tools provided a preliminary analysis of the categorical frequencies for variables such as patient admission type, insurance type, religion, marital status, and ethnicity. By visualizing categorical predictors, we were able to infer for instance that the older patients more frequently experienced medium to longer stays, whereas younger patients had the shortest hospital stays (6). We also found that admissions to the hospital were characterized largely by high numbers of emergency admissions, but had low mortality rates. Most patients were covered by either Medicare or private insurance, and most patients were either neonatal or in middle-to-older age groups. Interestingly, we found that the majority of white patients were older compared to other ethnic groups (5).

2.3 Outlier detection and collinearity

Clustering models are distance-based and can thus be sensitive to the presence of outliers. As such, we applied the *outlierTest()* function to the numerical variables describing patient-caretaker interactions to identify anomalies. Although the latter observations can convey useful information regarding unusual patient cases or exceptions, we decided to remove them since their presence can significantly influence the position of cluster centroids and alter the overall formation of clusters. Furthermore, for continuous predictors, collinearity was assessed using the correlation matrix results derived from the correlation test *cor()* along with the Variance Inflation Factors (VIF) test. As expected, some patient-caretaker interaction variables showed high levels of collinearity, and were not included in the clustering model in order to avoid potential issues arising from multicollinearity. Figure (1) depicts pairs of variables with coefficients above 0.75 that were identified as being highly correlated. Visualization of the correlation matrix was also conducted using the *corrplot* library and results can be seen in the Appendix (3).

Table 1: Correlation coefficients in pairs correlated predictors

First predictor	Second predictor	Correlation Coefficient
Number of transfers	Number of diagnoses	0.78
Number of chart events	Number of notes	0.79
Number of notes	Number of transfers	0.79
Total number of interactions	Number of chart events	0.97

3 Model Selection and Methodology

Given the the richness of the data, we opted to build two types of models: a classification model and a clustering model. The classification model aims to classify patients into short, medium, and long term stays. The clustering model aims to identify patterns and recognize similarities amongst patients based on the number of patient-caretaker interactions observed throughout a stay.

3.1 Classification Model

3.1.1 Feature selection

In order to build a classification model that would feasibly predict rather than simply observe the length of stay of the patients, only predictors that were recorded upon admission of the patient into the hospital were included in the model. With the exception of patient age, the continuous predictors were found to describe treatment and care related events occurring during a patient's stay. Thus, continuous predictors describing measures, tests, and procedures performed or observed throughout the duration of a patient stay at the hospital were excluded. Including the latter would likely result in a high-performing model but would be misleading from a feature validity perspective.

Following the initial feature engineering process, the remaining variables considered for the classification task were categorical with the exception of age. To perform feature selection, we conducted Chi-Squared tests of independence using the *Scipy* package in Python, at a significance level of 0.05. Chi-Squared is a hypothesis test used to establish the presence of a significant relationship between two categorical variables, and was selected as the most optimal feature selection technique given that both the target variable and predictors are categorical in nature. We built contingency tables to test the presence of a significant relationship between the target category – length of stay (LOS), and each of the candidate predictors. For candidate predictors identified as significantly related to LOS, we used Bonferroni-adjusted post-hoc tests to perform pair-wise comparisons and identify which specific categories were significantly related to LOS. Self-payed insurance status was also removed at this stage as it was virtually unary. This reduced our categorical predictors from 50 to 38, which were dummified prior to model building. Age, which was observed to increase with the length of stay, was the only continuous predictor selected for modelling. A full list of predictors used for the classification models is in Appendix (6).

3.1.2 Model selection

The first step towards building the classification model was to identify the advantages and limitations of popular models given the classification task at hand. The various models considered included Linear Probability Model, Multinominal Logistic Regression models, Linear Discriminant Analysis, Quadratic Discriminant Analysis, as well as tree based models such as Classification Tree, Random Forests and Gradient Boosted Trees. Given the limitations of the linear probability model, namely lack of boundaries of the probabilities and linearity assumptions, this model was excluded from the model testing phase. Moreover, Linear and Quadratic Discriminant analysis models were also abandoned given that the predictors selected for classification were largely categorical. Given the multi-class nature of our classification task, we tested a Multinomial Logistic Regression (MLR) model rather than a simple logistic regression model. Finally, Random Forest (RF) and Gradient Boosting Machine (GBM) were two tree based models that were selected in favour of simple classification trees, which suffer from high-classification variance. Overall, the following models were built:

1. Multinominal Logistic Regression (MLR)
2. Random Forest (RF)
3. Gradient Boosted Trees (GBM)

3.1.3 Model training

For the selected models, the dataset was split randomly into training and test validation sets comprising of 70% and 30% of the observations, respectively. After training the models using the training dataset, predicted values were generated using the test dataset, whereby the models assigned a length of stay label to each test observation based on the highest probability of the predicted value. From the comparing the predicted classifications to the observed classifications in the test dataset, we were able to generate the test accuracy, error rate, and a 95% confidence interval for each prediction.

Confusion matrices were also generated to compare the predicted length of stay to the actual length of stay observed, and accuracy and recall was used to evaluate the performance of the model at predicting short, medium, versus long term stays. Recall (1) is a measure of the classifiers' completeness; the ability of the predictive model to identify correctly the length of stay of the patients (true positives) while Precision (2) is the measure of the correctly labeled observations over the total number of observations labeled as positives (true and false positives).

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

For relative comparison purposes, RF and GBM were trained at 200 trees. After initial comparison of the base models, it was found that GBM achieved the highest test-accuracy; thus, we further optimized the GBM model by performing hyper-parameter tuning. We specified different numbers of boosting iterations to build, maximum tree depths, and shrinkage parameters based on recommended ranges for datasets of a similar size, and performed a five-fold cross validation for each combination of hyper-parameters to find the optimal levels of each.

3.2 Clustering Model

3.2.1 Feature selection

We aimed to complement the results of the classification task by exploring unsupervised methods for clustering patients by the average daily number of various patient-caretaker interactions, as a measure of the quantity of human or physical resources consumed by patients during their stay. The clustering task had the given objective to potentially reveal hidden resemblances between patients by grouping patients with similar features into homogeneous groups, to examine which patients were potentially more costly. We chose to target a priority patient group of 1349 patients by selecting only patient admissions classified as "emergency", and which were diagnosed with one of the five most prevalent conditions at the hospital: **gastrointestinal bleed, coronary artery disease, pneumonia, sepsis, and congestive heart failure**. We subsequently selected eight quantitative variables to cluster patients by, which are measured on an average daily scale: **number of diagnoses made, number of procedures performed, number of inputs taken and outputs taken, number of labs and microbiology labs performed, number of drugs prescribed, and number of procedural events performed**. Due to different ranges of each variable, variables were scaled using min-max normalization before clustering.

3.2.2 Cluster formation using K-Means

We opted to use the K-Means method to cluster patients into a predetermined number of groups. To determine the optimal number of clusters, three different techniques were consulted: the elbow method, the silhouette method, and the *NbClust* library in R, which compares 30 indices for determining the number of clusters and proposes the best cluster number based on these indices. These metrics weigh clustering priorities differently, but aim to maximize variation between clusters while minimizing variation within each cluster. Using these methods, we identified three clusters as the optimal number. After assigning each patient to a cluster, Principal Component Analysis was applied to the dataset to perform dimension reduction in order to visualize the clusters by the top two principal components, and the characteristics of patients assigned to each cluster were plotted.

4 Results and Discussion

4.1 Classification of patients by length of hospital stay

4.1.1 Accuracy and error rate

The measures of the predictive performances of the models can be found in table (2). The GBM model marginally outperformed the two other models explored, with an accuracy of 45.71 %, and with a 95% confidence interval of: 0.4523-0.462. Using the ensemble approach, the GBM grows trees sequentially, with each tree grown using knowledge gained from previously grown trees to improve aggregate performance. Thus, we expected the boosting algorithm to outperform the other tested models. However, it should be noted that overall the performance of the classification models is relatively poor.

Table 2: Accuracy and error rate of classification models

Model	**Multinominal Logistic Regression**	**Random Forest**	**Gradient Boosting Machine**
Accuracy	43.41%	43.62%	45.71 %
Error rate	56.58%	56.37%	54.28 %
95% CI	(0.4293, 0.4389)	(0.4315, 0.441)	(0.4523, 0.462)

4.1.2 Precision and recall of classification models

The confusion matrices reveal that all models performed best at identifying short stays. However, it appears that this may be due to the tendency of the models to overwhelmingly predict short stays over any other class, as the models were much less likely to identify medium and long term stays when they did occur. When examining the prediction recall of the GBM model, we find that out of all short stays, 68.9% were identified as short stays, while only 27.4% of all medium stays and 40.9% of all long stays were identified correctly. Precision was more similar across classes. We observed that 48.6% of predictions classified as short stays were in fact short stays, while 44.4% of predicted medium stays and 41.6% of predicted long stays were accurately medium and long stays. This higher recall, lower precision nature of the GBM model in relation to short stays confirms our observation that the model is best at identifying patients with short stays, however still performs considerably poorly given the frequency of misclassified patients with medium and long stays.

Table 3: Confusion matrices of classification models

		Actual								
		MLR			RF			GBM		
		Short	Medium	Long	Short	Medium	Long	Short	Medium	Long
Predicted	Short	9745	4931	5474	12915	8173	8792	10255	5508	5342
	Medium	2625	3664	2297	1837	2899	1567	2291	3479	2074
	Long	3947	4086	4783	1295	1609	2195	3501	3694	5138

4.1.3 Feature Importance

To determine which predictors played the largest roles in classifying patient stays, we used three different approaches to determine the relative feature importances of the Multinominal LR, Random Forest and GBM models, as each defines its own specificity in terms of feature importance. For the MLR model, the *VarImp()* function was used to assess the sum of the absolute value of the coefficients of the variables of the model. For the RF model, graphical interpretation of the *VarImpPlot()* dotchart was conducted. As for GBM, the *summary()* function outputs a graph with the relative influence of the predictors. Table (4) outlines the top 4 predictors of the three models, and suggest that overall, age, admission location, and admission diagnosis were most important in determining length of stay.

Table 4: Top four most important predictors for classification models

Rank	**MLR**	**RF**	**GBM**
1	Admit diagnosis: Liver failure	Age	Age
2	Admit location: Transfer	Admit location:Physician Referral	Admit location: Unknown
3	Admit location: Unknown	Admit location: Clinic Referral	Ethnicity: Unknown
4	Admit diagnosis: Overdose	Admit diagnosis: Coronary Heart Disease	Insurance: Government

4.2 Clustering of patients by patient-caretaker interactions

With three clusters, the total sum of squares variation – calculated as the variation between clusters relative to the total variation – was 64.5%, representing a total measure of variance in the patients which can be explained by the three clusters. The plot of clusters by the top two principal components reveals cluster 3 represents the vast majority of patients, who experienced comparably lower average daily numbers of patient-caretaker interactions and with a relatively low degree of variation. Meanwhile, patients assigned to cluster 2 show a higher variation in patient-caretaker interactions, while cluster 1 represents patients with the highest overall number of and variation in patient-caretaker interactions.

When examining the relative frequency of patient characteristics within each cluster, we find that cluster 1 consists entirely of patients with short-stays, the majority of whom died during their stay. Compared to the other clusters, patients in this group also had the highest relative proportion of emergency room admissions, and the most common diagnosis upon admission in this group was sepsis. Meanwhile, cluster 3 mostly represent patients with a longer length of stay, low mortality, and the lowest relative frequency of emergency room admissions. Cluster 2 represents somewhat of an “intermediate” state between cluster 1 and 3, and contains characteristics of both clusters. While we also compared the insurance status, ethnicity, and age of patients across these clusters, we did not find that any of the clusters were distinguished by these factors.

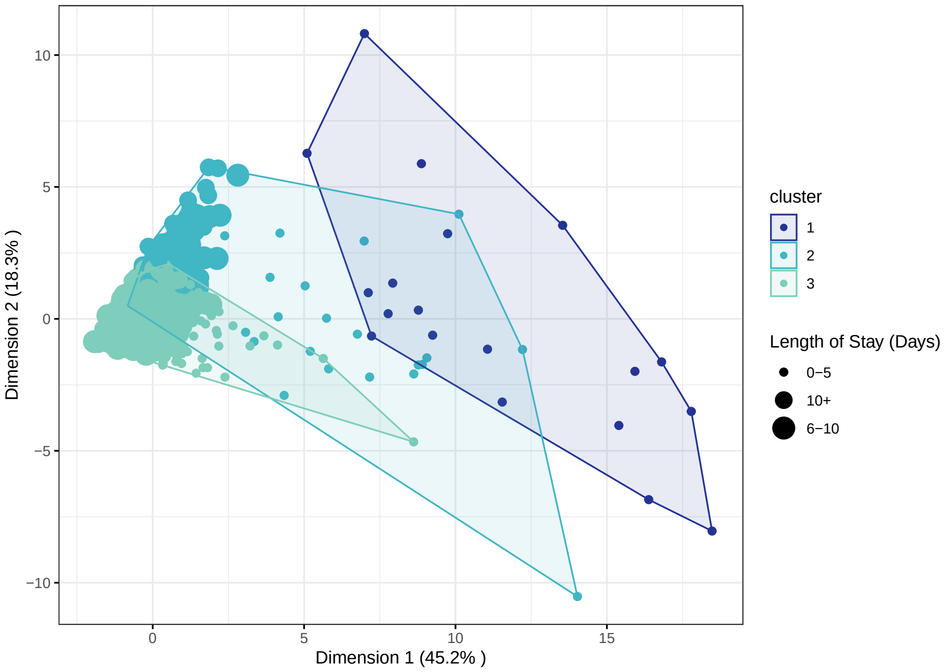

Figure 1: Results of K-Means clustering of patients by patient-caretaker interactions

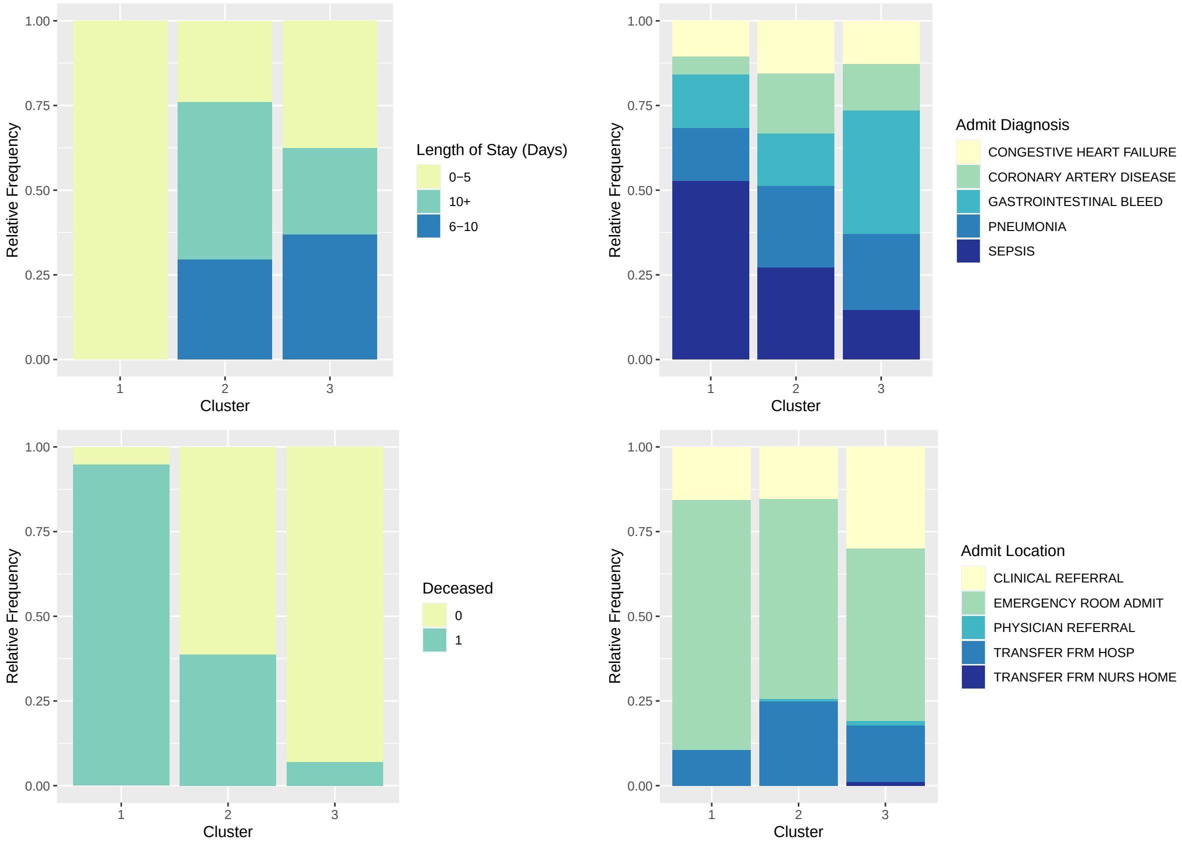

Figure 2: Relative frequency of patient traits clustered by patient-caretaker interactions

Overall, it is important to note that the cluster formation based on the number of patient-caretaker interactions is relatively poor. We observed that there were just 19 patients assigned to cluster 1, while 129 patients were assigned to cluster 2, and the vast majority of patients – 1,201 – were assigned to cluster 3. While there are certain characteristics differentiating the clusters, there remains a significant degree of overlap between patients in these groups.

4.3 Discussion and Managerial implications

The results of our classification analyses reveal that it is difficult to predict the length of a hospital stay based on patient characteristics, diagnosis, and admission circumstances alone. This is unsurprising, given that the length of a hospital stay may be determined by many factors exogenous to the patient, such as the availability of physicians, specialized equipment, and the effectiveness of managing patient cases. Teaching hospitals such as the Beth Israel Deaconness Medical Center invest significant funds into medical research and towards the training of medical interns and fellows, and patients may be used as teaching tools. It is therefore possible that patient stays may be extended due to extenuating circumstances, such as if extra laboratory tests are ordered, or if the treatment of an initial condition leads to the discovery of additional conditions requiring treatment. Our classification analysis revealed that the models were most successful at predicting short stays, but over-estimated their likelihood and thus performed poorly at identifying medium and long term stays. Using our current model, hospitals would most likely underestimate the duration of patient stays, which could lead to an inadequate preparation of beds, staff, and resources to handle realized capacity needs. Given the high-consequential risk of misinformation in the context of healthcare resource allocation and patient care, we do not recommend that hospitals make decisions based on patient demographic characteristics and admission condition alone.

Our K-Means clustering model revealed a few key insights. Namely, for patients admitted under one of the top five most common emergency conditions, patients in cluster 1 had the had the highest number of and variability in patient-caretaker interactions and would potentially incur the highest daily “cost” to the hospital. These patients were most likely to have been diagnosed with sepsis, had the shortest stays, and most frequently died. This observation is logical given that sepsis is classified as a medical emergency which can rapidly progress to septic shock, triggering tissue damage and which often leads to fatal and complete organ failure [8]. The large variation in patient-caretaker interactions in this cluster of patients could be driven by the dominance of high-fatality conditions whereby a patient either died before treatment, or died while being treated, although we cannot definitively determine the causality of such an observation.

We found that the majority of patients in the clusters 2 and 3 had a much smaller and less variable number of patient-caretaker interactions, and greater variability in the lengths of their stays. It could be that the number of interactions may be initially high immediately following admission, but eventually plateau once the patient is stabilized. Patients in these groups were also less-likely to die, and were more likely to be diagnosed with conditions such as gastrointestinal bleed, coronary artery disease, and congestive heart failure. These conditions have varying levels of severity, but are less likely to be immediately life-threatening.

Overall, we were able to identify a few key traits associated with the number of patient-caretaker interactions, which mainly concerned a patient's condition rather than a patient's demographic traits. The distribution of clusters suggest that there are few patients which may require more resources on an average daily basis, but that the majority of patients utilized a similar amount of resources. Due to the overlap in clusters, we suggest that the number of patient-caretaker interactions may be influenced by factors outside of the scope of this analysis – such as patient history and condition severity – and is likely to vary on a case-by-case basis.

4.4 Limitations and Future steps

4.4.1 Model interpretation

Applying predictive analytics to the healthcare setting can be quite challenging, given that the mathematical modeling of patient outcomes must take into consideration both the generalizability of models as well as the substantial variation in treatment costs and outcomes even amongst similar patient groups. Patients are admitted with conditions requiring varying treatment procedures, and admitted into different units with different capacities and resources. Thus, to quantify the cost of different lengths of stay, it may be useful to segment patients by diagnosis and admission location. Since data used was also collected over an 11 year period, restructuring of hospital programs, units, and divisions may have occurred over time, thus further segmentation into comparable time periods may be necessary.

Our cluster model utilized variables such as number of diagnoses, number of procedures, and number of laboratory exams performed to cluster patients, thus we were able to estimate which patient groups consumed the highest relative number of average daily resources. However, since an interaction such as taking patient vitals is considerably less costly than performing a surgical procedure, it would be difficult for us to quantify the dollar-cost of each patient group to hospitals based on our results alone. While the cluster model was exploratory in nature, it did not result in perfectly heterogeneous clusters, and interpretation of the results is limited to a few select features.

4.4.2 Sample bias

It should be noted that the data used for the analysis consists of admissions to one hospital – the Beth Israel Deaconness Medical Center, a teaching hospital of Harvard University and one of the most highly regarded hospitals in the United States. The patients we analyzed were predominantly white, either covered by private insurance or Medicare (federal health coverage for patients 65 years and older), with the median age of all patients being 59 years old. Interestingly, we found that 70% of patients were classified as emergency room admissions, and over 10% of admitted patients ultimately died, compared to the US average of 1.48-0.77 deaths per 1,000 emergency room admissions [9]. It is evident from the patient profile of the dataset that this sample of patients is not representative of the wider demographic composition of patients across the United Status, thus care must be taken not to extrapolate the results of our analysis to the wider US population.

4.4.3 Feature validity

One of the major limitations of the classification model is the lack of comprehensive measures. By prioritizing feature validity over model accuracy, we found ourselves left with six categorical predictors (admission type, location and diagnosis, religion, marital status, ethnicity), one binary predictor (gender male) and only one continuous predictor (age). While the MIMIC III database offers more detailed information on the patient-caretaker interactions we considered, we chose only to utilize the average quantity of these measures rather than the results of such tests, procedures, and patient notes. In order to improve the performance of the classification model, critical metrics recorded upon the time of admission such as Body Mass Index (BMI), heart rate, temperature, nervous reflexes, as well as general indicators such as pain levels and pre-existing medical conditions, could be incorporated into the model to improve the predictive power of the classification models.

5 Conclusion

One of the central aims of this project was to create models that would bring value in understanding underlying patterns of hospitalizations. Using retrospective data from the MIMIC III dataset, we were able to develop classification and clustering models to shed light on the potential underlying factors that can influence the length of a patient's stay. We were able to compare the performance of three different predictive classification models and concluded that given the current limitations of the dataset, the present models cannot be successfully used to predict a patient's length of stay. On the other hand, the current data is more suited for exploratory purposes, as we were able to describe three groups of patients on the basis of hospital resource utilization, as well as their characteristics, diagnoses, and potential causes of mortality during their stay.

This project allowed us to better understand the complexity of analytical tools that are used in health care analytics as well as the implications these models will have in practice. Predictive modelling shows great potential for contributing to advances in human resources management, prognostic and diagnostic analysis, and even for the early detection of disease and disability. However, in order to operationalize predictive models for such tasks, data collection, algorithm tuning, and model interpretation must be undertaken in a manner which ultimately furthers the interest of the individuals, communities, and populations they are designed to serve. Such models can have large benefits, but must be approached with care when human lives are on the line.

References

[1] World Health Organization (WHO), "Integrated chronic disease prevention and control," 2020.

[2] Friedman, B., Jiang, H. J., Elixhauser, A., Segal, A., "Hospital inpatient costs for adults with multiple chronic conditions," *Medical care research and review*, vol. 63, no. 3, pp. 327–346, 2006.

[3] Health Catalyst, "Patient-centered los reduction initiative improves outcomes, saves costs," 2016.

[4] Hassan M., David K., "Hospital length of stay and probability of acquiring infection," *International Journal of Pharmaceutical and Healthcare Marketing*, vol. 4, no. 4, pp. 324–338, 2010.

[5] Panch, T., Szolovits, P., Atun, R., "Artificial intelligence, machine learning and health systems.," *Journal of global health*, vol. 8, no. 2, 2018.

[6] Alistair E.W., Johnson A., Pollard T., Shen L., Lehman H., Feng M., Mohammad G., Moody B., Szolovits P., Celi L., Mark R., "Mimic iii, a freely accessible critical care database," *Scientific Data*, 2016.

[7] Scarlat, A., "Aggregated mimic iii data," 2019.

[8] Center for Health and Diseases (CDC), "What is sepsis?," 2020.

[9] Shmerling, R., "Where people die," 2018.

A Appendix

A.1 Tables

Table 5: Predictors Descriptions

MIMIC III Data Description		
Variable Name	**Type**	**Description**
LOS days	Continuous	Length of stay in days across all units
LOSgroupNum	Categorical	Categorized length of stay
Categorical predictors		
Gender	Binary	Gender of the patients (Male or Female)
Admit type	Categorical	Type of admission (i.e. Emergency, elective)
Admit Diagnosis	Categorical	Diagnosis upon admission
Insurance	Categorical	Insurance type of the patients
Religion	Categorical	Religious belief of the patients
Marital status	Categorical	Marital status
Ethnicity	Categorical	Ethnicity
Admit procedure	Categorical	Procedure upon admission
Expired hospital	Binary	Flag indicating if a patient has died during stay
Continuous predictors		
Age	Continuous	Age of the patients (in years)
Num callouts	Continuous	Avg. daily number of callouts for consultation
Num procedures	Continuous	Avg. daily number of procedures performed
Num diagnoses	Continuous	Avg. daily number of diagnosis made during stay
Num drugs prescribed	Continuous	Avg. daily number of drugs prescribed
Num CPT events	Continuous	Avg. daily number of events recorded in current procedural terminology code
Num inputs	Continuous	Avg. daily number of events related to fluid inputs for patients (i.e. IV)
Num labs	Continuous	Avg. daily number of events relating to laboratory tests
Num micro labs	Continuous	Avg. daily number of microbiology tests
Num notes	Continuous	Avg. daily number of notes associated with hospital stay (nursing, MD notes, radiology etc)
Num outputs	Continuous	Avg. daily number number of outputs and measurements (i.e. HR, IV line change)
Num transfers	Continuous	Avg. daily number of transfers of patients to different locations within the hospital
Num chart events	Continuous	Avg. daily number of events occurring on a patient chart
Total num interactions	Continuous	Total aggregated number of interactions of any type during the stay (summary of above)

Table 6: Selected features for Classification model

Variable Name	Classes
Age	(continuous predictor)
Gender Male	(binary predictor)
Admit type	Elective, Emergency, Newborn, Urgent
Admit location	Clinical referral, Emergency room, HMO referral, Physician referral, Transfer from hospital
Admit diagnosis	Congestive heart failure, Coronary artery disease, Diabetic Ketoacidosis, Fever, Gastrointestinal bleed, Intracranial hemorrhage, Liver failure, Newborn, Overdose, Pneumonia, Sepsis, Other
Religion	Catholic, Unspecified
Marital Status	Married, Divorced, Widowed, Unknown
Ethnicity	African American, Asian, Hispanic or Latino, White, Unknown

Table 7: Selected features for Clustering model

Variable Name	Variable Range
Num diagnoses	0 to 450
Num procedures	0 to 275
Num inputs	0 to 6825
Num outputs	0 to 375
Num labs	0 to 5175
Num microbiology labs	0 to 375
Num drugs prescribed	0 to 750
Num procedural events	0 to 100

A.2 Graphs

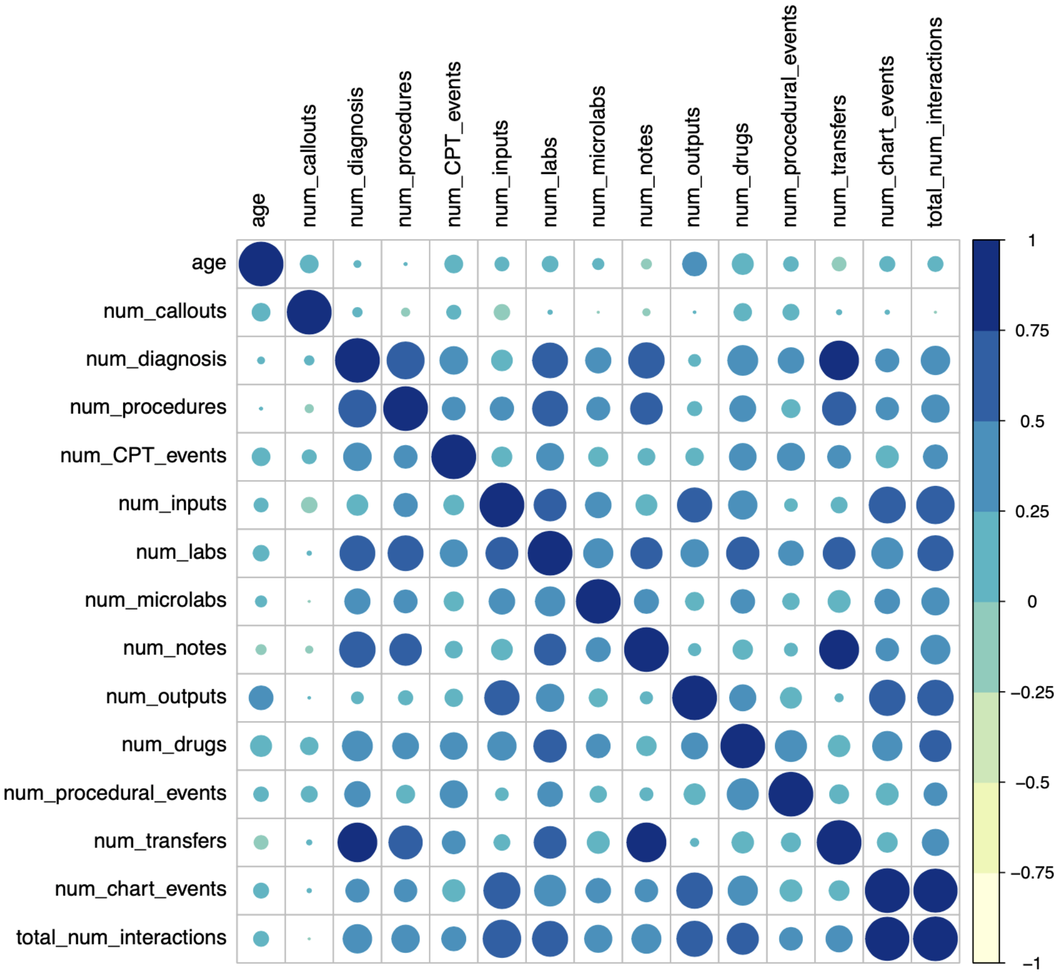

Figure 3: Correlation matrix for continuous predictors

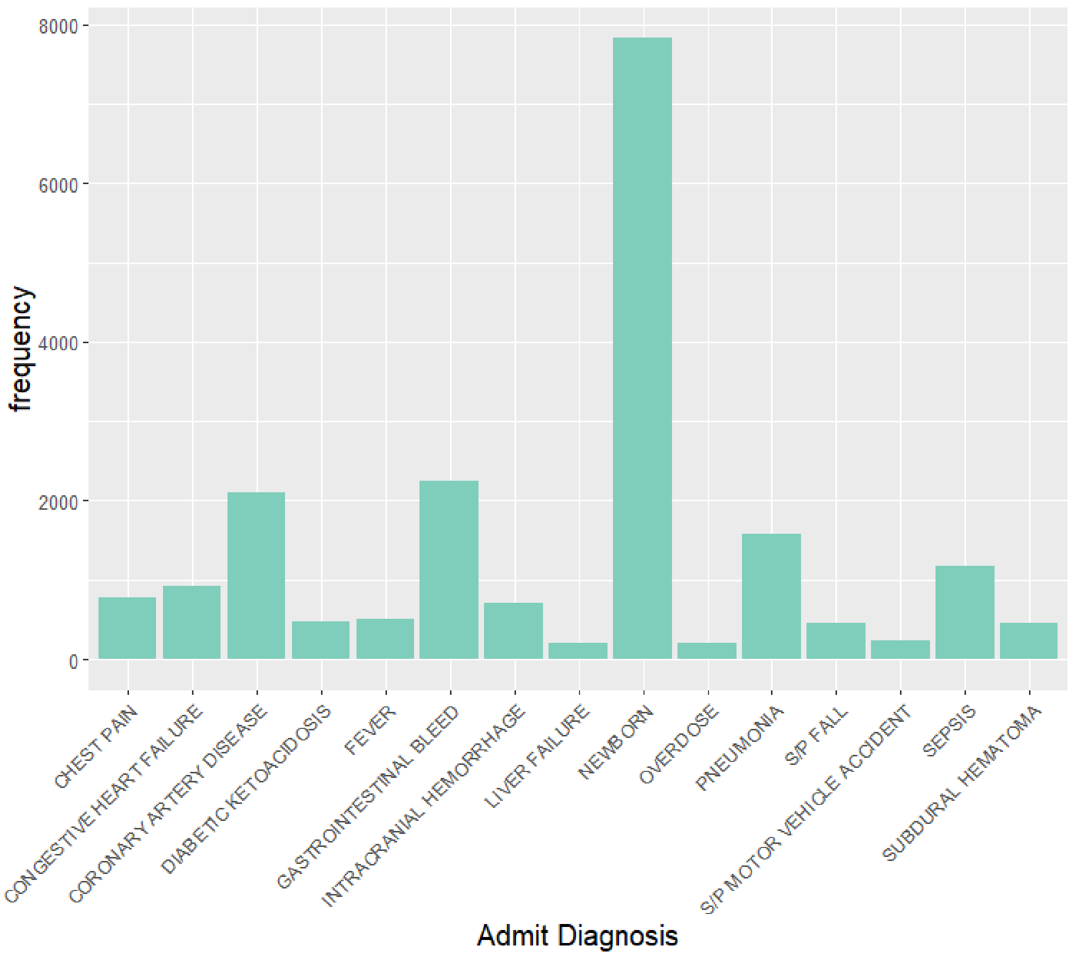

Figure 4: Admission diagnoses considered for classification

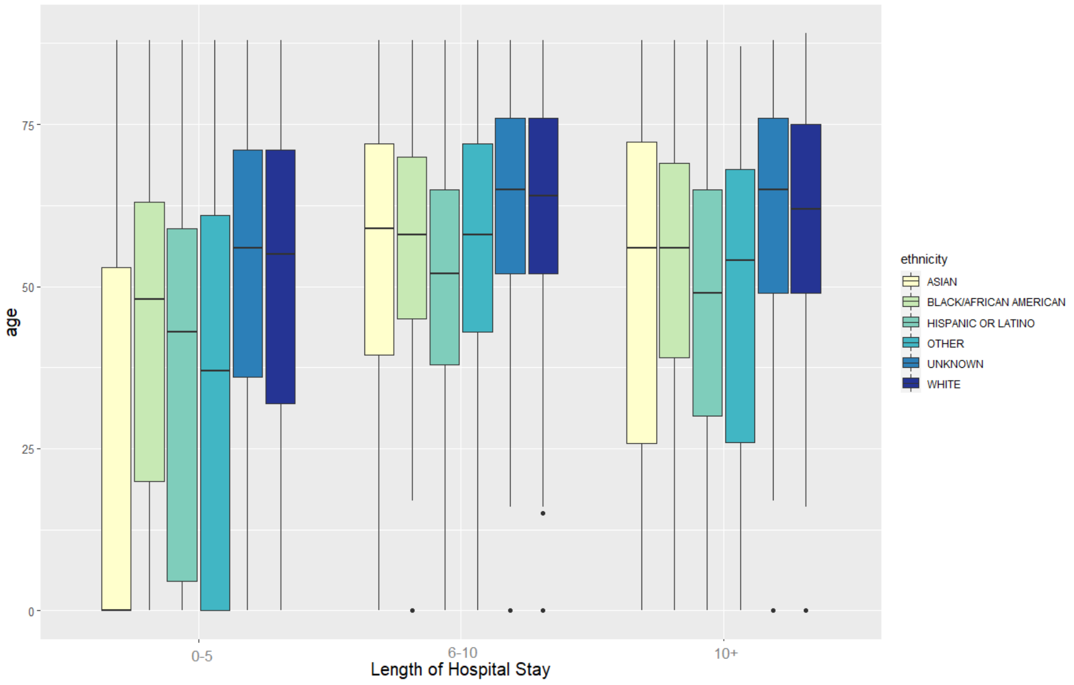

Figure 5: Length of patient stay by age and ethnicity

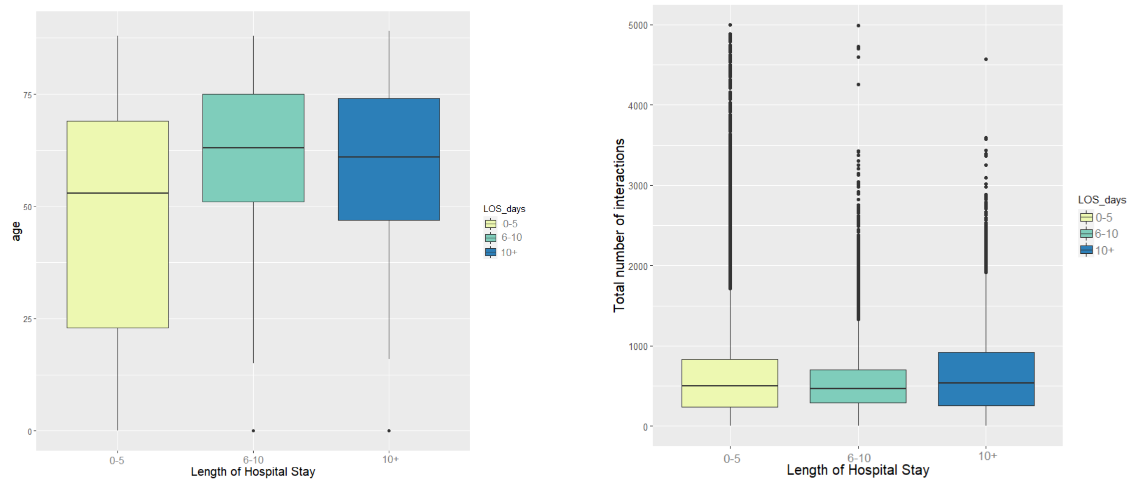

Figure 6: Length of patient stay by age & total number of patient-caretaker interactions

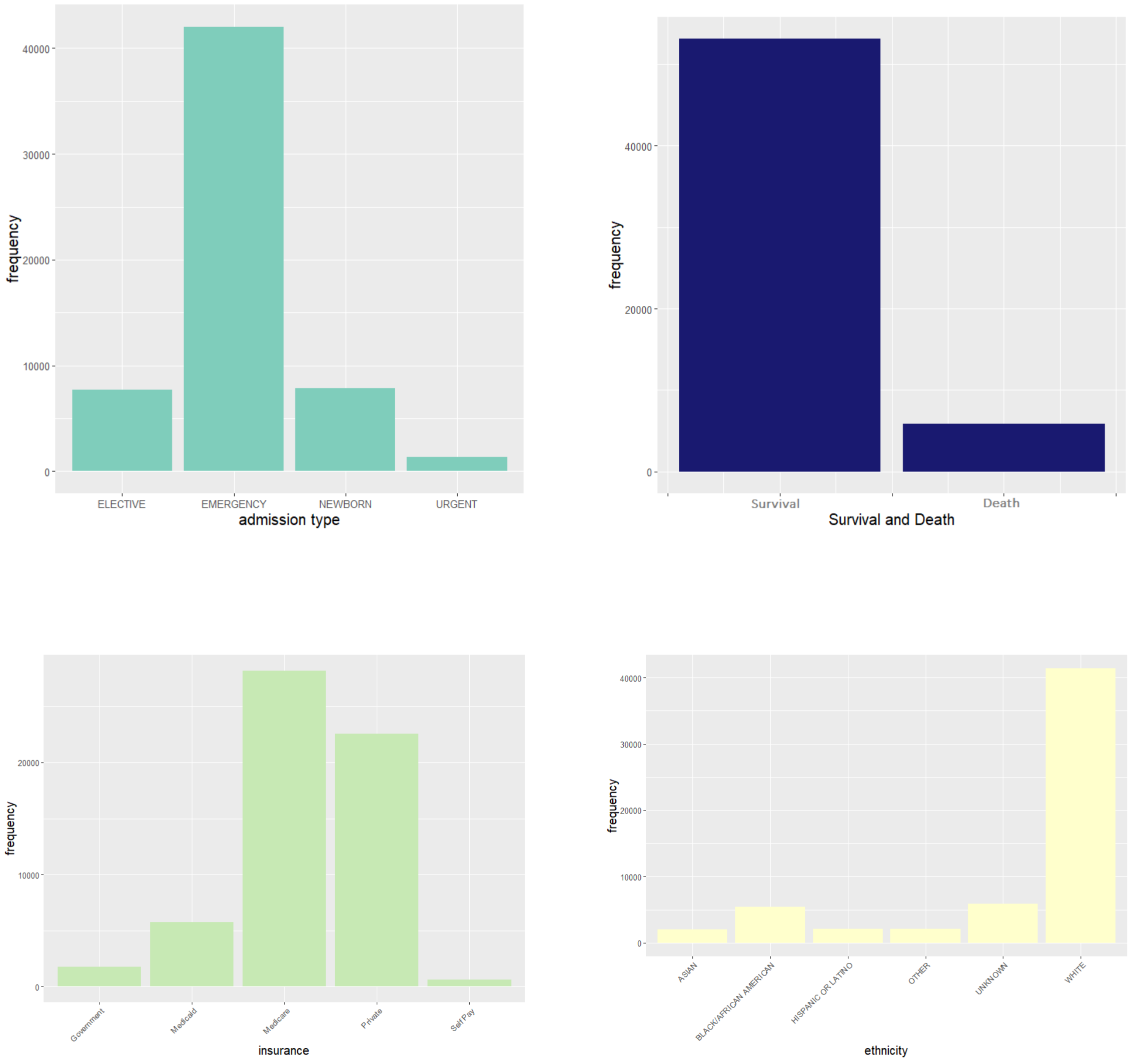

Figure 7: Distribution of patients characteristics

Pre-processing, Feature Selection and Model Code

Data Cleaning and Pre-processing

```
#Load libraries
install.packages('caret')
install.packages('generics')
library(readr)
library(car)
library(boot)
library(stargazer)
library(MASS)
library(klaR)
library(gbm)
library(caret)
library(gbm)
library(randomForest)
require(lmtest)
require(plm)
require(visreg)
require(dplyr)
require(psych)
require(ggplot2)
require(ggpubr)
require(methods)
require(caTools)
require(foreign)
require(nnet)
require(reshape2)
library(e1071)


##### READ THE DATA
hospital_data <- read_csv("C:/mimic3d.csv")
attach(hospital_data)


#rename hospital data to consistent format
hospital_data <- hospital_data%>%
  rename(
    patient_id = hadm_id,
    LOS_days = LOSdays,
    admit_diagnosis = AdmitDiagnosis,
    num_callouts = NumCallouts,
    num_diagnosis = NumDiagnosis,
    num_procedures = NumProcs,
    admit_procedure = AdmitProcedure,
    num_CPT_events = NumCPTevents,
    num_inputs = NumInput,
    num_labs = NumLabs,
    num_microlabs = NumMicroLabs,
    num_notes = NumNotes,
    num_outputs = NumOutput,
    num_drugs = NumRx,
    num_procedural_events = NumProcEvents,
    num_transfers = NumTransfers,
    num_chart_events = NumChartEvents,
    expired_hospital = ExpiredHospital,
    total_num_interactions = TotalNumInteract,
    LOS_group = LOSgroupNum
  )
attach(hospital_data)

#######STEP 1: NULL VALUES##########
#note that missing values from the continuous variables have already been
    imputed with the average by the creator
lapply(hospital_data,function(x) { length(which(is.na(x)))})
#admit diagnosis, religion, and marital status have null values and will
    be encoded as their respective "unknown" grouping category
hospital_data$admit_diagnosis[is.na(hospital_data$admit_diagnosis)] <-
    'OTHER'
hospital_data$religion[is.na(hospital_data$religion)] <- 'NOT SPECIFIED'
hospital_data$marital_status[is.na(hospital_data$marital_status)] <-
    'UNKNOWN (DEFAULT)'

#######STEP 2: DROP NON-PREDICTIVE VARIABLES##########
#drop patient ID, a unique identifier categorical variable
hospital_data = select(hospital_data, -patient_id)
#drop admit procedure
hospital_data = select(hospital_data, -admit_procedure)
#drop the pre-grouped LOS variable
hospital_data = select(hospital_data, -LOS_group)


#######STEP 3: RECODE TARGET VARIABLE ##########


#divide target into relatively equal sized bins, and regroup

length(LOS_days[LOS_days <= 5]) #23K
length(LOS_days[LOS_days > 5 & LOS_days <= 10]) #18K
length(LOS_days[LOS_days > 10]) #17K


hospital_data$LOS_days[hospital_data$LOS_days <= 5] <- 0
hospital_data$LOS_days[hospital_data$LOS_days > 5 & hospital_data$LOS_days
    <= 10] <- 1
hospital_data$LOS_days[hospital_data$LOS_days > 10] <- 2


#######STEP 4: RECODE CATEGORICAL VARIABLES INTO FEWER CLASSES##########
#drop patient ID, a unique identifier categorical variable

#recode admit location class data class
hospital_data$admit_location[hospital_data$admit_location == '** INFO NOT
    AVAILABLE **'] <- 'UNKNOWN'
hospital_data$admit_location[hospital_data$admit_location == 'TRSF WITHIN
    THIS FACILITY' | hospital_data$admit_location == 'TRANSFER FROM OTHER
    HEALT' | hospital_data$admit_location == 'TRANSFER FROM SKILLED NUR' |
    hospital_data$admit_location == 'TRANSFER FROM HOSP/EXTRAM'] <-
    'TRANSFER'
hospital_data$admit_location[hospital_data$admit_location == 'HMO
    REFERRAL/SICK'] <- 'PHYS REFERRAL/NORMAL DELI'
```

```
#recode religion into fewer classes
hospital_data$religion[hospital_data$religion == 'UNOBTAINABLE'] <- 'NOT
    SPECIFIED'
hospital_data$religion[hospital_data$religion != 'CATHOLIC' &
    hospital_data$religion != 'PROTESTANT QUAKER'& hospital_data$religion
    != 'JEWISH' & hospital_data$religion != 'NOT SPECIFIED'] <- 'OTHER'


#recode ethnicity into fewer classes -- classes are designed to minimize
    high-cardinality and create groups most representative of groups of
    interest in the USA.
hospital_data$ethnicity[hospital_data$ethnicity == 'ASIAN - CHINESE' |
    hospital_data$ethnicity == 'ASIAN - ASIAN INDIAN' |
    hospital_data$ethnicity == 'ASIAN - VIETNAMESE' |
    hospital_data$ethnicity ==  'ASIAN - FILIPINO' |
    hospital_data$ethnicity == 'ASIAN - CAMBODIAN' |
    hospital_data$ethnicity == 'ASIAN - OTHER' | hospital_data$ethnicity ==
    'ASIAN - KOREAN' | hospital_data$ethnicity == 'ASIAN - JAPANESE' |
    hospital_data$ethnicity ==  'ASIAN - THAI'] <- 'ASIAN'
hospital_data$ethnicity[hospital_data$ethnicity == 'WHITE - RUSSIAN'|
    hospital_data$ethnicity ==  'WHITE - OTHER EUROPEAN' |
    hospital_data$ethnicity == 'WHITE - BRAZILIAN'| hospital_data$ethnicity
    == 'WHITE - EASTERN EUROPEAN' | hospital_data$ethnicity == 'PORTUGUESE'
    ] <- 'WHITE'
hospital_data$ethnicity[hospital_data$ethnicity == 'HISPANIC/LATINO -
    PUERTO RICAN' | hospital_data$ethnicity == 'HISPANIC/LATINO -
    DOMINICAN' | hospital_data$ethnicity == 'HISPANIC/LATINO - GUATEMALAN'
    | hospital_data$ethnicity == 'HISPANIC/LATINO - CUBAN' |
    hospital_data$ethnicity ==  'HISPANIC/LATINO - SALVADORAN'|
    hospital_data$ethnicity ==  'HISPANIC/LATINO - CENTRAL AMERICAN
    (OTHER)' | hospital_data$ethnicity == 'HISPANIC/LATINO - MEXICAN' |
    hospital_data$ethnicity == 'HISPANIC/LATINO - COLOMBIAN'|
    hospital_data$ethnicity == 'HISPANIC/LATINO - HONDURAN'] <- 'HISPANIC
    OR LATINO'
hospital_data$ethnicity[hospital_data$ethnicity == 'AMERICAN INDIAN/ALASKA
    NATIVE' | hospital_data$ethnicity == 'AMERICAN INDIAN/ALASKA NATIVE
    FEDERALLY RECOGNIZED TRIBE' |hospital_data$ethnicity == 'BLACK/AFRICAN'
    | hospital_data$ethnicity == 'BLACK/CAPE VERDEAN' |
    hospital_data$ethnicity == 'BLACK/HAITIAN' | hospital_data$ethnicity ==
    'CARIBBEAN ISLAND' | hospital_data$ethnicity == 'MIDDLE EASTERN' |
    hospital_data$ethnicity == 'MULTI RACE ETHNICITY' |
    hospital_data$ethnicity == 'NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER'
    | hospital_data$ethnicity == 'SOUTH AMERICAN'] <-'OTHER'
hospital_data$ethnicity[hospital_data$ethnicity == 'UNABLE TO OBTAIN' |
    hospital_data$ethnicity == 'UNKNOWN/NOT SPECIFIED' |
    hospital_data$ethnicity == 'PATIENT DECLINED TO ANSWER'] <- 'UNKNOWN'

#recode marital status into fewer classes
hospital_data$marital_status[hospital_data$marital_status == 'LIFE
    PARTNER'] <- 'MARRIED'
hospital_data$marital_status[hospital_data$marital_status == 'UNKNOWN
    (DEFAULT)'] <- 'UNKNOWN'
hospital_data$marital_status[hospital_data$marital_status == 'DIVORCED' |
    hospital_data$marital_status == 'SEPARATED'] <- 'DIVORCED OR SEPARATED'

#recode admission diagnoses into fewer classes
#correct encoding errors & group diagnoses by generalized category
```

```
hospital_data$admit_diagnosis[hospital_data$admit_diagnosis == 'CORONARY
   ARTERY DISEASE\\CORONARY ARTERY BYPASS GRAFT /SDA' |
   hospital_data$admit_diagnosis ==  'CORONARY ARTERY DISEASE\\CORONARY
   ARTERY BYPASS GRAFT/SDA' | hospital_data$admit_diagnosis == 'CORONARY
   ARTERY DISEASE\\CATH'] <- 'CORONARY ARTERY DISEASE'
hospital_data$admit_diagnosis[hospital_data$admit_diagnosis == 'UPPER GI
   BLEED' | hospital_data$admit_diagnosis == 'LOWER GI BLEED' |
   hospital_data$admit_diagnosis == 'GI BLEED'  |
   hospital_data$admit_diagnosis == 'UPPER GASTROINTESTINAL BLEED' |
   hospital_data$admit_diagnosis == 'LOWER GASTROINTESTINAL BLEED'] <-
   'GASTROINTESTINAL BLEED'
hospital_data$admit_diagnosis[hospital_data$admit_diagnosis == 'ACUTE
   SUBDURAL HEMATOMA'] <-'SUBDURAL HEMATOMA'

#take top 30 hospital admission diagnoses and code the rest as OTHER
hospital_data$admit_diagnosis[hospital_data$admit_diagnosis != 'NEWBORN' &
   hospital_data$admit_diagnosis != 'PNEUMONIA' &
   hospital_data$admit_diagnosis != 'SEPSIS' &
   hospital_data$admit_diagnosis != 'CONGESTIVE HEART FAILURE' &
   hospital_data$admit_diagnosis != 'CORONARY ARTERY DISEASE' &
   hospital_data$admit_diagnosis != 'CHEST PAIN' &
   hospital_data$admit_diagnosis != 'INTRACRANIAL HEMORRHAGE' &
   hospital_data$admit_diagnosis !='ALTERED MENTAL STATE' &
   hospital_data$admit_diagnosis != 'GASTROINTESTINAL BLEED' &
   hospital_data$admit_diagnosis != 'FEVER' &
   hospital_data$admit_diagnosis != 'S/P FALL' &
   hospital_data$admit_diagnosis != 'LIVER FAILURE' &
   hospital_data$admit_diagnosis != 'OVERDOSE'&
   hospital_data$admit_diagnosis != 'S/P MOTOR VEHICLE ACCIDENT'&
   hospital_data$admit_diagnosis != 'DIABETIC KETOACIDOSIS' &
   hospital_data$admit_diagnosis != 'SUBDURAL HEMATOMA'] <- 'OTHER'

View(hospital_data)
attach(hospital_data)

write.csv(hospital_data, 'hospital_data.csv')
```

Feature Selection

The feature selection was done in Python

```
#######  Feature Selection Using Chi-Squared  #########

import pandas as pd

#import data
df = pd.read_csv('hospital_data.csv')

#drop invalid predictors that cannot be observed at the time of hospital
    visit
df.drop(['num_callouts','num_diagnosis','num_procedures','num_CPT_events','num_inputs','
    axis = 1, inplace = True)

#we will perform a cross tabulation between all levels of all variables
    and all levels of the target
pd.crosstab(df['admit_type'], df['LOS_days'])

#perform chi-squared tests on categorical variables
categorical_columns = df.drop(['LOS_days','age'], axis = 1).columns
chi2_check = []
for i in categorical_columns:
    if chi2_contingency(pd.crosstab(df['LOS_days'], df[i]))[1] < 0.05:
        chi2_check.append('Reject Null Hypothesis')
    else:
        chi2_check.append('Fail to Reject Null Hypothesis')
res = pd.DataFrame(data = [categorical_columns, chi2_check]
             ).T
res.columns = ['Column', 'Hypothesis']

#perform bonferroni adjusted pair-wise comparisons as post-hoc tests to
    see which levels of the significant variables are related to LOS days
#this produces the final list of categorical predictors
check = {}
for i in res[res['Hypothesis'] == 'Reject Null Hypothesis']['Column']:
    dummies = pd.get_dummies(df[i])
    bon_p_value = 0.05/df[i].nunique()
    for series in dummies:
        if chi2_contingency(pd.crosstab(df['LOS_days'],
           dummies[series]))[1] < bon_p_value:
            check['{}-{}'.format(i, series)] = 'Reject Null Hypothesis'
        else:
            check['{}-{}'.format(i, series)] = 'Fail to Reject Null
               Hypothesis'
res_chi = pd.DataFrame(data = [check.keys(), check.values()]).T
res_chi.columns = ['Pair', 'Hypothesis']

#create a new list with significant facotrs to subset original dataset by
significant_chi = []
for i in res_chi[res_chi['Hypothesis'] == 'Reject Null
    Hypothesis']['Pair']:
    significant_chi.append('{}_{}'.format(i.split('-')[0],i.split('-')[1]))


#dummify variables
```

```
new_df = pd.get_dummies(data = df, columns = df.select_dtypes(exclude =
    'number').columns)

#only subset those selected by the chi squared analysis
dummified_vars = new_df[significant_chi]
dummified_vars['age'] = df['age']
dummified_vars['LOS_days'] = df['LOS_days']

#reformat columns
dummified_vars.columns =
    dummified_vars.columns.str.strip().str.lower().str.replace(' ', '_')

#export the selected columns as a CSV
dummified_vars.to_csv('hospital_data_factors.csv')
```

```
#########  CLASSIFICATION   #######

hospital_C <- read_csv("C:/hospital_data_factors.csv")

#set target levels as factor
hospital_C$LOS_days = as.factor(hospital_C$LOS_days)

#remove outliers for age
hospital_C = hospital_C[-c(20),]

#Remove index column and unary variable
drop <- c("X1", 'insurance_Self Pay') #insurance self_pay is almost
   unary thus should be removed
hospital = hospital_C[,!(names(hospital_C) %in% drop)]
attach(hospital)



## For performance evaluation: Train and test data
require(caTools)
set.seed(101)
sample = sample.split(hospital$los_days, SplitRatio = .70)
train_T = subset(hospital, sample == TRUE)
test_T  = subset(hospital, sample == FALSE)
write.csv(train_T, 'train_data.csv')
write.csv(test_T, 'test_data.csv')

train_dataset = read_csv("C:/train_data.csv")
test_dataset = read_csv("C:/train_data.csv")


#########################################################
############## Model 1: Logistic regression #############

## Logistic regression
multiLR = multinom(LOS_days ~ . , data = hospital)
summary(multiLR)

#Variable importance
var_MLR <- varImp(multiLR, scale = FALSE)

#summary of model in stargazer
stargazer(multiLR, type = "html", out = "multiLR.htm")
z <- summary(multiLR)$coefficients/summary(multiLR)$standard.errors


## EVALUATION OF THE MODEL
#model
multinom_train = multinom(los_days~., data = train_dataset)

#predict new observations
pred = predict(object = multinom_train, newdata = test_dataset, type =
   "probs")
```

```
#Labels
labels = colnames(pred)[apply(pred, 1, which.max)]
result = data.frame(labels, test_dataset$los_days)

result$labels = as.factor(result$labels)
result$test_dataset.los_days = as.factor(result$test_dataset.los_days)

#Compare predicted values with the actual values in the test dataset
matrix = confusionMatrix(result$labels,result$test_dataset.los_days)
matrix


#calculate error
count = 0
L = length(result$labels)
for (i in 1:L){
  if (result$labels[i] == result$test_dataset.los_days[i]){
    count = count + 0
  }
  else{
    count = count + 1}
}

error_rate = count/L
count
error_rate


##########################################
######## Model 2: Random Forest ##########

#RANDOM FOREST
attach(hospital)
forest = randomForest(LOS_days~., data = hospital, ntree = 200)
forest
varImpPlot(forest, n.var = 5)


## EVALUATION OF THE MODEL
set.seed(1234)
# Run the model
train_dataset$los_days = as.factor(train_dataset$los_days)

rf_default <- train(los_days~.,
                    data = train_dataset,
                    method = "rf",
                    metric = "Accuracy",
                    ntree = 200)
# Print the results
print(rf_default)
varImpPlot(rf_default, n.var =4)

#Evaluate the model
predicted_values = predict(rf_default, newdata = test_dataset)


#Labels
```

```
results_RF = data.frame(predicted_values, test_dataset$los_days)

results_RF$predicted_values = as.factor(results_RF$predicted_values)
results_RF$test_dataset.los_days =
    as.factor(results_RF$test_dataset.los_days)

#Confusion Matrix
Matrix_RF = confusionMatrix(results_RF$predicted_values,
    results_RF$test_dataset.los_days)
Matrix_RF


#calculate error
count = 0
L = length(results_RF$predicted_values)
for (i in 1:L){
  if (results_RF$predicted_values[i] ==
      results_RF$test_dataset.los_days[i]){
    count = count + 0}
  else{
    count = count + 1}}

error_rate = count/L
count
error_rate



#####################################################
################# Model 3: Gradient Boost ###########

#GRADIENT BOOST
set.seed(1234)
boosted = gbm(los_days~., data = train_dataset,
              distribution = "multinomial",
              n.trees = 200,
              interaction.depth = 7,
              shrinkage = 0.01,
              n.minobsinnode = 10,
              cv.folds = 5)

summary(boosted)

#predict new values
pred = predict.gbm(object = boosted, newdata = test_dataset, n.trees =
    200, type = 'response')

# assign the label with the highest probability to the predicted value
labels_GBM = colnames(pred)[apply(pred, 1, which.max)]

#create a new data frame to compare actual labels with test result
result_GBM = data.frame(labels_GBM, test_dataset$los_days)

result_GBM$labels = as.factor(result_GBM$labels)
result_GBM$test_dataset.los_days =
    as.factor(result_GBM$test_dataset.los_days)

#compare predicted values with actual values in the test dataset
```

```
matrix_GBM = confusionMatrix(result_GBM$labels,
   result_GBM$test_dataset.los_days)
matrix_GBM


#calculate error
count = 0
L = length(result_GBM$labels)
for (i in 1:L){
  if (result_GBM$labels[i] == result_GBM$test_dataset.los_days[i]){
    count = count + 0}
  else{
    count = count + 1}}

error_rate_GBM = count/L
count
error_rate_GBM



############## Hyperparameter tuning #############

#see what hyperparameters can be tuned
modelLookup('gbm')

#specify hyperparameter grid
gb_grid = expand.grid(n.trees = c(100, 200, 500), #number of boosting
   iterations to build
                      interaction.depth = c(1,3,5,7), #max tree depth
                      shrinkage = c(0.1, 0.01, 0.001), #shrinkage
                         parameter, smaller shrinkage requires more
                         trees
                      n.minobsinnode = c(10))  #use 10 as the default
                         for most models


#use 5 fold cross validation repeated 5 times
fitControl <- trainControl(method = 'repeatedcv', number = 5, repeats
   = 5)

#train model using hyperparameter grid and 5-fold x5 cross validation
   to find optimal hyperparameters
tuned_gbm <-train(train[,c(1:48)], train_dataset$los_days,
                  method='gbm',
                  trControl=fitControl,
                  tuneGrid=gb_grid)
```

Clustering Model

```
################## CLUSTERING ##########################

library(ggplot2)
library(factoextra)
library(NbClust)
library(ggpubr)
library(ggfortify)
library(RColorBrewer)
library(dplyr)
library(readr)
library(gridExtra)

hospital_data <- read_csv("Final/hospital_data.csv")
attach(hospital_data)

#remove outliers first
hospital_data <- hospital_data[-c(3,20,32,34,1138,1690,837,301,65,29583,
    31512,34533, 19673, 45458, 16593, 2986, 29432, 42772,  6734, 15871,
    35476, 15871, 46920, 19673, 42772, 2986, 53905,28998,19673, 51941,
    19673, 46766, 1266, 11440,  1266,  19673, 5581, 45744, 14880, 38305,
    30427, 31484, 42560,19673, 20642),]

#filter data to top 5 emergency admissions first
emergency_admissions <- hospital_data%>%
  filter(admit_type == 'EMERGENCY', admit_diagnosis == c('GASTROINTESTINAL
     BLEED','CORONARY ARTERY DISEASE','PNEUMONIA','SEPSIS','CONGESTIVE
     HEART FAILURE'))

#define variables used for clustering
cluster_vars <- as.data.frame(emergency_admissions[,c(11:23)])
#don't include notes, chart events, and transfers
cluster_vars <- cluster_vars[,c(2, 3, 5, 6, 7, 9, 10, 11)]

#normalize data
normalize <-function(x){
  return((x-min(x))/max((x)-min(x)))
}
cluster_vars <- normalize(cluster_vars)


#######################CLUSTER SELECTION#################################


#elbow method suggests 3 clusters is optial
fviz_nbclust(cluster_vars, kmeans, method = "wss") +
  labs(subtitle = "Elbow method")

#silhouette method suggests 2 clusters is optimal
fviz_nbclust(cluster_vars, kmeans, method = "silhouette") +
  labs(subtitle = "Silhouette method")

#NB clust compares 30 clustering indices and takes their frequency --  3
    clusters is found to be optimal
nbclust_out <- NbClust(
  data = cluster_vars,
  distance = "euclidean",
  min.nc = 2, # minimum number of clusters
  max.nc = 5, # maximum number of clusters
  method = "kmeans"
)
# create a dataframe of the optimal number of clusters
nbclust_plot <- data.frame(clusters = nbclust_out$Best.nc[1, ])
# select only indices which select between 2 and 5 clusters
nbclust_plot <- subset(nbclust_plot, clusters >= 2 & clusters <= 5)

# create plot of optimal number of clusters
ggplot(nbclust_plot) +
  aes(x = clusters) +
  geom_histogram(bins = 30L, fill = "#0c4c8a") +
  labs(x = "Number of clusters", y = "Frequency among all indices", title
     = "Optimal number of clusters")
  theme_minimal()

####################### K MEANS #################################

#create 3 clusters using K-Means
km.3 = kmeans(cluster_vars, centers = 3, nstart = 10)

cluster_vars$cluster_labels = as.factor(km.3$cluster)
attach(cluster_vars)

####################### PLOT CLUSTERS #################################

#add variables to plot by to cluster dataframe
cluster_vars$LOS_days <- emergency_admissions$LOS_days
cluster_vars$admit_diagnosis <-
    factor(emergency_admissions$admit_diagnosis)
cluster_vars$expired <- factor(emergency_admissions$expired_hospital)
cluster_vars$admit_location <- emergency_admissions$admit_location
cluster_vars$age <- emergency_admissions$age

#rename variables for clustering
cluster_vars$LOS_days[cluster_vars$LOS_days == 0] <- '0-5'
cluster_vars$LOS_days[cluster_vars$LOS_days == 1] <- '6-10'
cluster_vars$LOS_days[cluster_vars$LOS_days == 2] <- '10+'
cluster_vars$admit_location[cluster_vars$admit_location == 'CLINIC
    REFERRAL/PREMATURE'] <- 'CLINICAL REFERRAL'
cluster_vars$admit_location[cluster_vars$admit_location == 'PHYS
    REFERRAL/NORMAL DELI'] <- 'PHYSICIAN REFERRAL'
cluster_vars$admit_location[cluster_vars$admit_location == 'TRANSFER FROM
    HOSP/EXTRAM'] <- 'TRANSFER FRM HOSP'
cluster_vars$admit_location[cluster_vars$admit_location == 'TRANSFER FROM
    SKILLED NUR'] <- 'TRANSFER FRM NURS HOME'


#relative counts per cluster, important to note that the total number of
    observations in each cluster is quite uneven, 19, 1202, and 129
p1 <- ggplot(cluster_vars, aes(cluster_labels))+geom_bar(aes(fill =
    admit_diagnosis), position = 'fill')+labs(x = 'Cluster', y ='Relative
    Frequency', fill = 'Admit Diagnosis                          ')+
    scale_fill_brewer(palette = "YlGnBu")
p2 <- ggplot(cluster_vars, aes(cluster_labels))+geom_bar(aes(fill =
    LOS_days), position = 'fill')+labs(x = 'Cluster', y ='Relative
    Frequency', fill = 'Length of Stay (Days)     ')+
    scale_fill_brewer(palette = "YlGnBu")
p3 <- ggplot(cluster_vars, aes(cluster_labels))+geom_bar(aes(fill =
    expired), position = 'fill')+labs(x = 'Cluster', y ='Relative
    Frequency', fill = 'Deceased                        ')+
    scale_fill_brewer(palette = "YlGnBu")
p4 <- ggplot(cluster_vars, aes(cluster_labels))+geom_bar(aes(fill =
    admit_location), position = 'fill')+labs(x = 'Cluster', y ='Relative
    Frequency', fill = 'Admit Location                             ')+
    scale_fill_brewer(palette = "YlGnBu")

ggarrange(p2, p1, p3, p4, ncol =2, nrow = 2)

####################### PLOT PCA CLUSTERS #################################

#perform dimension reduction, and add individual coords
res.pca <- prcomp(cluster_vars[,c(1:8)], scale = TRUE)
ind.coord <- as.data.frame(get_pca_ind(res.pca)$coord)

#add clusters, LOS, etc. to PCA
ind.coord$cluster <- factor(km.3$cluster)
ind.coord$LOS_days <- factor(cluster_vars$LOS_days)
ind.coord$admit_diagnosis <- factor(emergency_admissions$admit_diagnosis)
ind.coord$ethnicity <- factor(emergency_admissions$ethnicity)
ind.coord$insurance <- factor(emergency_admissions$insurance)

# Percentage of variance explained by dimensions
eigenvalue <- round(get_eigenvalue(res.pca), 1)
variance.percent <- eigenvalue$variance.percent
head(eigenvalue)

#Plot clusters
ggscatter(
  ind.coord, x = "Dim.1", y = "Dim.2",
  color = "cluster",   palette = c("#253494","#41B6C4" ,"#7FCDBB"),
     ellipse = TRUE, ellipse.type = "convex",
  size = 'LOS_days', legend = "right", ggtheme = theme_bw(),
  xlab = paste0("Dimension 1 (", variance.percent[1], "% )" ),
  ylab = paste0("Dimension 2 (", variance.percent[2], "% )" )
)+labs(size = 'Length of Stay (Days)')
```