PREDICTION OF HOSPITAL STAY DURATION

PROJECT REPORT

Group 53

Venkata Nithya Ala
Prabhat Chanda

ala.v@northeastern.edu
chanda.p@northeastern.edu

Percentage of Effort Contributed by Student 1: **50**

Percentage of Effort Contributed by Student 2: **50**

Signature of Student 1: Venkata Nithya Ala

Signature of Student 2: Prabhat Chanda

Submission Date: 21st April 2023

TABLE OF CONTENTS

PREDICTION OF HOSPITAL STAY DURATION

Problem Setting:

Healthcare organizations are facing growing pressure to enhance patient care and provide better facilities. It is possible to significantly improve the quality of care by leveraging data and extracting more value from it. Healthcare analytics involves analyzing health data using quantitative and qualitative methods to identify patterns and trends. Within healthcare management, various metrics are used to measure performance, however, the length of a patient's stay is crucial to providing improved care.

Problem Definition:

The ability to predict the length of stay (LOS) at the time of admission enables hospitals to optimize their treatment plans in order to decrease LOS, reduce costs and efficiently manage resources. This project aims to accurately predict the LOS of a patient during the time of admission depending on various factors like the age, severity of illness, type of injury, hospital region etc.

Data Sources:

This project is based on the research paper "Analysis of length of hospital stay using electronic health records: A statistical and data mining approach" published by the US National Center for Biotechnology Information. For the analysis, electronic health records were retrieved from a database of patients admitted to a tertiary general university hospital. To protect the sensitive information of patients and their medical history, the researchers de-identified the data and published a sample dataset with 18 variables for analysis.

(https://pubmed.ncbi.nlm.nih.gov/29652932/)

Data Description:

The dataset comprises 3,18,438 records and 18 variables with 'Stay' being the target variable that indicates the duration of a patient's stay in the hospital. The target variable 'Stay' has 11 classes (0-10, 11-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80, 81-90, 91-100, and more than 100 days). The 17 predictors/input variables that determine the target class are as follows:

case_id, Hospital_code, Hospital_type_code, City_Code_Hospital, Hospital_region_code, Available_Extra_Rooms_in_Hospital, Department, Ward_Type, Ward_Facility_Code, Bed_Grade, patientid, City_Code_Patient, Type_of_Admission, Severity_of_Illness, Visitors_with_Patient, Age, Admission_Deposit

The following **data mining tasks** have been performed to prepare the data for model building:

1. **Data Exploration:**

Exploration of the data revealed that there are 14 categorical predictors and 3 numerical predictors that determine the response variable 'Stay'. The response variable 'Stay' is an ordered categorical variable with 11 classes.

The datatype and description of each variable in the dataset are tabulated below.

Variable	Description	Datatype
case_id	Unique ID given to each case	int64
Hospital_code	Unique code for the hospital	int64
Hospital_type_code	Unique code for the type of hospital	object
City_Code_Hospital	City Code of the hospital	int64
Hospital_region_code	Region Code of the hospital	object
Available_Extra_Rooms_in_Hospital	Number of rooms available in the hospital to accommodate the patient	int64

Department	Department to which the patient is assigned	object
Ward_Type	Code for the ward type	object
Ward_Facility_Code	Code for the ward facility	object
Bed_Grade	Type of bed in the ward	float64
patientid	Unique ID given to each patient	int64
City_Code_Patient	Code of the city where the patient lives	float64
Type_of_Admission	Admission type registered by the hospital	object
Severity_of_Illness	Severity of the illness recorded at the time of admission	object
Visitors_with_Patient	Number of visitors with the patient	int64
Age	Age of the patient	object
Admission_Deposit	Amount deposited at the time of admission	int64

The number of categories in each categorical variable in the dataset are tabulated below.

Input Variable	**No.of categories**
Hospital_code	32
Hospital_type_code	7
City_Code_Hospital	11
Hospital_region_code	3
Department	5
Ward_Type	6
Ward_Facility_Code	6
Bed_Grade	4
City_Code_Patient	37
Type_of_Admission	3
Severity_of_Illness	3

Age	10

2. Data Pre-processing:

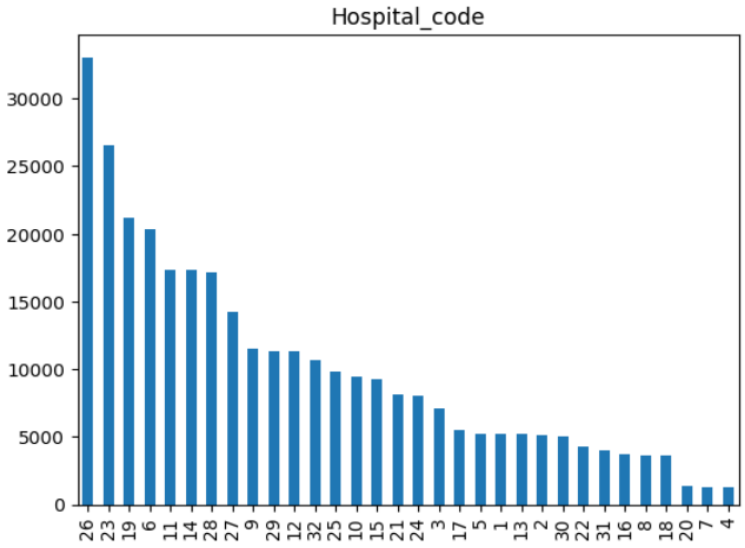
i. **Cleaning the data by removing errors:** It was observed that the category 11-20 was misrepresented as 'Nov-20' in the Age and Stay columns. So, the erroneous data has been corrected by replacing with appropriate category.

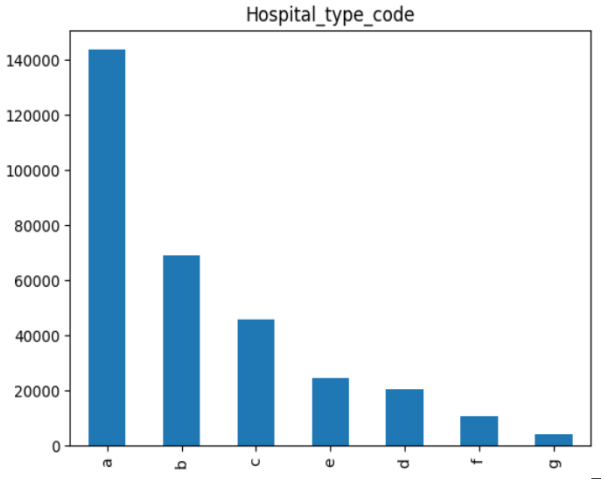
ii. **Missing values:** The columns 'Bed Grade' and 'City_Code_Patient' have 113 and 4532 missing values respectively.

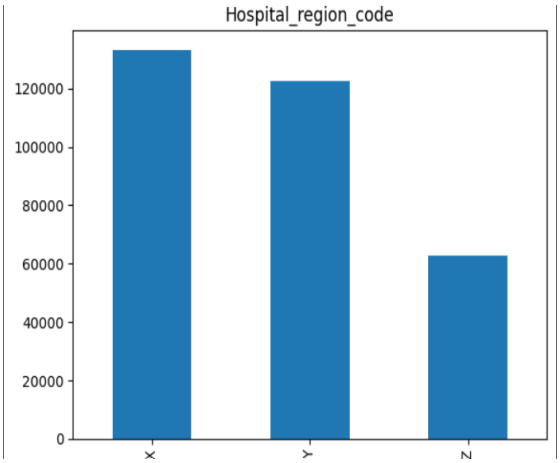
The rows where 'Bed Grade' value is missing have been dropped because there would not be significant data loss (<0.05%). The missing values in the 'City_Code_Patient' have been imputed with mode of the column (since the variable is categorical).

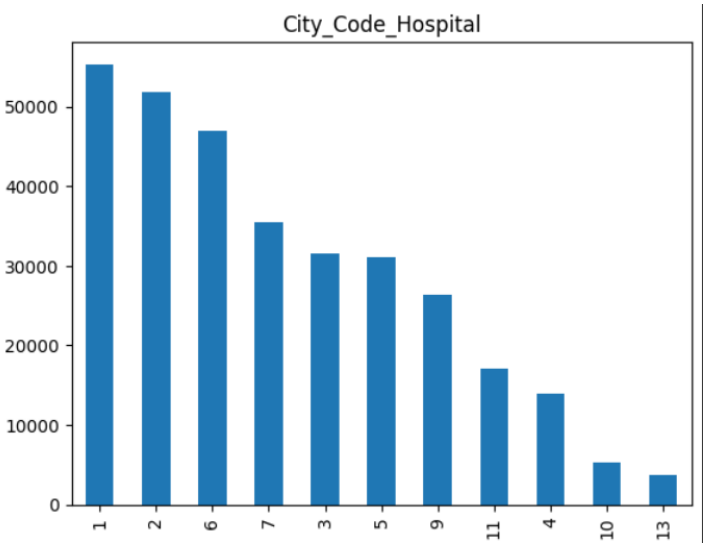
3. Data Visualization:

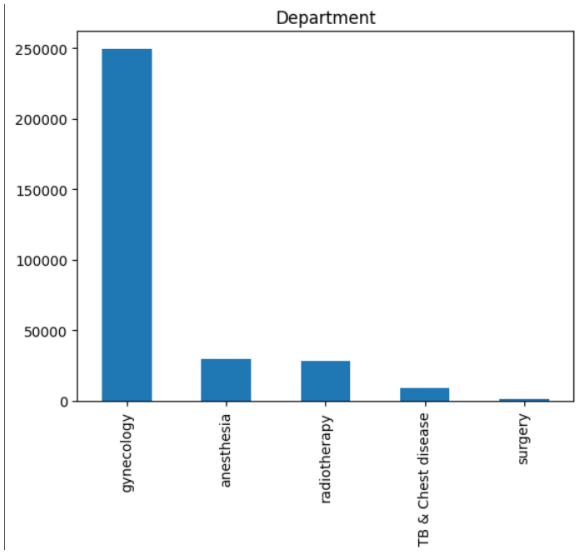
The input variables are visualized to understand the data distribution. Categorical predictors have been visualized with bar charts.

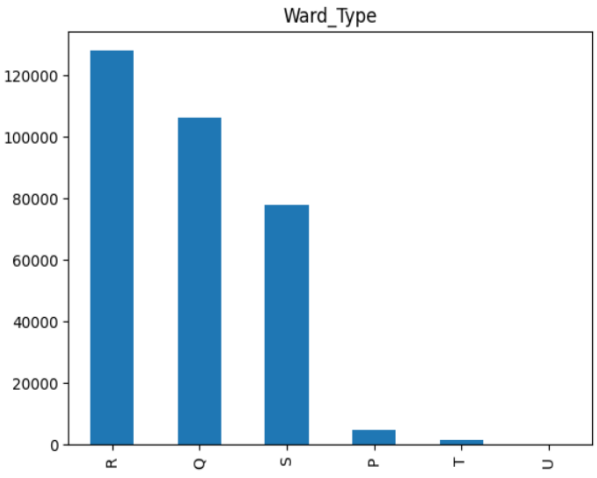


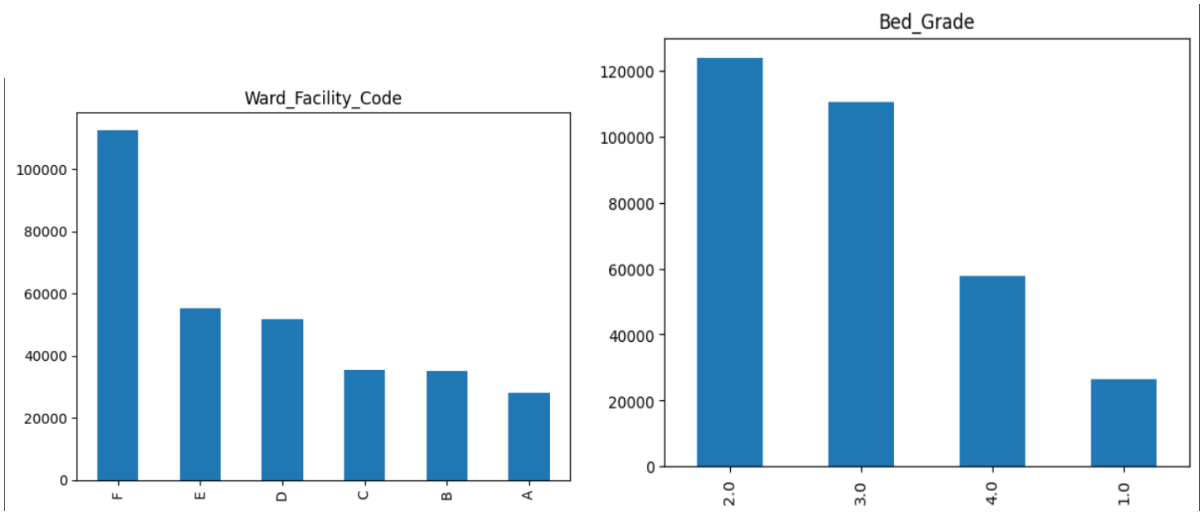

Hospital_type_code
140000
120000
100000
80000
60000
40000
20000
0
a
b
c
e
d
f
g


Hospital_region_code
120000
100000
80000
60000
40000
20000
0
X
Y
Z

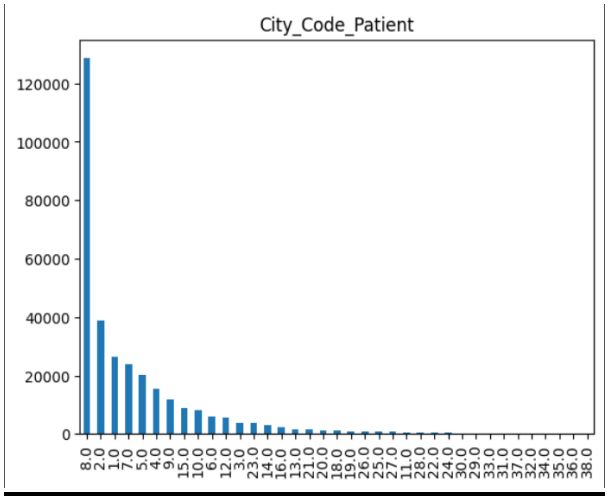

City_Code_Hospital
50000
40000
30000
20000
10000
0
1
2
6
7
3
5
9
11
4
10
13

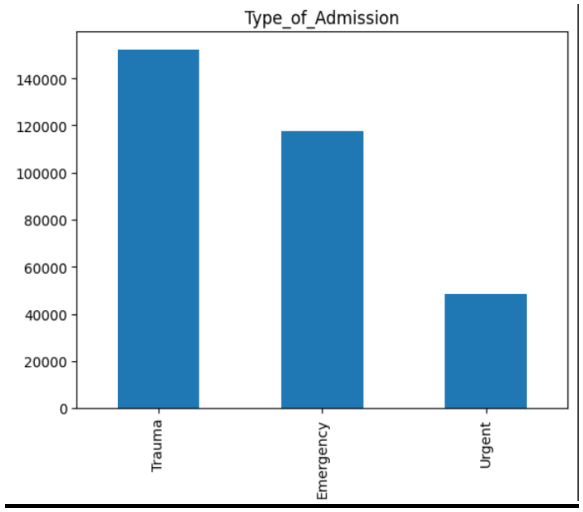

Department
250000
200000
150000
100000
50000
0
gynecology
anesthesia
radiotherapy
TB & Chest disease
surgery

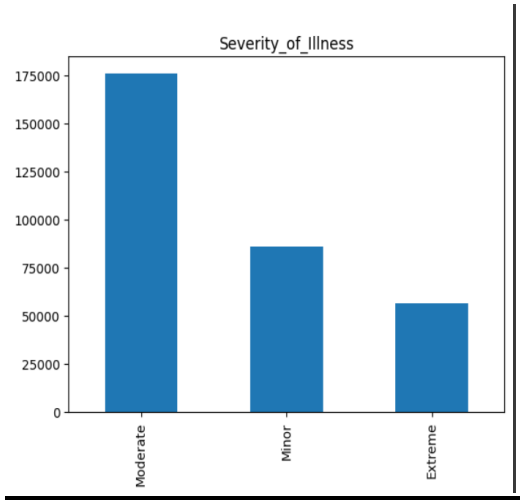

Ward_Type
120000
100000
80000
60000
40000
20000
0
R
Q
S
P
T
U

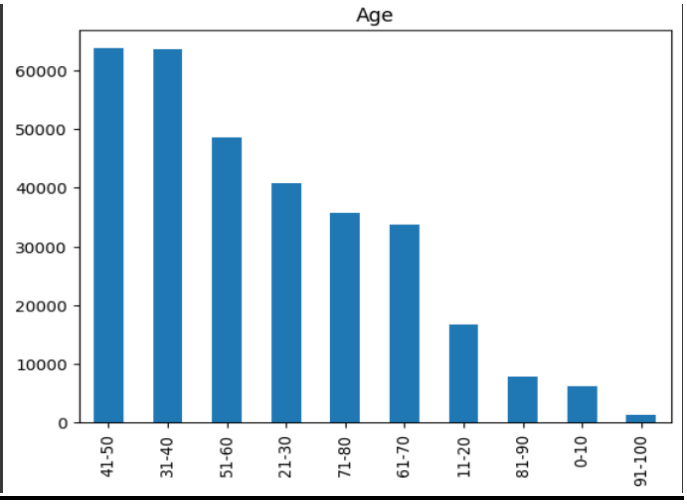

Ward_Facility_Code
100000
80000
60000
40000
20000
0
F
E
D
C
B
A

Bed_Grade
120000
100000
80000
60000
40000
20000
0
2.0
3.0
4.0
1.0

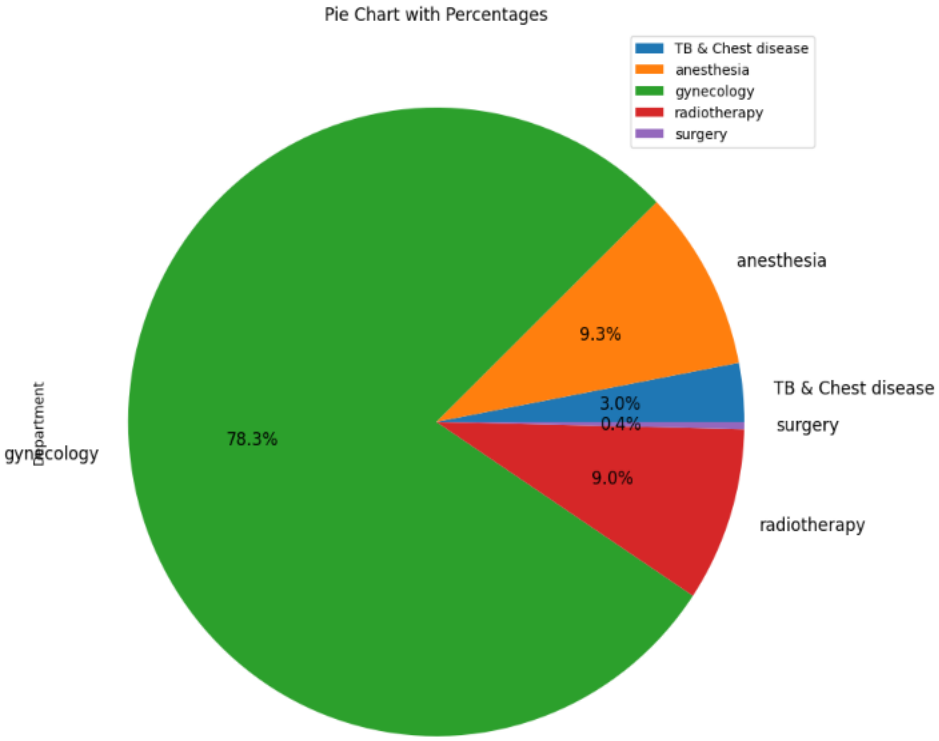

City_Code_Patient
120000
100000
80000
60000
40000
20000
0
8.0
2.0
1.0
7.0
5.0
4.0
9.0
15.0
10.0
6.0
12.0
3.0
23.0
14.0
16.0
13.0
21.0
20.0
18.0
19.0
26.0
25.0
27.0
11.0
28.0
22.0
24.0
30.0
29.0
33.0
31.0
37.0
32.0
34.0
35.0
36.0
38.0


Type_of_Admission
140000
120000
100000
80000
60000
40000
20000
0
Trauma
Emergency
Urgent

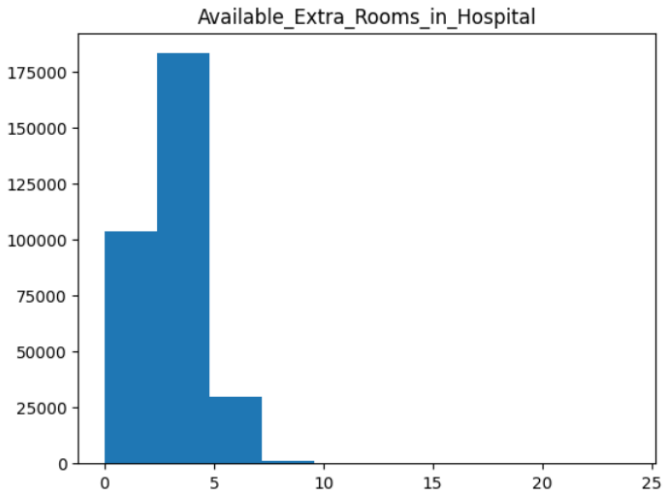

Severity_of_Illness
175000
150000
125000
100000
75000
50000
25000
0
Moderate
Minor
Extreme

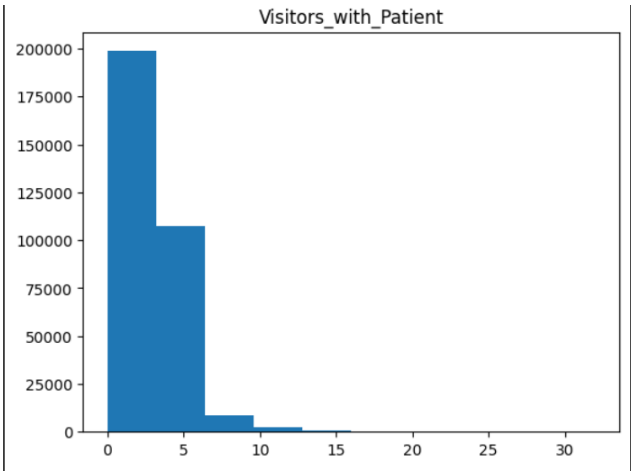

Age
60000
50000
40000
30000
20000
10000
0
41-50
31-40
51-60
21-30
71-80
61-70
11-20
81-90
0-10
91-100

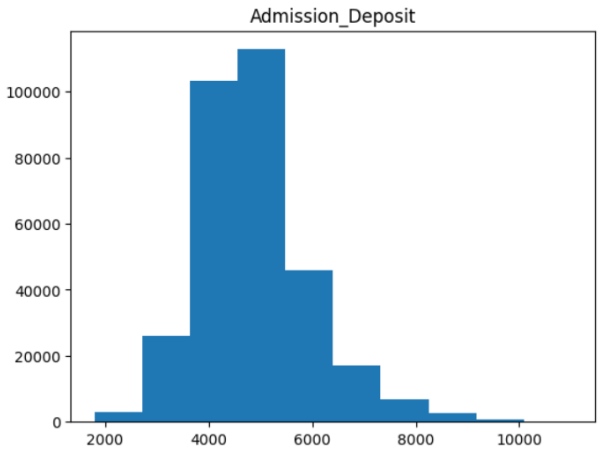
A pie chart has been used to visualize the departments in a better way.



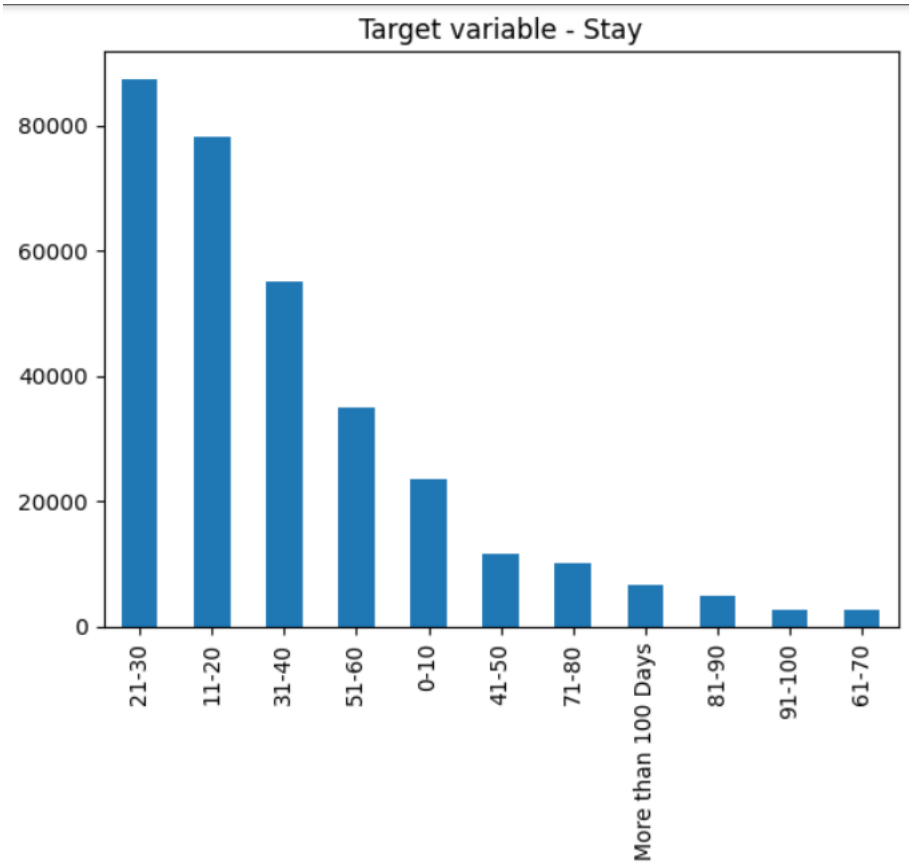
Histograms have been plotted to visualize the numerical variables.







The response variable is visualized to understand the class distribution.



4. Label encoding the categorical variables:

The following unordered categorical variables have been encoded using a label encoder to represent categories with numerical values.

'Hospital_code', 'Hospital_type_code', 'City_Code_Hospital','Hospital_region_code', 'Department', 'Ward_Type', 'Ward_Facility_Code','Bed_Grade','patientid','City_Code_Patient'

The following ordered categorical variables have been encoded manually to represent categories with numerical values and also preserve the order.

'Type of Admission', 'Severity of Illness', 'Age'

The categories in the response variable have also been encoded with numerical values as follows:

'0-10':0, '11-20':1, '21-30':2, '31-40':3, '41-50':4, '51-60':5, '61-70':6, '71-80':7, '81-90':8, '91-100':9, 'More than 100 Days':10

5. Finding the relationship among variables (Correlation and association analysis):

i. Finding the relationship of predictors with response variable:

Since the response variable is categorical and ordered, Kendall Rank Correlation Coefficient (Kendall's Tau) is used to find the association between ordered categorical predictors and response variable.

The Kendall rank correlation coefficient (Kendall's Tau) is a non-parametric measure of correlation that measures the strength of association between two variables that are measured on an ordinal scale.

```
Kendall's tau: -0.047048412212309924
P-value: 3.819181063646705e-216
Kendall's tau: 0.10931745434050788
P-value: 0.0
Kendall's tau: 0.0700812260552696
P-value: 0.0
```

The Tau and p-values for all three variables indicate that 'Type_of_Admission', 'Severity_of_Illness', 'Age' are highly significant for prediction of 'Stay'.

Spearman's Rank Correlation Coefficient (Spearman's Rho) is used to find the association between unordered categorical predictors and response variable.

Spearman's rank correlation coefficient (ρ) is a non-parametric measure of rank correlation that measures the degree of association between two variables based on their ranks.

```
Spearman's rank correlation coefficient: 0.022412521185368996
p-value: 1.1659918478058354e-36
Spearman's rank correlation coefficient: 0.06648877074054381
p-value: 1.186386233997727e-308
Spearman's rank correlation coefficient: -0.02366613428196435
p-value: 1.1177237136557954e-40
Spearman's rank correlation coefficient: 0.007349349305588868
p-value: 3.3750117563634684e-05
Spearman's rank correlation coefficient: 0.026326901628815652
p-value: 6.339106455805078e-50
Spearman's rank correlation coefficient: 0.16167351791632936
p-value: 0.0
Spearman's rank correlation coefficient: 0.007952354613051752
p-value: 7.230242236353188e-06
Spearman's rank correlation coefficient: -0.0018584423736621595
p-value: 0.2943919511271515
Spearman's rank correlation coefficient: 0.0007460716962956892
p-value: 0.6738030737545939
Spearman's rank correlation coefficient: 0.037596397610964484
p-value: 6.320195502744297e-100
```

All the p-values are extremely close to 0, indicating that all unordered categorical variables are also highly significant for prediction of 'Stay'.

ii. Finding the relationship of predictors with each other:

Spearman's Rank Correlation Coefficient (Spearman's Rho) can be used for both ordered and unordered categorical variables, so it has been used to find association of all categorical input variables.

Spearman's Rho is observed to be close to zero for all the pairs of predictors, indicating that no two predictors are correlated. Therefore, dimension reduction is not necessary for this dataset.

6. **Feature Engineering:**

The dataset consists of multiple records for each patient where patient was admitted multiple times in different hospitals of the same region and was allocated different wards for every admit. So, feature engineering is necessary.

3 new columns namely, count_of_admits_of_a_patient, count_of_admits_of_a_patient_in_ hospitalregion, count_of_wards_allocated_to_patient have been created.

- count_of_admits_of_a_patient: Group case_id and patientid to get the count of multiple admits of a patient.
- count_of_admits_of_a_patient_in_hospitalregion: Group case_id, patientid, and Hospital_region_code to get the count of multiple admits of a patient in a hospital region.
- count_of_wards_allocated_to_patient: Group case_id, patientid, and Ward_Facility_Code to get the count of wards allocated to a patient.

After feature engineering, the columns used to create the new columns have been dropped to avoid duplication of information.

7. **Normalization of numerical variables:**

After feature engineering there are 6 numerical columns in the data.

'Available_Extra_Rooms_in_Hospital', 'Visitors_with_Patient', Admission_Deposit', 'count_of _admits_of_a_patient','count_of_admits_of_a_patient_in_hospitalregion','count_of_wards_alloca ted_to_patient'

These columns have been normalized to avoid the difference in impact of variables on ‘Stay’ prediction due to difference in scales.

8. **Sampling the data for training and testing models:**

To avoid data leakage (to ensure that test data is new and unseen by the models), training data is taken from the first 200000 records in the dataset and test data is taken from the last 10000 records.

Training data: Since the class distribution is highly imbalanced, to ensure that training data is balanced, 1000 records are taken from each class for training the models.

Test data: A stratified sample (one that is representative of the population) consisting of 4000 records is taken for testing.

Data Mining Models:

The following data mining models have been considered for predicting the target variable 'Stay' and various advantages and disadvantages of each model have been explored.

- Logistic regression
- K-Nearest Neighbors (KNN)
- Decision Trees
- Random Forests
- Naive Bayes
- XGBoost

LOGISTIC REGRESSION:

The target variable is categorical with 11 classes. Logistic regression is typically used for binary classification problems where the target variable has two classes. However, extensions of logistic regression such as multinomial logistic regression or softmax regression can be used for multi-class classification problems.

In multinomial logistic regression or softmax regression, the model learns the probability of each class and assigns the observation to the class with the highest probability. It is important to ensure that the assumptions of the model are met, and the data is appropriately pre-processed and feature engineered.

Advantages:

Simple and easy to implement: It is a simple and computationally efficient algorithm that is easy to implement and interpret.

Handles both linear and non-linear relationships: Logistic regression can handle both linear and non-linear relationships between the input features and the target variable.

Easy to improve generalization: It can be easily regularized to avoid overfitting and improve generalization performance.

Disadvantages:

Logistic regression is only efficient when the number of classes is relatively small and the input features are not too complex or high-dimensional. However, it may not perform as well as more complex algorithms like random forests or neural networks in some cases, especially when the data is highly non-linear or the number of classes is very large.

K-NEAREST NEIGHBORS (KNN):

K-Nearest Neighbors (KNN) algorithm can handle both categorical and numerical predictors and hence is useful predicting the length of stay of a patient based on the features of other patients who are similar to the new patient.

To use KNN for hospital stay prediction, it is important to pre-process the data, normalise the input features, and select the appropriate distance metric for the data.

Advantages:

Non-parametric: KNN is a non-parametric algorithm that does not make any assumptions about the underlying distribution of the data. This makes it a useful algorithm when the underlying distribution of the data is unknown or cannot be assumed to be Gaussian. The algorithm works by finding the k nearest neighbours to a new patient based on the distance between their feature values. The length of stay for the new patient is then predicted based on the average length of stay of the k nearest neighbours.

Simple and easy to implement: KNN is a simple and easy-to-implement algorithm that requires minimal training time. This makes it useful in situations where time is limited, and quick predictions are required.

No training required: KNN does not require any training of a model in advance, and it can adapt to new data quickly.

Handles both continuous and categorical data: KNN can handle both continuous and categorical data, making it a versatile algorithm for hospital stay prediction, where input features can be both types of data.

Disadvantages:

Computationally expensive: KNN can be computationally expensive, especially for large datasets or datasets with many input features such as hospital admission data. This can be a limitation in situations where computational resources are limited.

Sensitivity to noise: KNN is sensitive to noisy data and outliers, which can negatively impact the accuracy of the model. This can be a limitation in hospital stay prediction data, where there may be some degree of noise or outliers in the data.

Curse of dimensionality: KNN is subject to the curse of dimensionality, which refers to the phenomenon where the distance between points in high-dimensional space becomes increasingly similar, making it difficult to distinguish between them. This can be a limitation in hospital stay prediction data, where there may be many input features to consider.

Bias towards majority class: KNN has a bias towards the majority class in the training data, which can result in inaccurate predictions for minority classes. This can be a limitation in hospital stay prediction data, where there may be imbalanced classes.

Choice of K value: The choice of K value in KNN can impact the accuracy of the model. If K is too small, the model may be too sensitive to noise, while if K is too large, the model may not be able to capture local patterns in the data.

DECISION TREES:

Decision tree classifiers are a popular algorithm used in hospital stay prediction data. A decision tree classifier is a predictive model that maps observations about an individual to conclusions about their hospital stay.

Advantages:

Easy to understand and interpret: Decision trees are easy to understand and interpret, making them useful in medical decision-making. The algorithm provides a clear visualisation of the decision-making process, which can be useful in understanding how the algorithm arrives at its predictions.

Feature importance: Decision trees provide a measure of feature importance, which can be useful for identifying the most important predictors of hospital stay. This can be particularly useful in medical decision-making, where understanding the predictors of a hospital stay can be important.

Handles both categorical and numerical data: Decision trees can handle both categorical and numerical data, making them versatile for a wide range of hospital stay prediction datasets.

Handles missing values: Decision trees can handle missing data, which is common in hospital stay prediction data. The algorithm can use surrogate variables to impute missing data.

Robust to noise: Decision trees are robust to noisy data and can handle outliers, making them useful in hospital stay prediction data.

Disadvantages:

Overfitting: Decision trees are prone to overfitting, especially when the tree is too complex or the dataset is small. Overfitting can lead to poor generalisation performance and inaccurate predictions.

Unstable: Decision trees are unstable, meaning that small changes in the data can result in large changes in the tree structure. This can be a limitation in hospital stay prediction data, where the dataset may be noisy or small.

Bias towards features with many levels: Decision trees can be biased towards features with many levels, as they can lead to more splits in the tree. This can be a limitation in hospital stay prediction data, where there are features with many levels.

Lack of interpretability: While decision trees are easy to interpret, they can become complex as the tree grows. This can make it difficult to interpret the tree, especially when there are many levels.

RANDOM FORESTS:

Random Forest is a machine learning algorithm that is commonly used in predictive analysis. It is a type of ensemble learning algorithm that combines multiple decision trees to make a prediction.

Advantages:

Accurate: Random Forest can be highly accurate, even for complex datasets with many input features. It can also handle both categorical and continuous variables, making it well-suited for hospital stay prediction data that may include a variety of data types.

Non-parametric: Random Forest is a non-parametric algorithm, meaning that it does not require assumptions about the distribution of the data. This makes it well-suited for hospital stay prediction data, where the distribution of the data may be unknown or complex.

Robust to overfitting: Random Forest is robust to overfitting, which can be a limitation of other machine learning algorithms. This is because Random Forest builds multiple decision trees and averages their predictions, which can reduce the impact of individual decision trees that may have overfit the data.

Interpretable: Random Forest can provide insights into the importance of different input features in the prediction, making it easier to understand and interpret the model.

Outlier detection: Random Forest can be used to detect outliers in the data, which can be useful in hospital stay prediction data where outliers may represent abnormal cases.

Disadvantages:

Overfitting: Random Forest is prone to overfitting, especially when there are a large number of input features or when the model is too complex. This can lead to poor generalisation performance and inaccurate predictions.

Computationally expensive: Random Forest can be computationally expensive, especially for large datasets. This can be a limitation in situations where computational resources are limited.

Lack of interpretability: Random Forest is a black box model, which means that it is difficult to interpret how the model arrives at its predictions. This can be a limitation in hospital stay prediction data, where it may be important to understand the predictors of a hospital stay.

Imbalanced data: Random Forest can be biased towards the majority class in the training data, which can result in inaccurate predictions for minority classes. This can be a limitation in hospital stay prediction data, where there may be imbalanced classes.

Tuning hyperparameters: Random Forest has several hyperparameters that need to be tuned to achieve optimal performance. This can be a time-consuming and iterative process.

NAIVE BAYES:

Naïve Bayes is a classification technique that works on the principle of Bayes theorem with an assumption of independence among the variables. Here the goal is to predict Length of Stay i.e., "Stay" column (Target Variable) and it is classified into 11 levels. We must find the probability of each patient's length of stay using feature variables, which contain the patient's condition and hospital-level information. These feature variables are ordinal and naïve Bayes is a perfect multilevel classifier.

In Bayes theorem, given a Hypothesis H and Evidence E, it states that the relation between the probability of Hypothesis P(H) before getting Evidence and probability of hypothesis after getting Evidence P(H|E):

P (H | E) = [*P* (*E* | *H*) / *P*(*E*)] P(H)

When we apply Bayes Theorem to our data it represents as follows.

• P(H) is the prior probability of a patient's length of stay (LOS).

• P(E) is the probability of a feature variable.

• P(E|H) is the probability of a patient's LOS given that the features are true.

• P(H|E) is the probability of the features given that patient's LOS is true.

Advantages:

Simple and easy to implement: Naive Bayes is a simple and easy-to-implement algorithm that requires minimal training time. This makes it useful in situations where time is limited, and quick predictions are required.

Fast and efficient: Naive Bayes is a fast and efficient algorithm that can handle large datasets with many input features. This makes it useful in situations where there are many input features to consider, and other algorithms may be computationally expensive or slow.

Handles both continuous and categorical data: Naive Bayes can handle both continuous and categorical data, making it a versatile algorithm for hospital stay prediction, where input features can be both types of data.

Interpretable: The Naive Bayes algorithm is interpretable, meaning that the predictions made by the algorithm can be explained based on the contribution of each input feature to the final prediction. This makes it useful in situations where the explanations behind the predictions are important, such as in medical decision-making.

Disadvantages:

Naive assumption: Naive Bayes assumes that all input features are independent of each other, which may not be true in real-world scenarios. This can lead to inaccurate predictions if there are correlations between the input features.

Sensitivity to irrelevant features: Naive Bayes is sensitive to irrelevant input features. This means that even if some input features have no correlation with the length of stay, they can still influence the final prediction.

Limited expressive power: Naive Bayes has limited expressive power compared to other machine learning algorithms such as neural networks or decision trees. This means that it may not capture complex relationships between the input features and the length of stay.

Limited ability to handle imbalanced data: Naive Bayes can be sensitive to imbalanced datasets, where one class has significantly fewer observations than the others. This can lead to biased predictions if the dataset is not balanced.

XGBOOST:

Extreme Gradient Boosting is a sequential technique that works on the principle of an ensemble. At any instant T, the model outcomes are weighed based on the outcomes of the previous instant (T -1). It combines the set of weak learners and improves prediction accuracy. Tree ensemble is a set of classification and regression trees. Trees are grown one after another, and they try to reduce the misclassification rate. The final prediction score of the model is calculated by summing up each individual score.

Before feeding train data to the XGB Classifier model, booster parameters must be tuned. Tuning the model can prevent overfitting and can yield higher accuracy.

Advantages:

High accuracy: XGBoost is known for its high accuracy. It uses an ensemble of weak decision trees to make accurate predictions and can handle both continuous and categorical data.

Regularisation: XGBoost has a regularisation feature that helps to reduce overfitting and improve the generalisation of the model. This is important in hospital stay prediction, where accurate predictions on new data are crucial.

Feature importance: XGBoost provides a measure of feature importance, which helps to identify which input features are most important in predicting hospital stay. This can aid in medical decision-making and identifying areas for further research.

Scalability: XGBoost can handle large datasets with many input features, making it useful in situations where there is a large amount of data available.

Disadvantages:

Complexity: XGBoost is a complex algorithm that requires careful tuning of hyperparameters to achieve optimal performance. This can be time-consuming and requires expertise in machine learning.

Computationally intensive: XGBoost can be computationally intensive, especially for large datasets or datasets with many input features. This can be a limitation in situations where computational resources are limited.

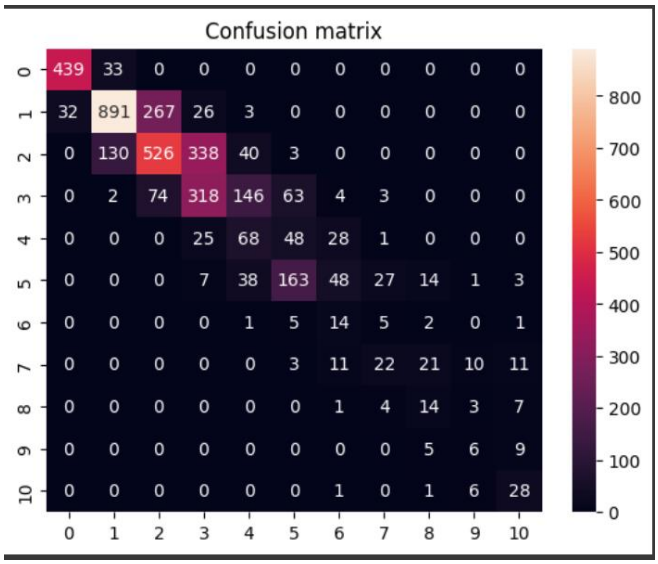
Black box model: XGBoost is a black box model, meaning that it can be difficult to interpret how the model is making predictions. This can be a limitation in medical decision-making, where understanding how the model arrived at a particular prediction is important.

Overfitting: While XGBoost has a built-in regularisation feature to prevent overfitting, it can still occur if the model is not properly tuned. Overfitting can lead to inaccurate predictions on new data.

Performance Evaluation and Insights:

LOGISTIC REGRESSION MODEL:

A logistic regression model that can handle multi-class classification has been built by specifying the model parameters as: LogisticRegression(multi_class='multinomial', solver='lbfgs'). The confusion matrix and scores for each class are shown below.



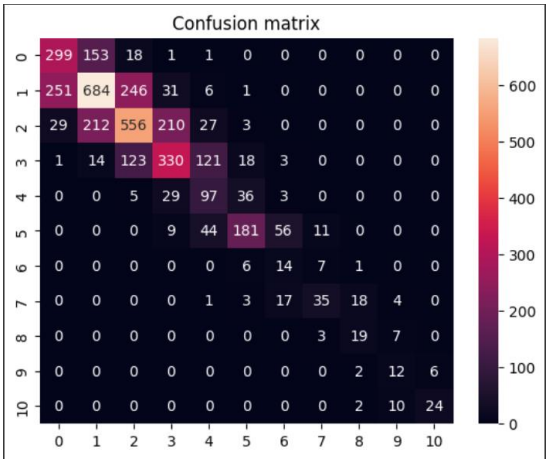
	precision	recall	f1-score	support
0	0.93	0.93	0.93	472
1	0.84	0.73	0.78	1219
2	0.61	0.51	0.55	1037
3	0.45	0.52	0.48	610
4	0.23	0.40	0.29	170
5	0.57	0.54	0.56	301
6	0.13	0.50	0.21	28
7	0.35	0.28	0.31	78
8	0.25	0.48	0.33	29
9	0.23	0.30	0.26	20
10	0.47	0.78	0.59	36
accuracy			0.62	4000
macro avg	0.46	0.54	0.48	4000
weighted avg	0.66	0.62	0.64	4000

INSIGHTS:

The multinomial logistic regression model exhibits a moderate performance, so other models have to be explored for better performance.

K-NEAREST NEIGHBORS (KNN) CLASSIFIER:

A best k value is chosen by iterating from 1 to 10 and finding the k with highest accuracy. The model is trained with this best k and the confusion matrix and scores are as follows.



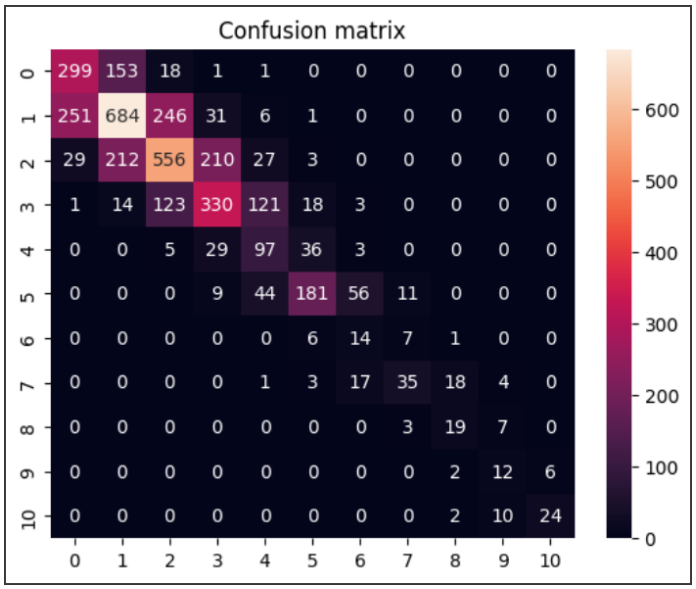

```
Accuracy with 1 neighbors: 0.56275
Accuracy with 2 neighbors: 0.54925
Accuracy with 3 neighbors: 0.507
Accuracy with 4 neighbors: 0.55375
Accuracy with 5 neighbors: 0.55
Accuracy with 6 neighbors: 0.54275
Accuracy with 7 neighbors: 0.5355
Accuracy with 8 neighbors: 0.5345
Accuracy with 9 neighbors: 0.523
Accuracy with 10 neighbors: 0.52425

Best accuracy of 0.56 was achieved with 1 neighbors.
```

```
Best accuracy of 0.56 was achieved with 1 neighbors.
              precision    recall  f1-score   support

           0       0.52      0.63      0.57       472
           1       0.64      0.56      0.60      1219
           2       0.59      0.54      0.56      1037
           3       0.54      0.54      0.54       610
           4       0.33      0.57      0.42       170
           5       0.73      0.60      0.66       301
           6       0.15      0.50      0.23        28
           7       0.62      0.45      0.52        78
           8       0.45      0.66      0.54        29
           9       0.36      0.60      0.45        20
          10       0.80      0.67      0.73        36

    accuracy                           0.56      4000
   macro avg       0.52      0.57      0.53      4000
weighted avg       0.59      0.56      0.57      4000
```

INSIGHTS: The KNN model performs worse than the logistic regression model. For a dataset where there are a large number of predictors, KNN should not be used. It is observed that for this data, best k value is small and hence the model is highly sensitive to noise leading to low accuracy.

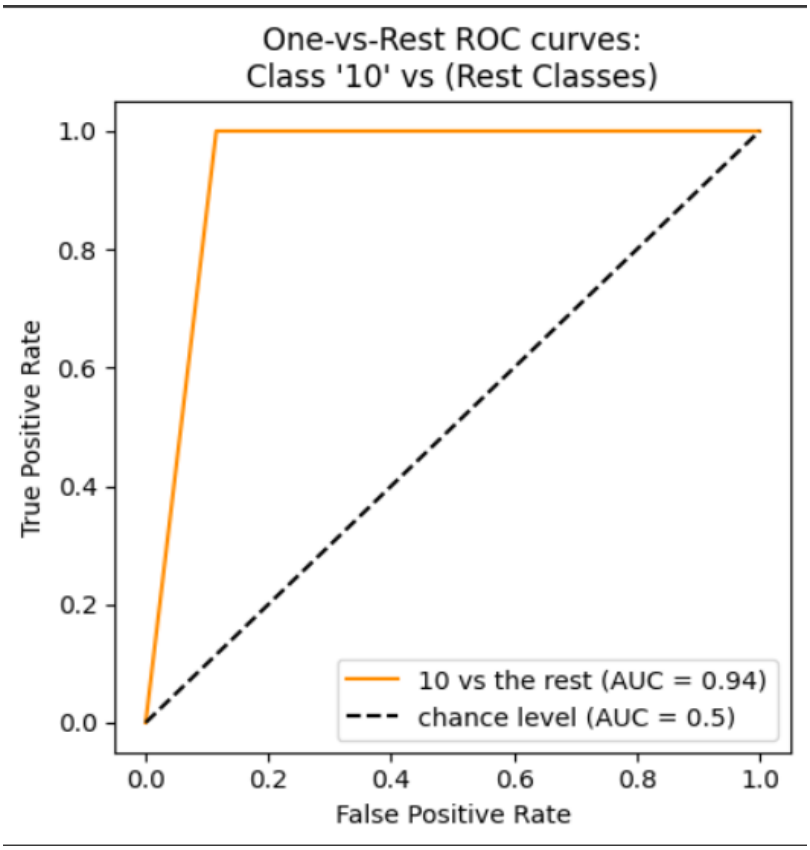
DECISION TREE CLASSIFIER:

A decision tree with maximum depth of 5 is built and the confusion matrix and scores are as follows.



	precision	recall	f1-score	support
0	1.00	1.00	1.00	472
1	1.00	1.00	1.00	1219
2	1.00	1.00	1.00	1037
3	1.00	1.00	1.00	610
4	1.00	1.00	1.00	170
5	0.61	1.00	0.76	301
6	0.00	0.00	0.00	28
7	0.00	0.00	0.00	78
8	0.00	0.00	0.00	29
9	0.00	0.00	0.00	20
10	0.00	0.00	0.00	36
accuracy			0.95	4000
macro avg	0.51	0.55	0.52	4000
weighted avg	0.92	0.95	0.93	4000

The area under the ROC is 0.94 which is another indicator of high performance and near-perfect classification of decision trees. (Class 10, i.e., more than 100 days is taken as class of interest to plot the ROC curve).

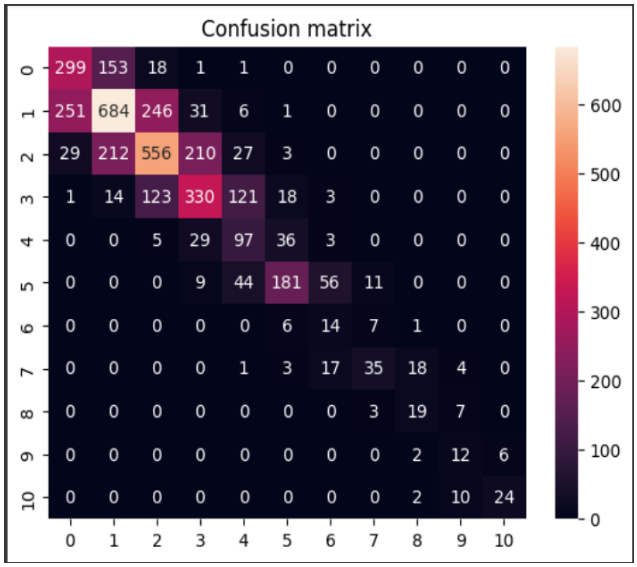


INSIGHTS:

A decision tree classifier exhibits very high performance. For multi-class classification problems, decision trees are one of the best choices. However, ensemble techniques have to be explored for achieving even better performance.

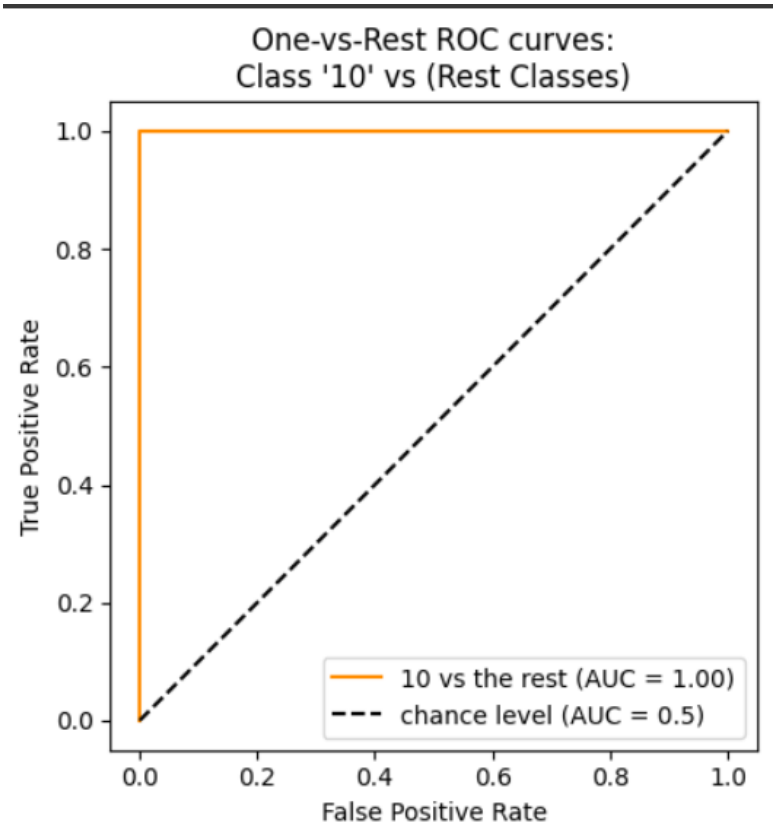
RANDOM FOREST CLASSIFIER:

Random forest is an ensemble learning method that combines multiple decision tree classifiers to improve the performance. A random forest is built with n_estimators=100 and the confusion matrix and scores as follows:



	precision	recall	f1-score	support
0	1.00	1.00	1.00	472
1	1.00	1.00	1.00	1219
2	1.00	0.98	0.99	1037
3	0.97	1.00	0.98	610
4	0.98	0.97	0.98	170
5	0.97	0.99	0.98	301
6	0.88	1.00	0.93	28
7	0.99	0.96	0.97	78
8	1.00	1.00	1.00	29
9	1.00	1.00	1.00	20
10	1.00	1.00	1.00	36
accuracy			0.99	4000
macro avg	0.98	0.99	0.98	4000
weighted avg	0.99	0.99	0.99	4000

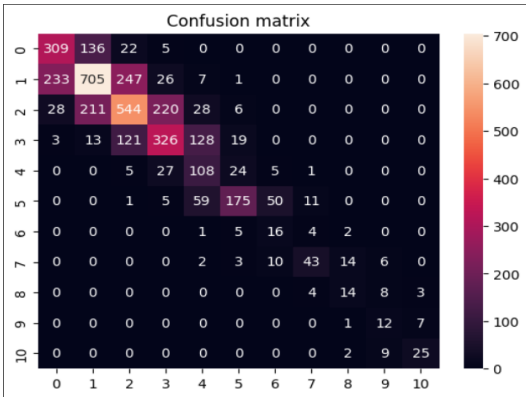
The area under the ROC is another indicator of high performance and near-perfect classification of random forest classifier. (Class 10, i.e., more than 100 days is taken as class of interest to plot the ROC curve).



INSIGHTS: Random forest is an ensembling technique that combines several decision trees to improve the model performance. Since the decision trees already perform well, combining several decision trees resulted in extremely high performance.

NAIVE BAYES CLASSIFIER:

The naïve assumption that the predictors are independent of each other leads to the following scores. The confusion matrix and scores are as follows,



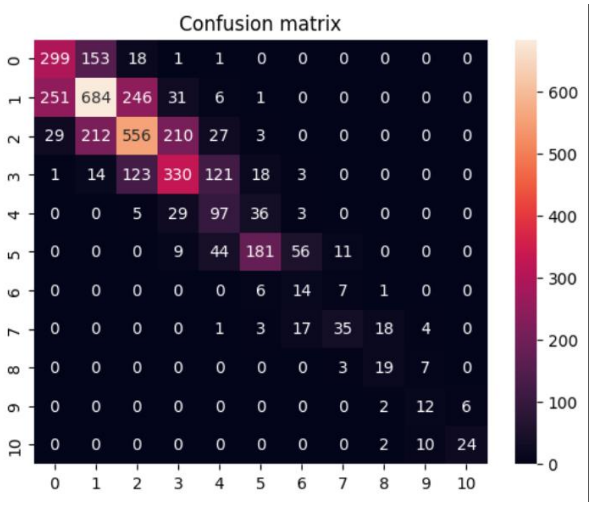
	precision	recall	f1-score	support
0	1.00	1.00	1.00	472
1	1.00	1.00	1.00	1219
2	1.00	1.00	1.00	1037
3	1.00	1.00	1.00	610
4	1.00	1.00	1.00	170
5	1.00	1.00	1.00	301
6	1.00	1.00	1.00	28
7	1.00	1.00	1.00	78
8	1.00	1.00	1.00	29
9	1.00	1.00	1.00	20
10	1.00	1.00	1.00	36
accuracy			1.00	4000
macro avg	1.00	1.00	1.00	4000
weighted avg	1.00	1.00	1.00	4000

INSIGHTS:

It is observed that naive bayes algorithm is overfitting the data. Although naive bayes is a simple and efficient model, in cases where the dataset is imbalanced, it fails to function properly. Here, the naive bayes model is memorizing the training data instead of detecting the underlying patterns. So, even though the overall accuracy is high, the model should not be used.

XGBOOST CLASSIFIER:

The XGBoost classifier is built with a maximum depth of 2 and 100 estimators and the confusion matrix and scores are as follows:

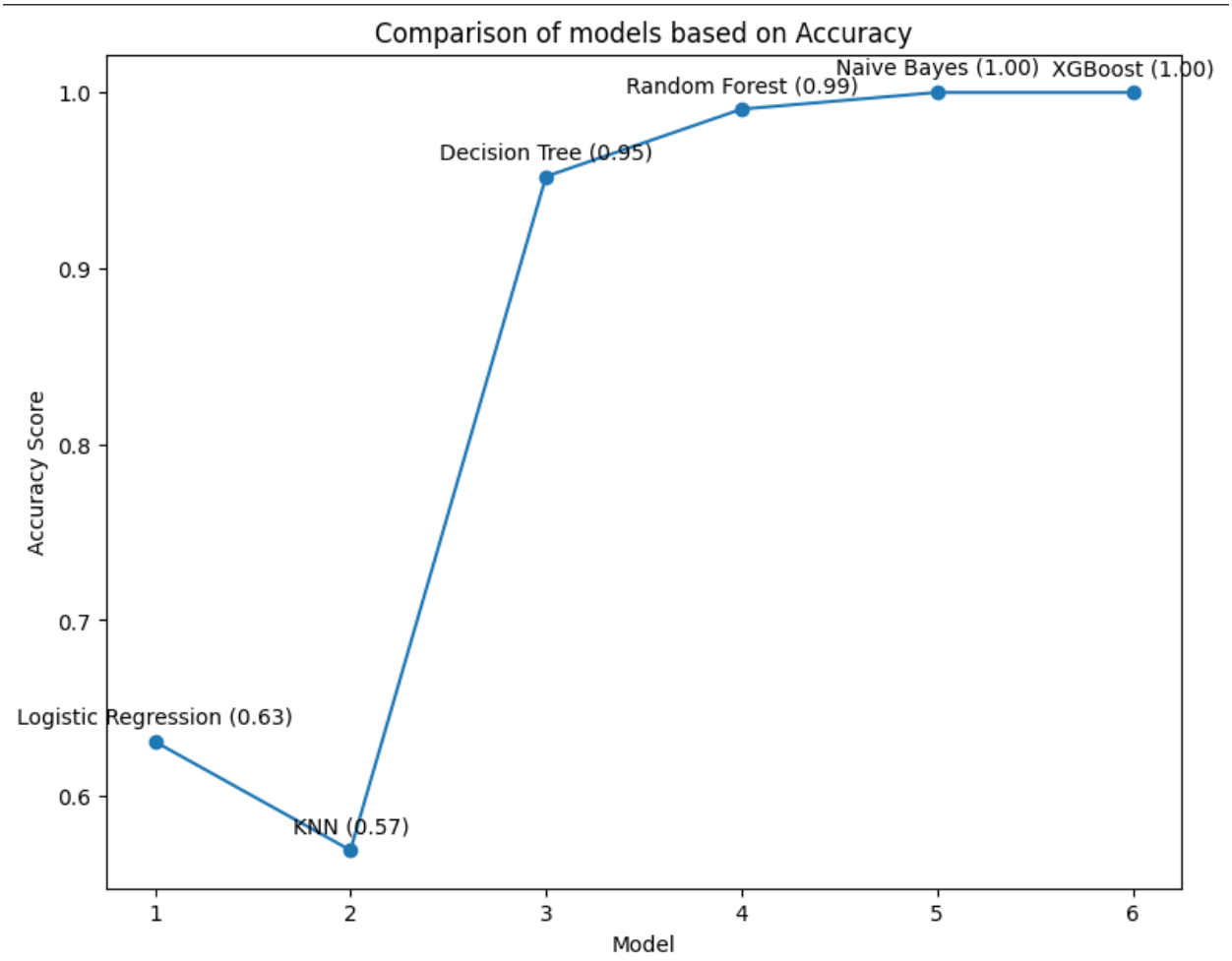


	precision	recall	f1-score	support
0	1.00	1.00	1.00	472
1	1.00	1.00	1.00	1219
2	1.00	1.00	1.00	1037
3	1.00	1.00	1.00	610
4	1.00	1.00	1.00	170
5	1.00	1.00	1.00	301
6	1.00	1.00	1.00	28
7	1.00	1.00	1.00	78
8	1.00	1.00	1.00	29
9	1.00	1.00	1.00	20
10	1.00	1.00	1.00	36
accuracy			1.00	4000
macro avg	1.00	1.00	1.00	4000
weighted avg	1.00	1.00	1.00	4000

INSIGHTS: Complex models often lead to overfitting. Although XGBoost models exhibit good performance for multi-class classification problems, it is too complex and not useful for this data because it leads to 100% accuracy indicating overfitting.

Project Results:

- Decision tree yields 95% accuracy and a F1 score of 93%. Random forest exhibits the highest accuracy of 99%, high precision (98%) and high F-1 score (99%).
- While a decision tree can be used for the prediction of hospital stay duration, a random forest classifier yields best results.



A random forest classifier should be used for the accurate prediction of hospital stay duration.

Impact of the project outcomes:

The implementation of this highly accurate predictive model can have a significant impact on the healthcare industry by enabling hospitals to better manage patient flow and allocate resources efficiently. Using this model, hospitals can plan and prepare for patient admissions and discharges, thereby improving patient experience. Additionally, allocating the resources efficiently helps to manage emergencies effectively. Overall, the use of this predictive model has the potential to improve patient outcomes, reduce healthcare costs, and enhance the overall quality of care provided by hospitals.

Conclusion and Future Scope:

A random forest classifier accurately predicts the duration of hospital stay of patients at the time of admission.

Hospitals can implement this model to manage the resources efficiently, handle emergencies effectively, reduce costs and improve patient experience.

Furthermore, the model can be improvised to predict a definite number for hospital stay instead of a category representing a range of days.

References:

Hyunyoung Baek, Minsu Cho, Seok Kim , Hee Hwang, Minseok Song, Sooyoung Yoo, *Analysis of length of hospital stay using electronic health records: A statistical and data mining approach.* https://pubmed.ncbi.nlm.nih.gov/29652932/