**PREDICTIVE MODELLING AND ANALYSIS OF HOSPITAL INPATIENT STAYS AND COSTS USING SPARCS DATASET**

**Abstract:** This study intends to analyse the SPARCS dataset, a comprehensive statewide planning and research system that contains records of hospital inpatient discharges in New York State, with the purpose of understanding the determinants of hospital length of stay and associated costs. Using exploratory data analysis and predictive modelling, we find key patterns, trends, and connections that can guide healthcare decision-making and resource allocation. The analysis focuses on two main objectives: calculating the total costs associated with hospital inpatient stays and predicting length of stay of patients along with their diagnoses and other relevant factors. Linear regression, random forest and K-Nearest Neighbours (KNN) models are utilized to create predictive models. The findings show that a linear regression model does better than a random forest model in predicting total costs, while the KNN model is the top performer at predicting length of stay. The results give healthcare managers helpful information concerning resource allocation, patient care, and specifically sustainable healthcare system development.

**Table of Contents**

1. **Introduction**

1.1. Background on SPARCS

1.2. Objective of the Analysis

1.3. Problem Statement and Research Questions

1.4. Impact and Significance

1. **Data Understanding and Preparation**

2.1. Overview of the Dataset

2.2. Handling Missing Values

2.3. Data Type Conversion

1. **Exploratory Data Analysis (EDA)**

3.1. Descriptive Statistics

3.2. Analysis of Categorical Variables

3.3. Correlation Analysis

3.4. Relationship between Total Cost and CCS Diagnosis Description across Facilities

1. **Prediction Modelling 1: Predicting Total Costs**

4.1. Problem Statement and Analysis

4.2. Linear Regression Model

4.3. Random Forest Model

4.4. Model Performance Comparison

1. **Prediction Modelling 2: Predicting Length of Stay**

5.1. Problem Statement and Analysis

5.2. Linear Regression Model

5.3. Random Forest Model

5.4. K-Nearest Neighbours (KNN) Model

5.5. Model Performance Comparison

1. **Conclusion and Recommendations**

6.1. Summary of Key Findings

6.2. Recommendations for Healthcare Stakeholders

1. **Introduction**
   1. **Background on SPARCS:**

The Statewide Planning and Research Cooperative System (SPARCS), which has been founded in the past in 1979, consists of a centralized data tracking system that records hospital admission and discharge data of New York residents. SPARCS was established jointly between industries in healthcare and government agencies so that the same unified system exists for integrating and processing hospital-related data. The data set for this analysis, 'Hospital Inpatient Discharges (SPARCS De-Identified)', is one fragment of the complete SPARCS data that carries the detailed discharge-level information for the patients' characteristics, diagnoses, interventions, services, and charges.

* 1. **Objective of the Analysis:**

The main goal of this analysis lies in understanding the relationship of those factors that impact hospital inpatient discharges and the associated costs. Through studying covariation of patient demographics, diagnoses, procedures, and hospitals characteristics, we would like to discover the relationships that will lead to the patterns and trends, which can be used in administration of healthcare resources and decision-making. The analysis will focus on two main areas: forecasting the total costs of admission for inpatient stays in the hospital and predicting patients' length of stay based on their diagnosis and other factors of concern.

1.3 **Problem Statement and Research Questions:**

The healthcare sector faces considerable problems in trying to balance expenses and optimize resource use, on the one hand, and ensure quality care for patients, on the other side. Realizing which elements lead to the variation in hospital inpatient stays and their associated costs is a crucial thing for the overcoming these challenges.

* Can we forecast the overall costs of a hospital inpatient stay by utilizing patient demographics, admission type, length of stay, diagnosis, procedures performed, severity of illness, and hospital characteristics?
* Are the factors such as diagnosis, the severity of illness and the procedures supposed to be used able to determine the length of the patient’s stay in a hospital?

Through answering these queries, we aim to create a report which can be used by healthcare providers and policymakers as the basis to direct decisions that will optimise cost-effectiveness, resource allocation, and patient care.

**1.4 Impact and Significance:**

This analysis will produce knowledge and models that will fundamentally influence the activities of the healthcare practitioners. Hospital administrators are able to make better decisions on resources allocation, operations management and financial planning based on the research findings. Healthcare providers can use the insights to improve diagnosis by identifying factors contributing to longer hospital stays and high costs. Policymakers can use those results and base their healthcare policies and initiatives on heightening health system efficiency and effectiveness. Researchers may be able to build upon the findings and hence investigate further the causes of hospital inpatient stays and come up with solutions that will address the problems in the healthcare industry.

SPARCS database represents a unique resource that offers a great deal of valuable data for investigating of hospital inpatient stays in New York. Through the comprehensive analysis of this particular dataset, our goal is to expand the knowledge base in healthcare analytics and extract valuable insights that would guide the future improvements in healthcare systems.

1. **Data Understanding and Preparation**

**2.1 Overview of the Dataset:**

The "Hospital Inpatient Discharges (SPARCS De-Identified)" dataset has 2,345,070 rows and 34 columns contained in it. Each row in this csv file represents each hospital inpatient discharge which has the patient’s demographics, diagnosis codes, procedure codes, hospital information and financial aspects.

The columns in the dataset are as follows: ['Health Service Area', 'Hospital County', 'Operating Certificate Number', 'Facility Id', 'Facility Name', 'Age Group', 'Zip Code - 3 digits', 'Gender', 'Race', 'Ethnicity', 'Length of Stay', 'Type of Admission', 'Patient Disposition', 'Discharge Year', 'CCS Diagnosis Code', 'CCS Diagnosis Description', 'CCS Procedure Code', 'CCS Procedure Description', 'APR DRG Code', 'APR DRG Description', 'APR MDC Code', 'APR MDC Description', 'APR Severity of Illness Code', 'APR Severity of Illness Description', 'APR Risk of Mortality', 'APR Medical Surgical Description', 'Payment Typology 1', 'Payment Typology 2', 'Payment Typology 3', 'Birth Weight', 'Abortion Edit Indicator', 'Emergency Department Indicator', 'Total Charges', 'Total Costs']

**2.2 Handling Missing Values:**

The initial missing value analysis for the dataset showed that many of the columns had missing values. Percentage of the missing values for each column was calculated and it was found out that some columns had a substantial number of missing values.

The columns with missing values and their respective counts and percentages are as follows:

* 'Health Service Area': 5,464 missing values (0.23%)
* 'Hospital County': 5,464 missing values (0.23%)
* 'Operating Certificate Number': 5,464 missing values (0.23%)
* 'Facility Id': 5,464 missing values (0.23%)
* 'Zip Code - 3 digits': 35,816 missing values (1.53%)
* 'APR Severity of Illness Description': 111 missing values (0.005%)
* 'APR Risk of Mortality': 111 missing values (0.005%)
* 'Payment Typology 2': 762,285 missing values (32.48%)
* 'Payment Typology 3': 1,645,513 missing values (70.11%)

A screenshot of a computer

Description automatically generated

To handle the missing values, different imputation strategies were employed based on the percentage of missing values in each column:

* For those columns that have a low percentage of missing values ('Health Service Area', 'Hospital County', 'Operating Certificate Number', 'Facility Id', 'APR Severity of Illness Description', 'APR Risk of Mortality'), missing values were imputed using the mode (most frequent value) of each respective column.
* For the 'Zip Code - 3 digits' column, missing values were imputed by the mode of the column.
* Regarding 'Payment Type 2' and 'Payment Type 3', missing values were replaced with the category 'Not Available'.

After imputing the missing values, the dataset was cross-checked to make sure that no missing values are left.

**2.3 Data Type Conversion:**

In order to guarantee best fitness and compatibility with different analyzation data methods and model techniques, the data types of some columns were changed.

'Operating Certificate Number' and 'Facility\_id' were converted from float64 to int32 and int16.

Several columns with the data type 'category' were changed to 'categories'. These columns are 'Health Service Area' 'County of the Hospital', 'Gender', 'Race', 'Ethnicity', 'Type of Admission', 'Patient Disposition', 'APR Severity of Illness Description', 'APR Risk of Mortality', 'APR Medical Surgical Description', 'Payment Typology 1', 'Payment Typology 2' 'Payment Typology 3', 'Emergency Department Indicator', and 'Abortion Edit Indicator'.

The 'Length of Stay' column which was turned into float64 from object, by pd.to\_numeric with the 'coerce' option to manage any non-numeric values. Following the data type conversion, the data types of the dataset were re-verified to ensure the equality.

Hence, data preparation and understanding steps secure the database is clear, complete and in the right format to go through the next level steps of analysis or modelling. Applying imputation techniques for missing values and data type conversions, the data set is now turning out to be quite ideal for exploratory data analysis and predictive modelling. The insights gained from the missing value analysis, such as the high percentage of missing values in 'Payment Typology 2' and 'Payment Typology 3', can guide further investigation and interpretation of the results.

1. **Exploratory Data Analysis (EDA)**

**3.1 Descriptive Statistics**

Exploratory data analysis started with a careful inspection of the descriptive statistics of the collection by numerical variables in the SPARCS dataset. This step gave us a sense of what the means and the standard deviation were and how skewed the distribution was towards positives/negatives.

**Overview of Key Numerical Variables**

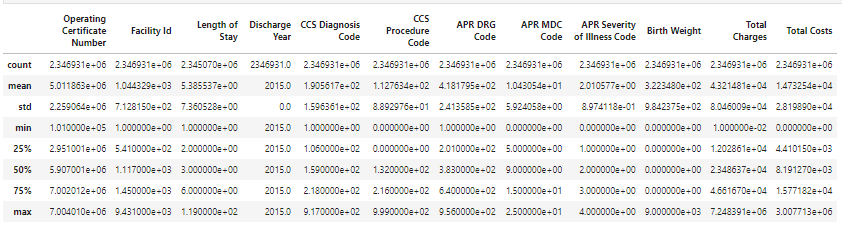
1. **Length of Stay:**

* Mean: The median length of stay for our hospitals patients was approximately 5.39 days, which is the average period inpatients reside in them.
* Standard Deviation: That standard deviation as 7.36 days available with this variation in length of stay indicates a wide range either from the shortest visits for simple issues to the longest for more complex and severe conditions.
* Minimum and Maximum: The shortest recorded stay was 1 day, whereas the longest stay for someone was 119 days. This protrudes the presence of both short admittances and severely prolonged stay due to the intensity of their medical conditions.

1. **Total Charges:**

* Mean: The average hospital stay's total bill was at $43,214.81 tone of the charge highlighting the weight of the financial pressure behind the hospitalization.
* Standard Deviation: The standard deviation of 80,460.09 shows that the hospital bill was very wide in range that was perhaps attributed to the myriad of different conditions, treatments, and procedures.
* Minimum and Maximum: The minimum charge stability obtained was just 0.01 cent that might have been exception or particular observed from the next experiment that gave minimal charges. On one hand, for a less complicated case, the average charge can reach $7,248,391, which could mean that there was something very complicated or a large-scale demand for the resource.

1. **Total Costs:** 
   * Mean: The average total cost incurred by hospitals for an inpatient stay was $14,732.54, offering a perspective on the actual expenses involved in providing care.
   * Standard Deviation: With a standard deviation of $28,198.90, there is a substantial variation in costs across different hospital stays, reflecting the diversity in resource utilization and medical complexity.
   * Minimum and Maximum: The minimum recorded cost was $0, which could indicate cases with no associated costs or potential data entry issues. The maximum cost, reaching $3,007,713, underscores the presence of highly resource-intensive and specialized treatments.



**Analysis of Diagnostic and Procedural Codes**

1. **APR DRG Code and CCS Procedure Code:** 
   * The dataset utilizes APR DRG (Diagnosis-Related Group) codes and CCS (Clinical Classifications Software) procedure codes to categorize hospital cases based on diagnoses and procedures performed.
   * The wide range of APR DRG codes, spanning from 1 to 956, reflects the diversity of medical conditions treated, ranging from simple to highly complex cases.
   * Similarly, the CCS procedure codes range from 0 to 999, indicating a broad spectrum of procedures performed, including cases with no procedures to those involving multiple interventions.
2. **APR Severity of Illness Code:**

* The APR severity of illness codes aid in the identification of the extent of severity in each hospital stay. The scores range from 0 when no illness is present to 4 in case of critical illness.
* The mean severity of illness code of 2.01 which is average of moderate level of severity of patient’s illness.
* The mortality rate at code 4 demonstrates the need for versatility in approaches to critical care and specialized management of more severe cases.

**Dispersion and Distribution Insights**

The extreme deviation observed in variables such as Length of Stay, Total Charges and Total Amount spent, highlights the intrinsic diversity and discrepancy in the healthcare management. The bigger standard deviations that are seen here to imply that hospital stays prediction of utilization of resources and financial aspects can be quite tough due to many state of the patient and nature of treatment that vary greatly.

On the next, it became clear that the variables' distribution, as shown in the histograms, has a right-skew characteristic which is a common feature in most of the healthcare data. In other words, most visits to the hospital incur only small costs and endure for just short period while very few of the cases, which are typically critical and require some special care, account for a higher cost level and a longer-term stay.

This descriptive statistics and distribution observations build a strong basis for displaying the appearance and variability within the dataset SPARCS. They underline the importance of the robust analytics and the value of the modelling tools to the correct prognosis of hospital inpatient stays considering the large number of factors which take part in the calculation of the length of stay, costs, and resources utilization.

**3.2 Analysis of Categorical Variables**

The demographic data of the SPARCS dataset includes categorical variables such as gender, race, and age which allows for examination and evaluation of the demographic factors influencing hospital inpatient discharges. Let's delve into the frequency distributions of these variables to uncover key patterns and trends.

**Geographic Distribution**

* **Health Service Area:** Firstly, most of the discharges from the inpatient treatment were recorded in the region of New York City (1,096,733), Long Island (339,039) and then Hudson Valley, which issued some of the largest discharge numbers (245,594). This distribution is a reflection of population density and health infrastructure availability in these areas, and New York City with the population is the most populous of the regions considered and has a greater number of healthcare facilities when compared to others.
* **Hospital County:** The counties having the greatest number of inpatient discharges are: Manhattan with 406,229, Kings with 248,475, Queens with 197,199, Bronx with 187,979, and Nassau with 181,015. Counties designated as mental health priorities, and all located in the metropolitan area of the New York City, are the main producers of the discharges, proving the impact of the high number of people in these highly populated areas.

A screenshot of a graph

Description automatically generated

**Patient Demographics**

* Gender: The females (1,307,574) had a slightly higher number of discharges compared to males (1,039,309). Perhaps this gender imbalance can be attributed to different healthcare patterns and specific health states, or the life expectation.
* Race: First of all, White patients (1,334,890) are the biggest racial group, and the second place is occupied by Other Race (544,720) and Black/African American (444,883) patients. The distribution of racial groups carries the demographic characteristic of the population being served by the hospitals that provided the services in the dataset.
* Ethnicity: Moreover, the population trend showed most of the patients were not Spanish nor Hispanic (1954846), while the remaining 278838 patients were Spanish/Hispanic. This ethnic composition of the patient population will reveal the cultural and language diversity and may be used to sensitize us to the cultural competency and communication skills required.

**Admission Characteristics**

* Type of Admission: Such the highest numbers of emergency admissions were observed (1,487,948) then the elective admissions (447,323), and the others were newborn admissions (227,196). The fact that about half of all attendances at the emergency departments are emergency admissions is a clear demonstration of the extent to which hospitals play the role of the providers of emergency care.
* Patient Disposition: The majority of patients (1,572,079) were discharged to home or self-care, indicating successful treatment and recovery. The remaining identified disposition qualities were home health with home services (304,373), nursing home with skilled care (224,088), and inpatient rehabilitation (44,544). A distribution of the range of patients’ post-hospital dispositions indicates the availability of various post-hospital services reflecting different levels of the need for urgent medical care.

**Medical and Surgical Classifications**

* APR Severity of Illness Description: Moderate severity (897,411) was the most prevalent, followed by minor (785,636) and major (517,086) severity levels. Such distribution shows that nearly half of admissions were for patients with moderately severe and severe medical complications needing plenty of medical help and resources.
* APR Medical Surgical Description: The majority (83%) of stays (1,779,268) were medical, while 21% (567,552) were surgical, respectively. Thus, the summary reflects that medical complications have a predominant role, and the number of surgeries is smaller in the inpatient sector.

**Financial Aspects**

1. Payment Typology:
   * Medicare (883,250) and Medicaid (731,012) were the primary payers for a significant portion of inpatient discharges.
   * Private health insurance (334,527) and Blue Cross/Blue Shield (275,583) also accounted for a substantial number of discharges.
   * The payment typology distribution reflects the mix of public and private healthcare coverage and the financial arrangements for inpatient care.

The outcomes of the categorical variable analysis, on the other hand, help to better understand such aspects as demographics, pathology and finances of inpatient episodes in the SPARCS dataset. They give rise to the variety of populations of patients, multiplicity of admission types and medical conditions, and the complex connections between healthcare services and payment systems.

**3.3 Correlation Analysis**

To obtain more exact knowledge in SPARCS dataset about the correlation between the variables the data Pearson correlation analysis was done. A correlation matrix look at the degree to which pairs of variables are linearly linked through the intensity and direction of those relationships.

Let's examine the key findings from the correlation analysis.

**Length of Stay**

1. Total Charges and Total Costs:
   * Length of stay has a strong positive correlation with both total charges (0.701230) and total costs (0.717270).
   * This indicates that as the length of stay increases, there is a corresponding increase in the financial burden associated with the hospital stay.
2. APR Severity of Illness Code:
   * Length of stay has a moderate positive correlation with the APR severity of illness code (0.362886).
   * This suggests that patients with more severe conditions tend to have longer hospital stays, likely due to the complexity of their medical needs.
3. Other Variables:
   * Length of stay has weak positive correlations with CCS diagnosis code (0.114503), APR DRG code (0.078750), and APR MDC code (0.106159).
   * These correlations indicate that certain diagnoses, diagnosis-related groups, and major diagnostic categories may be associated with slightly longer hospital stays.

A screenshot of a graph

Description automatically generated

**Total Charges and Total Costs**

1. Length of Stay:
   * As we found earlier, total charges and total costs have strong positive correlations with length of stay.
   * This highlights the direct impact of prolonged hospital stays on the financial aspects of healthcare.
2. APR Severity of Illness Code:
   * Total charges (0.330829) and total costs (0.315720) have moderate positive correlations with the APR severity of illness code.
   * This suggests that more severe conditions tend to incur higher healthcare costs and charges, likely due to the intensity of treatment and resource utilization.
3. Other Variables:
   * Total charges and total costs have weak negative correlations with CCS diagnosis code, APR DRG code, and APR MDC code.
   * These correlations indicate that certain diagnoses, diagnosis-related groups, and major diagnostic categories may be associated with slightly lower healthcare costs and charges.

**Other Notable Correlations**

1. CCS Diagnosis Code and APR DRG Code:
   * CCS diagnosis code has a moderate positive correlation with APR DRG code (0.440195).
   * This suggests that certain diagnosis codes are more frequently associated with specific diagnosis-related groups.
2. APR DRG Code and APR MDC Code:
   * APR DRG code has a very strong positive correlation with APR MDC code (0.968579).
   * This indicates that diagnosis-related groups are closely tied to major diagnostic categories, as expected based on the hierarchical nature of these classification systems.

**3.4 Relationship between Total Cost and CCS Diagnosis Description across Facilities**

To examine the relation between total cost and diagnosis CCS name across various health facilities, an analysis was performed by categorizing the data by the name of the facility and diagnosis type name and calculation of average total cost for each of the combinations. Such investigation gives the general picture of how the cost of care changes among different diagnoses across different facilities.

**Key Findings**

1. Highest Average Total Cost:
   * The diagnosis of "Nephritis; nephrosis; renal sclerosis" at Henry J. Carter Specialty Hospital had the highest average total cost of $667,453.14.
   * This suggests that the treatment of kidney-related conditions at this specific facility is associated with substantial healthcare costs.
2. Complications and Infections:
   * "Complications of surgical procedures or medical care" at Henry J. Carter Specialty Hospital had the second-highest average total cost of $636,134.82.
   * "Septicemia (except in labor)" at the same facility also had a high average total cost of $504,184.41.
   * These findings highlight the significant financial impact of complications and infections on healthcare costs, particularly at specialized facilities.

A graph of different colored bars

Description automatically generated with medium confidence

1. Neonatal and Pediatric Conditions:
   * "Short gestation; low birth weight; and fetal growth retardation" at Woodhull Medical & Mental Health Center had an average total cost of $460,263.02.
   * "Respiratory distress syndrome" at New York Presbyterian Hospital - Columbia Presbyterian Center also had a high average total cost of $414,651.60.
   * These results underscore the substantial costs associated with the care of premature and critically ill newborns and infants.
2. Complex Medical Conditions:
   * "Non-Hodgkin's lymphoma" at a confidential facility had an average total cost of $437,979.24.
   * "Respiratory failure; insufficiency; arrest (adult)" at Henry J. Carter Specialty Hospital had an average total cost of $433,597.82.
   * "Encephalitis (except that caused by tuberculosis or sexually transmitted disease)" at North Central Bronx Hospital had an average total cost of $429,753.19.
   * These findings highlight the high costs associated with the treatment of complex medical conditions that require specialized care and intensive resources.

The results reveal that the treatment of the most complex illnesses, accompanied by a need for specialized care and massive resources, is expensive, indeed.

The cost analysis and CCS diagnosis description reviews from facilities to facilities can give us the knowledge to identify the variation in healthcare costs because of diagnostic specifics and the facilities services. This kind of data will be quite revealing for healthcare providers, health insurers, policymakers, and patients. By utilizing this data, stakeholders can work together to create customizable plans for fund allocation, budgeting, quality improvement, cost reduction, patient education, and applicable decision-making. Health care facilities can be run in a way that is optimal with regards to resources use and can introduce best practices that will lead to efficiency improvement and lower expenses. Reimbursement and the cost containment strategies at the payer and insurance company level can be negotiated and developed. Patients can be empowered to make their health care decisions, since they will have transparent cost information on hand. Policy makers have an opportunity to highlight the unequal cost situations and make strategic solutions to maintain equitable treatment of care and cost reduction.

**4. Prediction Modelling 1: Predicting Total Costs**

**4.1 Problem Statement and Analysis:**

Accurately determining the overall cost of a hospital inpatient stay is a fundamental job for healthcare providers, payers and patients respectively. Appropriate cost estimation assists in planning, budgeting and management of resources, as well as making decisions. The question here is how we can estimate the total expenses when we have some given features like patient age, gender, race, ethnicity, admission type, length of stay, diagnoses, procedures, severity of illness, and hospital characteristics.

To resolve this question, we are planning to use the machine learning approach which will take the necessary elements from the SPARCS dataset and build a predictive model. The model can be trained on historical data and the outputs can be examined to determine the effectiveness and accuracy of cost prediction based on the studied factors.

**4.2 Linear Regression Model**

**4.2.1 Model Overview:**

Linear regression is a simple model algorithm that underlines a linear relation of the input features and the response variable. In this case, we will be using linear regression to predict the total costs pertaining to a hospital inpatient stay on the given predictor characteristics.

**4.2.2 Model Training and Evaluation:**

Dataset was partitioned into 80% training set and 20% validation set while keeping that structure. The training set was used for training a linear regression model which was equipped with a pipeline that contains steps such as categorical variable encoding and median imputation in numerical variables.

**4.2.3 Model Performance:**

The trained linear regression model was evaluated on the validation set using several performance metrics:

* Mean Absolute Error (MAE): 4264.45
* Mean Squared Error (MSE): 144006157.98
* Root Mean Squared Error (RMSE): 12000.26
* R-squared (R^2): 0.8205

MAE is measured in $4,264 and shows that model predictions are on average more than $4,264 away from the real total costs. The RMSE is the measure of average loudness of the prediction errors, and this value is roughly $12,000. With the R-squared value of 0.8205, 82.05% of the variance in the cost would be captured by the linear regression model.

A number on a white background

Description automatically generated

**4.2.4 Prediction vs. Actual Plots:**

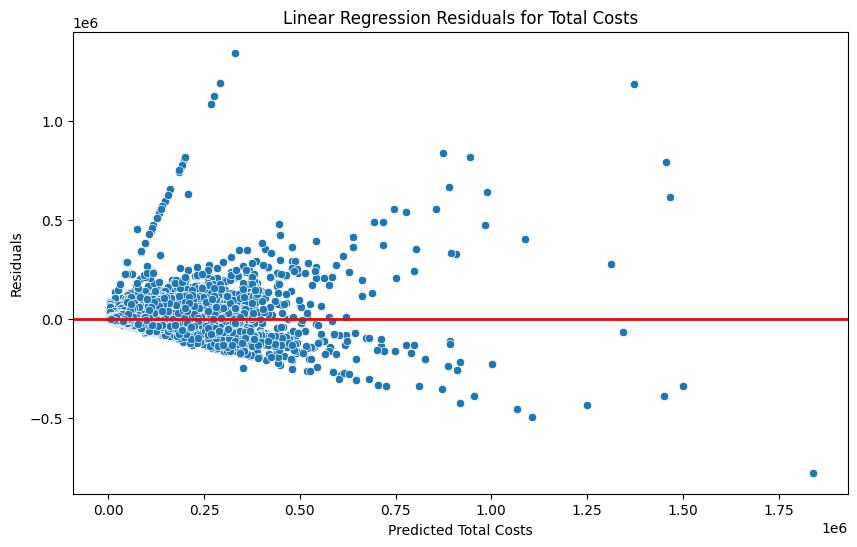
Scatterplot generate a correlation of predicted total costs and. The actual total costs graph gives a clearer expression of the model’s effectiveness. The blue dots represent the specific forecasts, whereas the red line is the perfect prediction (where the forecast costs are exactly the same as the actual costs). The plot reveals a line that follows a linear trend where the predictions are very close to the perfect prediction line. Nevertheless, there is some dispersion of the model's forecasts at higher values of expenses, which shows that the model doesn't provide the best forecasts.

A graph with a red line and blue dots

Description automatically generated

**4.2.5 Residual Plots:**

The residual plot will represent the variation between the real and predicted values of the overall costs. Residuals are the vertical distances between blue dots and the zero residual red line. In ideal conditions, residuals must be arbitrarily scattered on the zero line without patterns of any kind. For this model, the residuals are more or less even (even though there is a slight downward trend), which means that the model tends to overestimate costs for high predicted values.



**4.2.6 Interpretation and Implications:**

The linear regression model's performance metrics and plots show that it may be able to fit the given data and predict total costs based on the features that have been given. By the R-squared value of the forecasting power, it can be seen that the selected features are highly related with the total costs. Nevertheless, one should acknowledge that the algorithm's forecasts are not always accurate and there is still a space for the improvement. The plot of the residuals displays some systematic features, thereby indicates the linear assumption disregards the complexities of how the features affect total expenses.

**4.3 Random Forest Model**

**4.3.1 Model Overview:**

Random forest has an ensemble learning algorithm through which multiple decision trees work together to produce predictive results. It is known for its ability to learn model patterns in data, and handle complex interactions between features. The next segment involves the application of the random forest algorithm for classifying the total cost associated with hospital inpatient care based on the input features.

**4.3.2 Model Training and Evaluation:**

The random forest model was based upon the same training and validation splits as the linear regression model. The pipeline was set for dealing with categorical variables (one-hot encoding) and numerical variables (imputation with median values). The random forest model was configured with the following hyperparameters: n\_estimators=5, max\_depth=5, min\_samples\_leaf=5, max\_features='sqrt', n\_jobs=-1, random\_state=0.

**4.3.3 Model Performance:**

The trained random forest model was evaluated on the validation set using the same performance metrics as the linear regression model:

* Mean Absolute Error (MAE): 8646.48
* Mean Squared Error (MSE): 536945710.04
* Root Mean Squared Error (RMSE): 23172.09
* R-squared (R^2): 0.3305

A screenshot of a computer screen

Description automatically generated

Random forest model displays not only higher MAE and RMSE values but also higher prediction errors in the absolute sense comparing to the linear regression model. R-squared value of 0.3305 implies that the random forest model explains just 33.05% of the variance in the total cost of the production of automobiles, which is much worse than the linear regression model.

**4.3.4 Prediction vs. Actual Plots:**

Scatter plot of estimated total costs against. actual total costs for random forest model follows less linear pattern in comparison to the linear regression model. Predictions are actually more spread and deviate more from the perfect prediction line especially at higher costs. This demonstrates the fact that the random forest model fails in distinguishing between high-cost and low-cost cases.

A graph with a red line

Description automatically generated

**4.3.5 Residual Plots:**

There is a more distinct pattern on the residual plot for the random forest model compared to the linear regression model. It is observable that the overvaluation is greater as the predicted costs fall and the undervaluation is higher as predicted costs rise. Residuals are not located around the zero line evenly, which means our model assumptions might not be perfectly valid.

A graph of a graph showing a number of blue dots

Description automatically generated

**4.3.6 Interpretation and Implications:**

From the random forest model performance metrics and plots, we observe that it is not as precise in predicting the total costs as is the linear regression model. The bigger MAE and RMSE values, along with the worse r-squared value, show that the random forest model does not portray the relationship between the features and total costs well. The residual plot indicates the presence of a potential systematic error, which manifests itself through overestimation at lower values and underestimation at higher values. This shows that the forest model is not the best fit for this issue and data set.

**4.4 Model Performance Comparison**

Analysing the linear regression and random forest model accuracy, the linear regression model has better accuracy than the random forest model in estimating the total costs associated with hospital inpatient stays.

The linear regression model gives a lower difference and error rate, that means smaller average prediction errors. It also has a higher R squared value, that means it covers the larger proportion of the variance in the total costs. The Scatter Plot between the Predicted and: the actual cost for the linear regression prediction model is more linear and the predictions are closer to the perfect prediction line as compared to the original prediction. Additionally, the residual plot for the adjusted linear regression model also appears to be more balanced and closer to the zero line, showing better alignment with the assumptions of the model.

The second model, random forest, has even bigger MAE and RMSE values and therefore higher prediction errors. Its R-squared value is highly low showing that the term alone accounts for a low proportion of the variation in total costs. Random forest model's scatter plot depicts an irregular trend away from that of the perfect line and the predictions farther from it for higher cost values. The residual plot produced by the random forest model indicates a predictable pattern of over- and underestimation, probably indicating that the model assumptions do not describe reality completely.

Finally, one can say that using the above-mentioned features from the SPARCS dataset to predict the total cost for in-patient hospital stays the linear regression model performs better than random forest model. It can be easier more interpretable and precise for calculating overal costs, and also acting as enabling and vital instrument to aligned priorities of healthcare payers and providers.

1. **Prediction Modelling 2: Predicting Length of Stay**

**5.1 Problem Statement and Analysis:**

Forecasting the length of patient who is in a hospital is a pivotal action in healthcare management. Accurate predicted can enable hospitals to well use their resources, to optimize patient flow, and to improve the hospitals’ operational performance. The topic considers whether it is possible to know beforehand the length of stay for a patient based on their diagnosis, health status, and the plans for the medical process for that individual.

To answer the above question, we will have to use the SPARCS dataset, along with some appropriate features like CCS diagnosis code, CCS procedure code, APR DRG code, APR MDC code, APR severity of illness code, type of admission, patient disposition, age group, and gender. Through the use of our predictive models which were developed using these features, we aim to calculate LOS (Length of Stay) for each individual patient.

This variable was treated as a categorical variable, as the length of stay values '120 +' were converted to their representative number: 121, that represents stay longer than 120 days. Apart from that, we removed the cases of length stay value missing from the records in order to maintain data accuracy.

**5.2 Linear Regression Model**

**5.2.1 Model Overview:**

Linear regression is a simple and interpretable algorithm that assumes a linear relationship between the input features and the target variable. In this case, we will use linear regression to predict the length of stay based on the selected features.

**5.2.2 Model Training and Evaluation:**

The dataset was divided into two sets of training and validation, 80% data for training set and 20% for validation set. Similarly, the linear regression model has been trained using a pipeline that involves the steps for handling categorical variables and numerical variables like one hot encoding and imputation with median values respectively.

**5.2.3 Model Performance:**

The trained linear regression model was evaluated on the validation set using several performance metrics:

* Mean Absolute Error (MAE): 3.4877
* Mean Squared Error (MSE): 44.7217
* Root Mean Squared Error (RMSE): 6.6874
* R-squared (R^2): 0.1774

A number on a white background

Description automatically generated

The weighted average error of the model is that on average its prediction fail to approximate the actual length of stay by about 3.49 days. RMSE is the average of all the magnitude of prediction errors, which provides value of about 6.69 days. The R-squared value of 0.1774 suggests that the linear regression model explains only 17.74% of the variance in the length of stay.

**5.2.4 Prediction vs. Actual Plots:**

The scatter plot of predicted length of stay vs. actual length of stay (Image 1) shows the model's performance visually. The blue dots represent individual predictions, while the red line indicates perfect predictions (where predicted length of stay equals actual length of stay). The plot reveals that the model's predictions have a positive correlation with the actual values, but there is significant scatter, especially for longer lengths of stay.

A graph with a red line

Description automatically generated

**5.2.5 Residual Plots:**

The plot of the residual is a scatterplot that shows the disparities between the lengths of actual stays and the length of stays predicted. The best-case scenario would be the residuals to be random around the zero line with no obvious patterns whatsoever. The residuals here in this case are indicated by the existence of a somewhat random pattern, where the forecasts tend to be higher for shorter stays and lower for the longer ones.

A graph of a line

Description automatically generated with medium confidence

**5.3 Random Forest Model**

**5.3.1 Model Overview**

Random Forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. It is known for its ability to capture non-linear relationships and handle complex interactions between features. We will apply the random forest algorithm to predict the length of stay based on the selected features.

**5.3.2 Model Training and Evaluation**

The random forest model was trained using the same training and validation splits as the linear regression model. The pipeline included preprocessing steps for handling categorical variables (one-hot encoding) and numerical variables (scaling with StandardScaler). The random forest model was configured with the following hyperparameters: n\_estimators=10, max\_depth=5, random\_state=42.

**5.3.3 Model Performance**

The trained random forest model was evaluated on the validation set using the same performance metrics as the linear regression model:

* Mean Absolute Error (MAE): 3.0556
* Mean Squared Error (MSE): 36.9989
* Root Mean Squared Error (RMSE): 6.0827
* R-squared (R^2): 0.3195

A screenshot of a computer

Description automatically generated

Compared to the linear regression model, the random forest model has slightly lower MAE and RMSE values, indicating smaller average prediction errors. The R-squared value of 0.3195 suggests that the random forest model explains 31.95% of the variance in the length of stay, which is an improvement over the linear regression model.

**5.3.4 Prediction vs. Actual Plots**

The scatter plot of predicted length of stay vs. actual length of stay for the random forest model (Image 2) shows a similar pattern to the linear regression model. The predictions have a positive correlation with the actual values, but there is still significant scatter, particularly for longer lengths of stay.

A graph with a red line

Description automatically generated

**5.3.5 Residual Plots**

The residual plot for the random forest model (Image 2) exhibits a more random distribution of residuals compared to the linear regression model. However, there are still some patterns visible, with the model overestimating shorter stays and underestimating longer stays.

A graph of a random forest residuals

Description automatically generated

**5.4 K-Nearest Neighbors (KNN) Model**

**5.4.1 Model Overview**

K-Nearest Neighbors (KNN) is a non-parametric approach that uses the similarity of input features to data points in training to infer predictions. Its functioning is based on the thought that more likely data points produce approximately the same output. Here we will utilize KNN to predict the stay length involving the mentioned features.

**5.4.2 Model Training and Evaluation**

The KNN model was trained using the same training and validation splits as the previous models. The pipeline included preprocessing steps for handling categorical variables (one-hot encoding) and numerical variables (scaling with StandardScaler). The KNN model was configured with the following hyperparameters: n\_neighbors=5, leaf\_size=50.

**5.4.3 Model Performance**

The trained KNN model was evaluated on the validation set using the same performance metrics as the previous models:

* Mean Absolute Error (MAE): 2.9631
* Mean Squared Error (MSE): 36.6876
* Root Mean Squared Error (RMSE): 6.0570
* R-squared (R^2): 0.3252

A number on a white background

Description automatically generated

The KNN model has the lowest MAE and RMSE values among the three models, indicating the smallest average prediction errors. The R-squared value of 0.3252 suggests that the KNN model explains 32.52% of the variance in the length of stay, which is slightly higher than the random forest model.

**5.4.4 Prediction vs. Actual Plots**

The scatter plot of predicted length of stay vs. actual length of stay for the KNN model (Image 3) shows a similar pattern to the previous models. The predictions have a positive correlation with the actual values, but there is still significant scatter, especially for longer lengths of stay.

A graph with a red line

Description automatically generated

**5.4.5 Residual Plots**

The residual plot for the KNN model (Image 3) exhibits a random distribution of residuals, with no clear patterns. The residuals are more evenly scattered around the zero-line compared to the linear regression and random forest models.

A graph of a line

Description automatically generated with medium confidence

**5.5 Model Performance Comparison**

Comparing the performance of the linear regression, random forest, and KNN models, we can draw the following conclusions:

1. **Prediction Accuracy:**

* The KNN model demonstrated the smallest average prediction errors (2.9631 for the MAE and 6.0570 for the RMSE) among all the studied approaches, suggesting that it has the lowest error compared to the two others.
* The Random Forest model displays worse MAE (3.0556) and RMSE (6.0827) in comparison to the KNN model, but remains the best among them, outperforming the linear regression model.
* In case of the linear regression model, the highest MAE that was computed is 3.4877 and the RMSE which is 6.6874, implying that it has the largest prediction errors in the average.

1. **Explained Variance:**

* Our study indicates KNN has the highest R-squared value (0.3252) among the three models considered. This implies that KNN can explain highest variability in the length of stay among the models examined.
* The random forest algorithm which has a slightly lower R-squared value (0.3195) than KNN model, however, still has the highest R-squared value among the three models here (more than linear regression model).
* The linear regression model that has the lowest R2-values (0.1774) explain the least variance of the length of stay so that is that model is less effective in predicting length of stay.

1. **Residual Plots:** 
   * The KNN model has the greatest scattering of residuals, depicting a random pattern without any apparent pattern, so it is best among the three models.
   * The pattern of random residuals' distribution in the random forest model can be observed here, identical to the linear regression model, but with some residuals underestimate and overestimate still.
   * The linear regression model has the most visible patterns in the residual plot, suggesting that it has the poorest fit among the three models.

Briefly, the KNN algorithm was seen to be the best performing one of all the models tested and the task of predicting the length of stay is indeed very complex. The hospital stay for each patient is influenced by several factors, including patient's health condition, response to treatment, and unforeseen complications, which challenges accurately predicting the length of hospitalizations.

6. **Conclusion and Recommendations**

**6.1 Summary of Key Findings:**

From this study of SPARCS data, we gained predominant understanding of inpatient hospital stays and their charges controlled by different circumstances. We have achieved this by using exploratory data analysis (EDA) and predictive modelling, which have unveiled the critical underlying patterns, trends and links, and therefore aid in formulating healthcare related decisions and resource allocation.

EDA exposed a lot of significant variation in stay duration, charges and all the medical bills across the diverse patient demographics, diagnosis and characteristics of the hospitals which were observed. We pinpointed the overall conclusion of the inpatient stays’ common diagnosis, procedure, and their respective payment codices. The cross correlation revealed strong positivity between length of stay and total cost, showing their close relationship and the impact of severity of illness on healthcare use and expenses.

For the predictive modelling phase, we have done the work of creating two models: one that estimates the total costs and one that is a length of stay for a hospital inpatient stay. For predicting total costs, the linear regression model performed better the random forest model, obtaining an R-squared value of 0.8205 which attested to the possibility of estimating the costs based on patient and hospital deciles. The KNN model is, hence, shown to be more appropriate in this context, given that it has the smallest prediction errors and the biggest explained variance among the applied models.

* 1. **Recommendations for Healthcare Stakeholders**

Based on the findings of this analysis, we propose the following recommendations for healthcare stakeholders:

* Leverage predictive modelling for resource planning and budgeting: Hospitals and health insurers can use the cost estimation model to predict the financial impact of inpatient stays, thereby ensuring that resources are allocated properly. The presence of a stay length prediction model can be employed in capacity planning, staffing decisions, and patient flow management.
* Focus on preventive care and early intervention: The analysis showed the level of illness impacted on healthcare use and its costs. By putting resources into preventive care and early intervention programs it becomes possible to decrease the number of serious conditions and the speed at which they develop and this leads to lowering of the healthcare costs.
* Identify and address disparities in healthcare access and outcomes: The EDA showed variations in health care utilization and costs within demographic groups and geographic areas. Healthcare entities need to evaluate and try to find the solutions for the disparities, in order to guarantee the equal access to the care and improve the patient's outcome.

By continuously refining and expanding the analytical approaches, healthcare organizations can unlock the full potential of data-driven insights to improve patient care, optimize resource utilization, and drive sustainable healthcare system improvements.

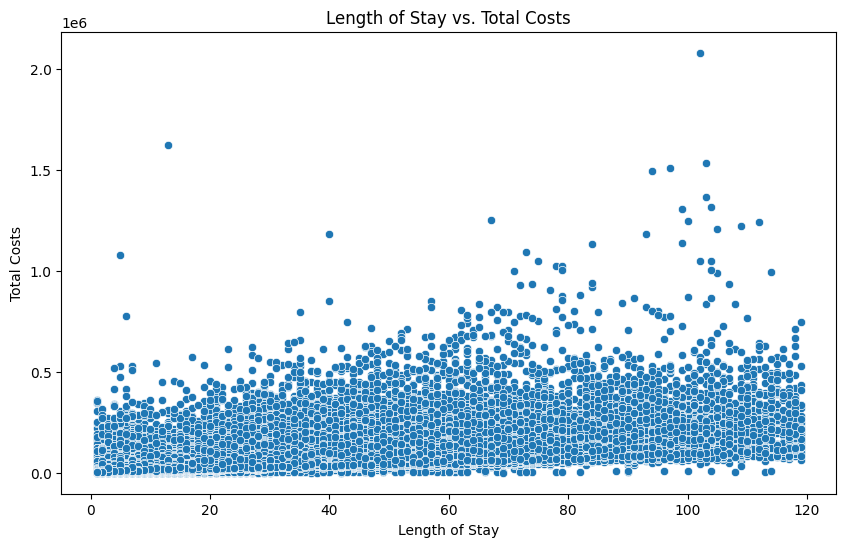
**References:**

1. Bhuvan, M. S., Kumar, A., Zafar, A., & Kishore, V. (2016). Identifying diabetic patients with high risk of readmission using data mining techniques. In Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (pp. 587-593). IEEE.
2. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1721-1730).
3. Daghistani, T. A., Elshawi, R., Sakr, S., Ahmed, A. M., Al-Thwayee, A., & Al-Mallah, M. H. (2019). Predictors of in-hospital length of stay among cardiac patients: A machine learning approach. International Journal of Cardiology, 288, 140-147.
4. Frizzell, J. D., Liang, L., Schulte, P. J., Yancy, C. W., Heidenreich, P. A., Hernandez, A. F., ... & Fonarow, G. C. (2017). Prediction of 30-day all-cause readmissions in patients hospitalized for heart failure: comparison of machine learning and other statistical approaches. JAMA Cardiology, 2(2), 204-209.
5. Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., ... & Dean, J. (2018). Scalable and accurate deep learning with electronic health records. NPJ Digital Medicine, 1(1), 1-10.
6. Shameer, K., Johnson, K. W., Yahi, A., Miotto, R., Li, L. I., Ricks, D., ... & Dudley, J. T. (2017). Predictive modeling of hospital readmission rates using electronic medical record-wide machine learning: a case-study using Mount Sinai Heart Failure Cohort. In Pacific Symposium on Biocomputing 2017 (pp. 276-287).

**Appendix**

A graph of blue dots

Description automatically generated



A graph showing a number of blue dots

Description automatically generated

A group of blue and white graphs

Description automatically generated

Python Code

import pandas as pd

# Load the dataset

file\_path = 'Hospital\_Inpatient\_Discharges.csv'

data = pd.read\_csv(file\_path)

data.head()

data.columns

# Check for missing values in each column

missing\_values = data.isnull().sum()

# Check the percentage of missing values in each column

missing\_percentage = (missing\_values / len(data)) \* 100

# Display the columns with missing values

print(missing\_values[missing\_values > 0])

print(missing\_percentage[missing\_percentage > 0])

# Imputing missing values for columns with low missing percentages using the mode

for column in ['Health Service Area', 'Hospital County', 'Operating Certificate Number', 'Facility Id', 'APR Severity of Illness Description', 'APR Risk of Mortality']:

mode\_value = data[column].mode()[0]

data[column].fillna(mode\_value, inplace=True)

# Impute missing values for 'Zip Code - 3 digits' using the mode

zip\_mode = data['Zip Code - 3 digits'].mode()[0]

data['Zip Code - 3 digits'].fillna(zip\_mode, inplace=True)

# Fill missing values in 'Payment Typology 2' and 'Payment Typology 3' with 'Not Available'

data['Payment Typology 2'].fillna('Not Available', inplace=True)

data['Payment Typology 3'].fillna('Not Available', inplace=True)

# Check if there are any missing values left

missing\_values\_updated = data.isnull().sum()

missing\_values\_updated

# Convert 'Operating Certificate Number' and 'Facility Id' to int, handling NaNs by converting them to a type that supports NaN

data['Operating Certificate Number'] = pd.to\_numeric(data['Operating Certificate Number'], downcast='integer', errors='coerce')

data['Facility Id'] = pd.to\_numeric(data['Facility Id'], downcast='integer', errors='coerce')

# Convert categorical fields to category type

categorical\_columns = ['Health Service Area', 'Hospital County', 'Gender', 'Race', 'Ethnicity', 'Type of Admission', 'Patient Disposition', 'APR Severity of Illness Description', 'APR Risk of Mortality', 'APR Medical Surgical Description', 'Payment Typology 1', 'Payment Typology 2', 'Payment Typology 3', 'Emergency Department Indicator', 'Abortion Edit Indicator']

for column in categorical\_columns:

data[column] = data[column].astype('category')

data['Length of Stay'] = pd.to\_numeric(data['Length of Stay'], errors='coerce')

# Recheck data types after conversion

print(data.dtypes)

EDA QUESTION 1: Statistical analysis of both numerical and categorical Variables

# Display summary statistics for numerical columns

numerical\_summary = data.describe()

numerical\_summary

# Display frequency distribution for categorical columns

for column in data.select\_dtypes(include=['category']).columns:

print(f"Frequency distribution for {column}:")

print(data[column].value\_counts())

print("\n")

import matplotlib.pyplot as plt

# Histograms for numerical variables

data.hist(bins=15, figsize=(15, 10))

plt.show()

import matplotlib.pyplot as plt

# Select categorical columns

categorical\_columns = data.select\_dtypes(include=['category']).columns

# Define the number of rows and columns for subplots

n\_cols = 3 # You can adjust this based on your display preferences

n\_rows = (len(categorical\_columns) + n\_cols - 1) // n\_cols # Computes the necessary number of rows

# Create a figure with subplots

fig, axes = plt.subplots(n\_rows, n\_cols, figsize=(n\_cols \* 5, n\_rows \* 5)) # Adjust figure size as necessary

fig.subplots\_adjust(hspace=0.5, wspace=0.5) # Adjust spacing to prevent label overlap

# Plot each categorical variable in a subplot

for i, column in enumerate(categorical\_columns):

ax = axes.flatten()[i]

data[column].value\_counts().plot(kind='bar', ax=ax, title=column)

ax.set\_xlabel(column)

ax.set\_ylabel('Counts')

ax.set\_xticklabels(ax.get\_xticklabels(), rotation=45, ha="right") # Rotate labels to prevent overlap

# If there are any leftover axes, turn them off

for j in range(i+1, n\_cols \* n\_rows):

axes.flatten()[j].axis('off')

# Show plot

plt.show()

EDA Analysis question 2. Correltion analyisis between target varaible and other numerical variables?

import matplotlib.pyplot as plt

import seaborn as sns

# Select only numeric columns for correlation calculation

numeric\_data = data.select\_dtypes(include=['float64', 'int64'])

# Calculate the correlation matrix

correlation\_matrix = numeric\_data.corr()

# Use seaborn to create a heatmap of the correlation matrix

plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Matrix of Numeric Features')

plt.show()

correlation\_matrix

import matplotlib.pyplot as plt

import seaborn as sns

# Length of Stay vs. Total Charges

plt.figure(figsize=(10, 6))

sns.scatterplot(x=data['Length of Stay'], y=data['Total Charges'])

plt.title('Length of Stay vs. Total Charges')

plt.xlabel('Length of Stay')

plt.ylabel('Total Charges')

plt.show()

# Length of Stay vs. Total Costs

plt.figure(figsize=(10, 6))

sns.scatterplot(x=data['Length of Stay'], y=data['Total Costs'])

plt.title('Length of Stay vs. Total Costs')

plt.xlabel('Length of Stay')

plt.ylabel('Total Costs')

plt.show()

# Total Charges vs. Total Costs

plt.figure(figsize=(10, 6))

sns.scatterplot(x=data['Total Charges'], y=data['Total Costs'])

plt.title('Total Charges vs. Total Costs')

plt.xlabel('Total Charges')

plt.ylabel('Total Costs')

plt.show()

EDA Analysis question 3. Can we find a relationship between Total Cost and CCS Diagnosis Description across different facilities?

# Grouping and aggregating the data

costs\_by\_diagnosis\_facility = data.groupby(['Facility Name', 'CCS Diagnosis Description'])['Total Costs'].mean().reset\_index()

# Sorting the results to make them more readable

costs\_by\_diagnosis\_facility\_sorted = costs\_by\_diagnosis\_facility.sort\_values(by='Total Costs', ascending=False)

costs\_by\_diagnosis\_facility\_sorted.head(10)

costs\_by\_diagnosis\_facility\_sorted.reset\_index().head(10)

# Selecting the top N for visualization for clarity, N can be decided based on the data

top\_n = costs\_by\_diagnosis\_facility\_sorted.head(20)

plt.figure(figsize=(10, 8))

sns.barplot(x='Total Costs', y='CCS Diagnosis Description', hue='Facility Name', data=top\_n, dodge=False)

plt.title('Average Total Costs by CCS Diagnosis Description across Facilities')

plt.xlabel('Average Total Cost')

plt.ylabel('CCS Diagnosis Description')

plt.legend(title='Facility Name', bbox\_to\_anchor=(1.05, 1), loc='upper left')

plt.show()

Prediction Question 1:

Question: Can we predict the total costs associated with a hospital inpatient stay based on available features such as the patient’s demographics (age, gender, race, ethnicity), admission type, length of stay, diagnosis, procedures performed, severity of illness, and hospital characteristics?

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.impute import SimpleImputer

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

# Selecting features and target variable

X = data.drop('Total Costs', axis=1) # all columns except Total Costs

y = data['Total Costs'] # Total Costs column

# Handling categorical variables with OneHotEncoding

categorical\_cols = [cname for cname in X.columns if X[cname].dtype == "object" or X[cname].dtype.name == "category"]

# Preprocessing for numerical data

numerical\_cols = [cname for cname in X.columns if X[cname].dtype in ['int64', 'float64']]

# Preprocessing for categorical data

categorical\_transformer = Pipeline(steps=[

('imputer', SimpleImputer(strategy='most\_frequent')),

('onehot', OneHotEncoder(handle\_unknown='ignore'))

])

# Bundling preprocessing for numerical and categorical data

preprocessor = ColumnTransformer(

transformers=[

('cat', categorical\_transformer, categorical\_cols),

('num', SimpleImputer(strategy='median'), numerical\_cols)

])

# Splitting data into training and validation sets

X\_train, X\_valid, y\_train, y\_valid = train\_test\_split(X, y, train\_size=0.8, test\_size=0.2, random\_state=0)

from sklearn.linear\_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.ensemble import GradientBoostingRegressor

# Model pipelines

linear\_regression\_pipeline = Pipeline(steps=[('preprocessor', preprocessor),

('model', LinearRegression())])

random\_forest\_pipeline = Pipeline(steps=[

('preprocessor', preprocessor),

('model', RandomForestRegressor(n\_estimators=5, max\_depth=5, min\_samples\_leaf=5, max\_features='sqrt', n\_jobs=-1, random\_state=0))

])

# Train the models

linear\_regression\_pipeline.fit(X\_train, y\_train)

random\_forest\_pipeline.fit(X\_train, y\_train)

# Function to evaluate models

def evaluate\_model(model, X\_valid, y\_valid):

predictions = model.predict(X\_valid)

mae = mean\_absolute\_error(y\_valid, predictions)

mse = mean\_squared\_error(y\_valid, predictions)

rmse = mean\_squared\_error(y\_valid, predictions, squared=False)

r2 = r2\_score(y\_valid, predictions)

print(f"MAE: {mae}")

print(f"MSE: {mse}")

print(f"RMSE: {rmse}")

print(f"R^2: {r2}")

return predictions

def plot\_predictions(model, X\_valid, y\_valid, model\_name):

predictions = model.predict(X\_valid)

plt.figure(figsize=(10, 6))

sns.scatterplot(x=y\_valid, y=predictions)

plt.title(f'{model\_name} Predictions vs. Actual for Total Costs')

plt.xlabel('Actual Total Costs')

plt.ylabel('Predicted Total Costs')

plt.plot([y\_valid.min(), y\_valid.max()], [y\_valid.min(), y\_valid.max()], color='red', lw=2) # Line for perfect predictions

plt.show()

def plot\_residuals(model, X\_valid, y\_valid, model\_name):

predictions = model.predict(X\_valid)

residuals = y\_valid - predictions

plt.figure(figsize=(10, 6))

sns.scatterplot(x=predictions, y=residuals)

plt.title(f'{model\_name} Residuals for Total Costs')

plt.xlabel('Predicted Total Costs')

plt.ylabel('Residuals')

plt.axhline(0, color='red', lw=2) # Line at zero for no residuals

plt.show()

# Evaluate Linear Regression Model

print("Linear Regression Performance:")

lr\_predictions = evaluate\_model(linear\_regression\_pipeline, X\_valid, y\_valid)

# Plot for Linear Regression

plot\_predictions(linear\_regression\_pipeline, X\_valid, y\_valid, 'Linear Regression')

# Residual plots for Linear Regression

plot\_residuals(linear\_regression\_pipeline, X\_valid, y\_valid, 'Linear Regression')

# Evaluate Random Forest Model

print("\nRandom Forest Performance:")

rf\_predictions = evaluate\_model(random\_forest\_pipeline, X\_valid, y\_valid)

# Plot for Random Forest

plot\_predictions(random\_forest\_pipeline, X\_valid, y\_valid, 'Random Forest')

# Residual plots for Random Forest

plot\_residuals(random\_forest\_pipeline, X\_valid, y\_valid, 'Random Forest')

Predicting Question 2: Predicting Length of Stay:

Question: Is it possible to predict the length of stay for a patient given their diagnosis, severity of illness, and the type of procedures they will undergo?

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.preprocessing import OneHotEncoder, StandardScaler

from sklearn.impute import SimpleImputer

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

# Replace '120 +' with 121 (or another number based on your understanding of the data)

data['Length of Stay'] = data['Length of Stay'].replace('120 +', 121)

# Convert 'Length of Stay' to numeric

data['Length of Stay'] = pd.to\_numeric(data['Length of Stay'])

data = data.dropna(subset=['Length of Stay'])

# Selecting relevant features

features = [

'CCS Diagnosis Code', 'CCS Procedure Code', 'APR DRG Code', 'APR MDC Code',

'APR Severity of Illness Code', 'Type of Admission', 'Patient Disposition',

'Age Group', 'Gender'

]

# Select features and target

X = data[features]

y = data['Length of Stay']

# Handling categorical variables with OneHotEncoding and numerical with StandardScaler

categorical\_cols = [col for col in X.columns if X[col].dtype == 'object']

numerical\_cols = [col for col in X.columns if X[col].dtype in ['int64', 'float64']]

# Preprocessing for categorical data

categorical\_transformer = Pipeline(steps=[

('imputer', SimpleImputer(strategy='most\_frequent')),

('onehot', OneHotEncoder(handle\_unknown='ignore'))

])

# Preprocessing for numerical data

numerical\_transformer = Pipeline(steps=[

('imputer', SimpleImputer(strategy='median')),

('scaler', StandardScaler()) # Scaling is generally helpful for KNN

])

# Bundle preprocessing for numerical and categorical data

preprocessor = ColumnTransformer(

transformers=[

('cat', categorical\_transformer, categorical\_cols),

('num', numerical\_transformer, numerical\_cols)

])

# Split data into train and validation sets

X\_train, X\_valid, y\_train, y\_valid = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

from sklearn.linear\_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.neighbors import KNeighborsRegressor

# Linear Regression

linear\_regression\_pipeline = Pipeline(steps=[

('preprocessor', preprocessor),

('model', LinearRegression())

])

# Random Forest Regression

random\_forest\_pipeline = Pipeline(steps=[

('preprocessor', preprocessor),

('model', RandomForestRegressor(n\_estimators=10, max\_depth=5, random\_state=42))

])

random\_forest\_pipeline

# K-Nearest Neighbors Regression

knn\_pipeline = Pipeline(steps=[

('preprocessor', preprocessor),

('model', KNeighborsRegressor(n\_neighbors=5, leaf\_size=50))

])

# Function to evaluate models

def evaluate\_model(model, X\_valid, y\_valid):

predictions = model.predict(X\_valid)

mae = mean\_absolute\_error(y\_valid, predictions)

mse = mean\_squared\_error(y\_valid, predictions)

rmse = mean\_squared\_error(y\_valid, predictions, squared=False)

r2 = r2\_score(y\_valid, predictions)

print(f"MAE: {mae}")

print(f"MSE: {mse}")

print(f"RMSE: {rmse}")

print(f"R^2: {r2}")

def plot\_predictions(model, X\_valid, y\_valid, model\_name):

predictions = model.predict(X\_valid)

plt.figure(figsize=(10, 6))

sns.scatterplot(x=y\_valid, y=predictions)

plt.title(f'{model\_name} Predictions vs. Actual')

plt.xlabel('Actual Length of Stay')

plt.ylabel('Predicted Length of Stay')

plt.plot([y\_valid.min(), y\_valid.max()], [y\_valid.min(), y\_valid.max()], color='red', lw=2) # Line for perfect predictions

plt.show()

def plot\_residuals(model, X\_valid, y\_valid, model\_name):

predictions = model.predict(X\_valid)

residuals = y\_valid - predictions

plt.figure(figsize=(10, 6))

sns.scatterplot(x=predictions, y=residuals)

plt.title(f'{model\_name} Residuals')

plt.xlabel('Predicted Length of Stay')

plt.ylabel('Residuals')

plt.axhline(0, color='red', lw=2) # Line at 0 for no residuals

plt.show()

# Train and evaluate Linear Regression

linear\_regression\_pipeline.fit(X\_train, y\_train)

print("Linear Regression Performance:")

evaluate\_model(linear\_regression\_pipeline, X\_valid, y\_valid)

# Plot for each model

plot\_predictions(linear\_regression\_pipeline, X\_valid, y\_valid, 'Linear Regression')

plot\_residuals(linear\_regression\_pipeline, X\_valid, y\_valid, 'Linear Regression')

# Train and evaluate Random Forest Regression

random\_forest\_pipeline.fit(X\_train, y\_train)

print("\nRandom Forest Regression Performance:")

evaluate\_model(random\_forest\_pipeline, X\_valid, y\_valid)

plot\_predictions(random\_forest\_pipeline, X\_valid, y\_valid, 'Random Forest')

plot\_residuals(random\_forest\_pipeline, X\_valid, y\_valid, 'Random Forest')

# Train and evaluate KNN Regression

knn\_pipeline.fit(X\_train, y\_train)

print("\nKNN Regression Performance:")

evaluate\_model(knn\_pipeline, X\_valid, y\_valid)

plot\_predictions(knn\_pipeline, X\_valid, y\_valid, 'KNN')

plot\_residuals(knn\_pipeline, X\_valid, y\_valid, 'KNN')