**Advanced Predictive Analytics in Healthcare Using Big Data**

A black background with blue letters

Description automatically generated

**Team Members: (GROUP - 2)**

Supreeth Mohan (Roll No. 36)

Kruthika Reddy Murthy (Roll No. 37)

Trina Xavier (Roll No. 66)

**Abstract:**

The objective of this project was to predict healthcare billing amounts using advanced analytics on a large dataset.

We used a cleaned dataset, which was exported from Amazon Redshift after data processing and feature engineering for further analysis and model development in jupyter lab.

Key features included in the analysis were age, length of stay, and medical condition, among others.  
  
  
**Dataset:**  
The dataset healthcare\_dataset.csv contains healthcare data, including patient demographics, medical history, treatments, and outcomes. This data will be used to develop predictive models.

Dataset taken from Kaggle which has name , age, gender, blood type, medical condition, date of admission, doctor, hospital, insurance provider, billing amount, room number, admission type.

* 55,000 entries with 16 columns
* Key features: age, length of stay, medical condition, admission type, etc.
* Target variable: billing amount.

**Objectives:**

**Data Ingestion:** Retrieve data from source, such as csv and store it in S3.

**Data Transformation:** Implement Amazon Redshift to preprocess and transform health data. This includes data cleaning, normalization, and enrichment with patient-specific information.

**Analytics:** Amazon SageMaker, using it we performed feature engineering to create a new feature ‘Length of Stay’. These analytics will provide immediate insights into readmission rates.

* **Linear Regression:** Served as the baseline model, evaluated with metrics such as Mean Squared Error (MSE) and R-squared.
* **Polynomial Regression:** Enhanced the model by including polynomial features to capture non-linear relationships.
* **Ridge Regression:** Applied regularization to improve the model’s performance.Key evaluation metrics include MSE, R-squared, Mean Absolute Percentage Error (MAPE), and prediction accuracy.

**Data Storage:** Store the processed health data in Amazon S3, organized by hospital and timestamp, and catalog it, using AWS Redshift Data Catalog for easy access and discovery.

**Data Visualization:** Utilize a data visualization tool, Amazon Redshift to view the graphs for the health data. Further developed 3 machine learning models using Amazon Sage Maker, and plotted graphs and correlation matrix to visualize insight.

**Metrics:** By calculating R square, MSE and Accuracy, we opt for the right model in the prediction of patient readmission rates.

**Data Pipeline:**

A diagram of process processing

Description automatically generated

**AWS S3 and Redshift Setup and Process:**

**Step 1:** Create bucket healthdataproject and Dataset upload in s3

A screenshot of a computer

Description automatically generated

**Step 2:** Upload the dataset obtained from Kaggle website onto AWS.

A screenshot of a computer

Description automatically generated

**Step 3:** Created an IAM role with name project-role with administrative access so that it can be used in all applications of AWS.

IAM ROLE – MYREDSHIFTROLE

ARN: arn:aws:iam::554894760327:role/myRedshiftRole

We use this IAM role ARN in copying data from S3 to redshift table

**Step 4:** In Redshift we create cluster by providing the following details.

Name – healthdata

Nodetype- dc2.large

User-awsuser

Pw-Awsuser123

Choose my redshiftrole

And create cluster.

A screenshot of a computer

Description automatically generated

**Step 5:** Query in query editor

A screenshot of a computer

Description automatically generated

In redshift we run some queries, and check if the count matches with the excel, and find the null count.

**Step 6:** Next in Redshift, click query on query editor and create a connection to the database so that we can query from the storage part, that is, S3 by providing Database name as – dev.

A screenshot of a computer

Description automatically generated

**Step 7: Ater connection is successful, run query to create table.** Before loading data, we need to create a table that matches the structure of your CSV file.

A screenshot of a computer

Description automatically generated

CREATE TABLE healthcare\_data (

Name VARCHAR(100),

Age INT,

Gender VARCHAR(10),

Blood\_Type VARCHAR(10),

Medical\_Condition VARCHAR(100),

Date\_of\_Admission DATE,

Doctor VARCHAR(100),

Hospital VARCHAR(100),

Insurance\_Provider VARCHAR(100),

Billing\_Amount DECIMAL(10, 2),

Room\_Number VARCHAR(10),

Admission\_Type VARCHAR(50),

Discharge\_Date DATE,

Medication VARCHAR(200),

Test\_Results VARCHAR(200)

);

**Step 8:** Use the COPY Command to Load Data as given below.

A screenshot of a computer

Description automatically generated

COPY healthcare\_data

FROM 's3://healthdataproject/healthcare\_dataset.csv'

IAM\_ROLE 'arn:aws:iam::554894760327:role/myRedshiftRole'

CSV

DELIMITER ','

IGNOREHEADER 1;

**Step 9: Verify the Data** by running the below query and confirm if the data is present.

SELECT \* FROM healthcare\_data LIMIT 10;

**Step 10:** **Data cleaning,** check whether the data presenent is null, if yes, then set to 0, for all the three columns.

A screenshot of a computer

Description automatically generated

UPDATE healthcare\_data

SET age = 0

WHERE age IS NULL;

UPDATE healthcare\_data

SET billing\_amount = 0

WHERE billing\_amount IS NULL;

UPDATE healthcare\_data SET date\_of\_admission = TO\_DATE(date\_of\_admission, 'YYYY-MM-DD');

**Verifying if all the null values have been set to 0.**

SELECT COUNT(\*) FROM healthcare\_data WHERE age = 0;

A screenshot of a computer

Description automatically generated

**Step 11: Feature engineering,** where we added a column ‘Length of Stay’ to predict the readmission rates easily.

-- Add the length\_of\_stay column

ALTER TABLE healthcare\_data

ADD COLUMN length\_of\_stay INT;

-- Update the length\_of\_stay with the difference between discharge\_date and date\_of\_admission

UPDATE healthcare\_data

SET length\_of\_stay = CASE

WHEN DATEDIFF(day, date\_of\_admission::date, discharge\_date::date) < 0 THEN NULL

ELSE DATEDIFF(day, date\_of\_admission::date, discharge\_date::date)

END;

Length of stay feature is created

A screenshot of a computer

Description automatically generated

**Step 12:** Exported it as csv and named as cleaned\_data and used for deployment.

A screenshot of a computer

Description automatically generated

**AWS Sage Maker Model Deployment:**

**Step 1:** Import the required libraries which are important for the execution.

**A screenshot of a computer program

Description automatically generated**

**Step 2: Load cleaned data and perform preprocessing for categorical and numerical features.**

**A screenshot of a computer

Description automatically generated**

**Step 3: Pipeline creation and fit the model**

**A screenshot of a computer code

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

Where we convert the numerical attributes to categorical and perform the model fitment. Also, split the data into 80:20 ration, where 80% is for training the model and remaining 20% is for testing the model, whether if it predicts correctly or not.

**Linear Regression Model:**

**A screenshot of a computer program

Description automatically generated**

Using linear regression, we obtained a Mean Squared Error (MSE) value of 245170862.10 and an R-squared value of -0.13. The high MSE indicates that the model's predictions are significantly deviating from the actual values, suggesting poor predictive accuracy. An R-squared value of -0.13, which is negative, is particularly concerning because it indicates that the model is performing worse than a simple horizontal line representing the mean of the dependent variable. In other words, the model fails to capture any meaningful linear relationship between the features and the target variable. This could be due to several reasons, such as the presence of outliers, multicollinearity among the independent variables, or a nonlinear relationship that linear regression cannot capture. To improve model performance, it may be necessary to explore different modeling techniques, conduct feature engineering, or transform the data to better fit the assumptions of linear regression.

**Polynomial Regression Model:**

**A screenshot of a computer program

Description automatically generated**

Using polynomial regression, we obtained a Mean Squared Error (MSE) value of 205,284,125.24 and an R-squared value of 0.048. The reduction in MSE compared to the linear regression model suggests that the polynomial model better captures the variability in the data, although the predictive accuracy is still relatively poor. The R-squared value of 0.048, while an improvement over the negative value obtained with linear regression, indicates that the model only explains a small portion of the variance in the dependent variable. This suggests that the polynomial model captures some of the complexity in the data but is still insufficient in representing the underlying relationships fully. The modest improvement in R-squared could be due to a slightly more complex model fitting some data points better, but it also indicates that other factors or higher-order terms might be needed to enhance the model further. Additional data preprocessing, feature engineering, or experimentation with different polynomial degrees might be necessary to achieve better model performance.

**Ridge Polynomial Regression Model:**

**A screenshot of a computer code

Description automatically generated**

Using ridge polynomial regression, we obtained a Mean Squared Error (MSE) value of 206,154,573.68 and an R-squared value of 0.044. Compared to the basic polynomial regression model, the slightly higher MSE and lower R-squared indicate that the ridge regularization did not significantly improve the model's predictive accuracy. The purpose of ridge regression is to prevent overfitting by penalizing large coefficients, which is particularly useful in polynomial regression where the risk of overfitting is higher. However, the similar performance metrics suggest that the original polynomial model was not overfitting to a great extent, or that the regularization parameter may need tuning. The model still explains a minimal amount of the variance in the dependent variable, indicating that there might be other factors at play or that more complex modeling techniques could be explored. Further investigation, such as adjusting the regularization strength or trying other forms of regularization, might be needed to enhance the model's performance.

A collage of blue and white squares

Description automatically generated

By plotting the above graph, we get to know that,

* Billing amount seems to have a wide range, with most values concentrated in the lower range but same high outliers.
* Age appears to be distributed across a wide range roughly from 20-80 years.

A screenshot of a computer

Description automatically generated

By interpreting the above correlation matrix, we get to know that.

* Age and Length of stay is 0.1, which is weak positive correlation, means the older patients might stay slightly longer.
* Age and billing amount has negative correlation, which means there is no correlation between them.
* Billing and length of stay is also the same, where there is no correlation.

**Model Performance Summary**

**Comparison of Linear, Polynomial, and Ridge Polynomial Regression Models**

1. **Linear Regression**
   * **MSE**: 245,170,862.10
   * **R-squared**: -0.13
   * **Interpretation**: The model shows high prediction error and a negative R-squared, indicating it performs worse than the mean model. Linear regression fails to capture the relationship between features and the target variable.
2. **Polynomial Regression**
   * **MSE**: 205,284,125.24
   * **R-squared**: 0.048
   * **Interpretation**: The model has a lower MSE compared to the linear model, indicating improved prediction accuracy. The R-squared value is slightly positive, suggesting the model explains some variance in the data, though it is still not very effective.
3. **Ridge Polynomial Regression**
   * **MSE**: 206,154,573.68
   * **R-squared**: 0.044
   * **Interpretation**: This model has a slightly higher MSE and a marginally lower R-squared value than the polynomial regression. Ridge regularization does not significantly improve the model’s performance, suggesting that overfitting was not a major issue in the polynomial model.

**Best Model Summary**

Based on the comparison, the **polynomial regression** model performs best among the three models, with the lowest MSE and a slightly higher R-squared value. While the improvements are modest, this model captures more variability in the data compared to the linear and ridge polynomial regression models. However, its performance indicates a need for further model refinement, possibly by experimenting with different polynomial degrees, other regularization techniques, or incorporating additional features to enhance predictive accuracy.

**Future Enhancement:**

To further refine the model, we can consider the following:

1. Hyperparameter Tuning: Adjust the regularization strength (alpha) in the Ridge regression model to see if it improves the results.
2. Polynomial Degree: Experiment with different degrees of polynomial features to capture more complex relationships in the data.
3. Feature Engineering: Look for additional features or transformations that could improve model performance.
4. Cross-validation: Use cross-validation techniques to ensure that your model's performance is consistent across different subsets of the data.
5. Alternative Models: Try other regression algorithms (e.g., Lasso regression, Elastic Net, or different types of ensemble methods) to see if they provide better accuracy or lower error metrics.

By iterating on these steps, we can aim to improve both the MAPE and the "accuracy" metric for Ridge Polynomial Regression model.

**References:**

* <https://quicksight.aws.amazon.com/>
* https://aws.amazon.com/