General Electric Healthcare

**REPORT**

Zora-Ann Webley-Johnson

Southern New Hampshire University

DAT-690 Capstone in Data Analytics

Professor: Daniel Letort

Email: d.letort@snhu.edu

Module Nine

February 18, 2024

**Chapter 1 Introduction**

In this project, we embark on a data mining endeavor to address a critical business challenge faced by General Electric Healthcare (GE Healthcare) portfolio operating in the Healthcare IT Services domain. By employing the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology, we aim to systematically analyze the available data to derive actionable insights that can inform decision-making and drive business growth.

* 1. ***Business Understanding***

**Business Objectives:**

In response to increased competition and a slight decline in customer retention related to its cellular service, GE Healthcare is striving to preserve its health IT service customer base. Previously, the company has utilized an Oracle database and adaptable data warehouse infrastructure to identify potential churn markers and pinpoint customers at risk of imminent churn. Presently, GE aims to establish a predictive framework for estimating customer revenue based on usage metrics. The primary objective of this data mining initiative is to utilize available data resources to uncover insights into projected revenue gains from individual customers. By doing so, the organization can determine whether employing marketing strategies for specific customers will result in financial gain or loss.

The telehealth industry, which includes virtual healthcare services, has seen significant growth. In 2022, it reached USD 29.6 billion in the U.S. market and is projected to grow at a rate of 23.4% annually from 2024 to 2030 as seen in figure 1 below. The growth is fueled by a shortage of healthcare specialists, with 76% of U.S. hospitals already using telemedicine. Investments by the Biden-Harris Administration aim to improve access, especially in rural areas. Globally, the telehealth market hit USD 101.2 billion in 2023 and is expected to grow at a rate of 24.3% annually from 2024 to 2030. Despite challenges with technology and integration, telehealth plays a crucial role in the digital health revolution by improving access, outcomes, and cost-efficiency (*Telehealth Market Size, Share & Trends Analysis Report by Product Type (Hardware, Software, Services), by Delivery Mode (On-Premise, Web-based), by End-use, by Disease Area, by Region, and Segment Forecasts, 2024 - 2030*, 2023).

*Figure 1: Shows projected revenue growth for US Telehealth Market*

A graph of sales growth

Description automatically generated with medium confidence

Image citation: *Telehealth Market Size, Share & Trends Analysis Report by product type (Hardware, software, services), by delivery mode (On-Premise, web-based), by end-use, by disease area, by region, and segment Forecasts, 2024 - 2030*. (2023, December 1). https://www.grandviewresearch.com/industry-analysis/telehealth-market-report

General Electric (GE) can capitalize on the booming telehealth market by investing in product development, forging strategic partnerships, and advancing research initiatives. By offering innovative telehealth solutions and collaborating with industry leaders, GE can expand its presence in the digital healthcare space. Given the rapid market growth projections, there is ample room for GE's growth. It is of paramount importance that the company addresses its data wrangling procedures to ensure that churn markers and revenue predictors are identified correctly. The integrity and dependability of the data will ultimately be the determining factor for its success in the Telehealth Industry.

* 1. ***Data Understanding***

The Revenue Churn Dataset consists of 800 rows and 38 variables, including character, integer, or numerical data types. Initial exploratory analysis revealed missing values in six variables—DIRECTAS, ROAM, CHANGER, MOU, OVERAGE, and CHANGEM. Variables like DROPVCE, BLCKVCE, UNANSVCE, CUSTCARE, THREEWAY, OUTCALLS, INCALLS, PEAKVCE, OPEAKVCE, DROPBLK, CALLFWDV, and CALLWAIT, which represent whole features, were converted from decimals to whole numbers to have a more realistic representation of the variable. Furthermore, examination of variable distributions through box plots uncovered prevalent outliers in over half of the variables of the dataset. A closer look at quantiles in these box plots suggests the presence of skewness in a predominant portion of the data set. Scatterplots indicate potential correlations between certain variables, adding depth to our understanding.

These exploratory measures guide further exploration of the dataset. Recognizing the iterative nature of the CRISP-DM methodology, we acknowledge the importance of refining our methods. Throughout the model synthesis process, data cleaning was conducted using various combinations. One model iteration included outliers, while another iteration removed them. The model without outliers was selected because it performed better in predicting revenue for unseen data (test data). The test dataset underwent the same data cleaning methodologies to ensure consistency in processing data, to increase realistic representations avoiding bias, and to improve generalization. As new insights regarding data quality and validity emerge, we are committed to enhancing our approach to fully comprehend the intricacies of the dataset.

**Chapter 2 Plan Definition**

***2.1 Analytic Architecture Pattern***

A data analytic architecture serves as the bedrock of organizational endeavors, offering a meticulous framework of policies and standards to steer the construction of analytical processes (Data Analytics Architecture: Data Lake, Big Data, Use Cases | Definition, 2023). Beyond mere technical specifications or capabilities, this architectural paradigm intricately interweaves analytics with a company's overarching business strategy (Data Analytics Architecture: Data Lake, Big Data, Use Cases | Definition, 2023). It serves as a linchpin, harmonizing the intricate dance between data insights and strategic imperatives, thereby fortifying the organization's capacity for informed decision-making and sustained growth. Figure two below shows how General Electric plans to embark on deriving insights from its data. However, the company must also implement measures to ensure stakeholders’ influence can be felt throughout any analytic plan.

*Figure 2 shows General Electric Analytic Architecture Plan*

**A diagram of a company

Description automatically generated**

**2.1.1 Analytic Architecture Plan Description:**

Data Sources

* Oracle Database: Contains customer information updated monthly, including account-level transactional data.
* Customer Churn Data (CSV file): Compiled for the pilot project, includes information on GE health IT application customers. These files can be stored also in any data processing tool like Python's pandas library as they handle CSV files effectively.

Data Warehouse:

* Architecture: A centralized data warehouse that integrates data from the Oracle database and the Customer Churn Data file.
* Update Frequency: Updated monthly at the end of each month by appending the most recent end-of-month data.
* Purpose: Provides a single source of truth for analysis and reporting purposes.
* Technology: a data warehouse solution such as Amazon Redshift, Snowflake, or Google BigQuery can be implemented to centralize and manage integrated data.
* ETL Tools: Use tools like Apache NiFi, Talend, or Informatica for Extract, Transform, Load (ETL) processes.

3. Data Preparation:

* + ETL Processes: Implement Extract, Transform, Load (ETL) processes to clean, transform, and integrate data from disparate sources into the data warehouse.
  + Data Quality Checks: Conduct data quality checks to ensure accuracy, completeness, and consistency of the integrated data.
  + Python and pandas: Use Python programming language along with the pandas library for additional custom data cleaning, transformation, and integration of data set.
  + Data Quality Tools: Implement data quality tools like Trifacta (*Validate Your Data*, n.d.) that has built in data validation methods.

4. Data Analysis:

* + Use data analysis tools such as Python, R, or SQL to perform exploratory data analysis (EDA) on the integrated dataset.
  + Identify potential churn indicators and drivers by analyzing metadata.
  + Identify drivers for revenue gain.

5. Predictive Analytics Model Development:

* + Develop a predictive analytics model to identify customers at risk of churn.
  + Utilize machine learning algorithms such as logistic regression, decision trees, or random forests to predict churn probabilities based on historical data.
  + Utilize machine learning algorithms such as linear regression, lasso regression etc to predict customer revenue.

6. Data Visualization and Reporting:

* + Use data visualization tools such as Tableau, Power BI, or Matplotlib to create visualizations and dashboards.
  + Visualize churn trends, subscriber behavior patterns, and predictive model results for stakeholders.

7. Stakeholder Collaboration:

* Engage essential stakeholders, including the customer account management team, IT department, and management, in both data analysis and decision-making.
* Consistently convey discoveries, insights, and recommendations to stakeholders through meetings, presentations, and reports. Utilize Project Management Software platforms like Trello or Monday.com to facilitate communication and collaboration. However for this project stakeholders can keep abreast of project progress using the project plan outlined in section 2.3.

8. Governance and Security:

* Encryption Technologies: Implementing encryption technologies like AES (Advanced Encryption Standard) ensures the security of sensitive data (Dray, 2021) and is government approved. More specifically HIPAA compliance requirements are for minimum AES 128-bit encryption of patient data (Alder, 2024).
* Access Control Mechanisms: Utilizing role-based access control (RBAC) and authentication mechanisms to enable precise management of user access to data, a key aspect of maintaining HIPAA compliance (Dray, 2021).
* Auditing and Monitoring Tools: Deploying auditing and monitoring tools facilitates the tracking and analysis of data access and usage, essential for meeting HIPAA's security rule requirements (Rights, 2022).

***2.2 Stakeholder Presentations***

The presentation attached targets the Finance and IT departments/stakeholders. It explains to the Finance department why retaining customers is the more cost-efficient approach while acquiring new customers would be more expensive. The IT department was introduced to the revenue predictive model and why it needs to be maintained.

***2.3 Project Plan***

To ensure project success for General Electric Healthcare, professional and effective collaboration will be paramount across all stages of the project lifecycle. By fostering an environment of open communication and collaboration, team members from various departments and disciplines, including research and development, engineering, marketing, and customer service, will be encouraged to share insights, expertise, and perspectives. This is enforced through regular meetings, workshops, and brainstorming sessions to provide opportunities for stakeholders to align on project objectives, priorities, and timelines. For this project communication of project status can be monitored through the project management plan outlined in figure 3 below. However, it is advised that GE Healthcare leverages modern collaboration tools and platforms, such as project management software and communication channels, to facilitate seamless coordination and information sharing among team members, regardless of their geographic location. The project management tool in figure 3 makes up for the lack of advanced project management software by establishing clear roles, responsibilities, and accountability frameworks. The plan ensures that each team member understands their contributions to the project and remains committed to achieving shared goals. By embracing a culture of professional collaboration and teamwork, General Electric Healthcare will optimize its resources, minimize risks, and drive innovation to deliver successful outcomes that meet the needs of its customers and stakeholders.

*Figure 3 Shows General Electric Healthcare Project Plan.*



**Chapter 3 Plan Implementation**

According to the documentation provided by IBM, deployment involves leveraging newfound insights to effect improvements within an organization (IBM, n.d.). This could entail a structured integration, such as implementing an IBM® SPSS® Modeler model to generate churn scores, which are then incorporated into a data warehouse. Alternatively, deployment might involve leveraging data mining insights to instigate organizational change. The deployment stages of GE Revenue Predictive Model will entail the following: Model Maintenance Protocols, Model Launch, Final Report and Presentation, and Routine Success Measurements.

***3.1 Deployment Plan***

Machine learning model deployment can be complex, and organizations often need to adjust their DevOps processes to accommodate it (Seldon, 2023). Typically, deployment involves four stages: model development, testing and code refinement, container preparation, and post-deployment monitoring and maintenance (Seldon, 2023).

Using containers is a popular approach for deploying machine learning models. Containers provide an isolated environment that simplifies maintenance, updates, and scaling (Seldon, 2023). They encapsulate all the necessary elements for the execution of machine learning code and ensure consistent performance (Seldon, 2023). Container orchestration platforms like Kubernetes automate important aspects of container management, such as monitoring and scaling, further enhancing efficiency (Seldon, 2023).

*Figure 4 Shows General Electric Healthcare Deployment Plan.*



**Chapter 4 Results**

***4.1 Model Performance***

Model performance refers to the ability of a predictive model to make accurate predictions or classifications based on the input data. It is a measure of how well the model generalizes to new, unseen data and how effectively it achieves its intended purpose. Model performance is typically evaluated using various metrics and techniques, depending on the nature of the problem and the type of model being used. For regression models, common performance metrics include mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination (R-squared). These metrics measure the accuracy and precision of the model's predictions compared to the actual values in the dataset. Though these metrics are excellent indicators of a model’s performance, they do not account for bias within the model (Yashwanth, 2021). As such, the Revenue predictor linear regression model created for GE Healthcare performance was first assessed by reviewing its residual plots. A residual plot is a scatter plot of residuals vs predicted values. A model can be deemed unbiased if its residuals are evenly distributed about the zero line of a residual plot. This indicates that there is no pattern in then we can assume that the model generalizes well and is unbiased. Figure 5 below shows that the residuals of the selected model were randomly and evenly distributed across the zero line; indicating that the model is unbiased.

*Figure 5 shows the residual plot for the revenue prediction linear regression model.*

*A graph with numbers and dots

Description automatically generated*

The model also performed well before and after feature engineering, producing the below MSE, RMSE, MAE and R squared values. Linear regression evaluation metrics offer insights into the model's fitting and predictive capabilities. Mean Squared Error (MSE) computes the average of squared differences between actual and predicted values, giving more weight to large errors, with lower MSE suggesting superior performance. Root Mean Squared Error (RMSE), the square root of MSE, expresses errors in the same units as the target variable, indicating the average magnitude of predictions' errors. Similarly, Mean Absolute Error (MAE) averages absolute differences between actual and predicted values, providing a measure of error magnitude less sensitive to outliers than MSE, with lower MAE values indicating better performance. Coefficient of Determination (R-squared) measures the proportion of variance in the dependent variable explained by independent variables, ranging from 0 to 1, where higher values signify better model fit. Collectively, these metrics gauge the accuracy, precision, and explanatory power of the linear regression model, informing its overall effectiveness and reliability in data analysis and prediction. Each of the metrics improved after feature engineering, but also showed that the base model (model before feature engineering) performed relatively well. The RMSE is a strong indication of how well the model fit. This value indicates that the expected error for predicting a customers revenue is under $50.

Top of Form

*Figure 6 shows model performance (a) before and (b) after feature engineering.*

*(a)*

*A computer error message

Description automatically generated*

*(b)*

*A computer error message

Description automatically generated*

***4.2 Feature Importance Analysis and Cross Validation***

Several techniques were used to derive the features of importance. These were namely: Principal Component Analysis, K-fold Cross Validation, and the Regression Coefficients. Variable importance is defined as a measure of each regressor's (or predictor variable) contribution to model fit. (Coleman, 2022). It in essence, variable importance helps analysts understand the key drivers of the target variable. In this research, it helps understand what variables better explain how much revenue a customer is likely to contribute to the business. The variables identified as important to the linear regression model were namely: DataUsageGB, EQPDAYS, CompetitivePackage, SETPRC, MONTHS, and CHANGEM. The reduced dimension was crafted with the insights from all three of the variable importance techniques: by comparing variables that were common among the results of the different techniques.

**4.2.1 Principal Component Analysis**

Principal Component Analysis (PCA) is a dimensionality reduction technique used to simplify complex datasets by transforming variables into a new set of uncorrelated variables called principal components. PCA indirectly identifies variables of importance by revealing patterns and relationships within the data. Principal components (PCs) are new variables created by PCA, capturing the maximum variance in the data (*11.1 - Principal Component Analysis (PCA) Procedure | STAT 505*, n.d.). The variables were deemed most important if they contributed significantly to the PC’s that explained the greatest variance in the dataset. Figure 7 below shows PC loadings where higher loading values are indicative of the variables contribution to the PC itself. From that diagram PC1 explained most of the variance in the dataset and so the loadings are sorted by PC1.

*Figure 7 showing PC loadings and variables that explain most of the variance in the data set.*

A screenshot of a computer

Description automatically generated

**4.2.2 K-Fold Cross Validation**

K-fold cross-validation is a widely used resampling technique that partitions a dataset into K equal-sized subsets or folds. It trains the model K times, using K-1 folds for training and the remaining fold for validation, assessing the model's performance by averaging metrics across all iterations. This method offers a robust evaluation of model performance and helps gauge its generalizability. During each iteration, variables' importance can be analyzed by observing their consistent impact on model performance across different folds. If a variable consistently enhances model performance across multiple folds, it suggests its importance in predicting the outcome. K-fold cross-validation thus provides a reliable and unbiased assessment of variable importance, enhancing the overall reliability and generalizability of predictive models. For this research, the K-fold technique was conducted in RStudio. After the regression model was developed using the K-fold technique the VarImp function was used to list the variables that were of must importance. See figure 8 below for the results from the K-fold VarImp function:

*Figure 8 shows the 20 most important variables identified from a k-foldc cross validation.*

*A screenshot of a computer program

Description automatically generated*

**4.2.3 Regression Coefficients**

The coefficient of a regression model determines variable importance by reflecting the strength and direction of the relationship between predictor and outcome variables. The magnitude of the coefficient indicates the change in the outcome for a one-unit change in the predictor, while the sign reveals the direction of the relationship. Statistical significance, assessed through tests, highlights the reliability of the relationship. Comparative analysis of coefficients aids in identifying variables with greater impact. Overall, larger, statistically significant coefficients signify variables of greater importance, crucial for understanding predictive power. In figure 9 below the variables with the larger coefficients have been identified with asterisks. Larger coefficients receive more asterisks.

*Figure 9 showing coefficients magnitude and statistical significance*

*A screenshot of a computer

Description automatically generated*

***4.3 Model Assumptions and Limitations***

Linear regression models, though prevalent and invaluable in numerous scenarios, possess several limitations. Firstly, they rely on the assumption of linearity, presupposing a linear relationship between predictor variables and the outcome variable. When the relationship deviates from linearity, the model may yield inaccurate results (*Linear Regression: Assumptions and Limitations*, 2024). Additionally, linear regression assumes independence among predictor variables, but multicollinearity can lead to unstable coefficient estimates and interpretation challenges (*Linear Regression: Assumptions and Limitations*, 2024). Another crucial assumption is homoscedasticity, assuming constant variance of residuals across predictor variable levels; heteroscedasticity can result in biased standard errors and unreliable hypothesis tests.

As delineated in the preceding passage, the limitations of linear regression are closely linked to the assumptions underlying the model. Hence, it is imperative to evaluate these assumptions within the dataset to ascertain the model's validity. Through scrutinizing these model features, analysts can effectuate the requisite alterations to the data, thereby ensuring the synthesis of an accurate and dependable model. The Revenue Prediction Linear Regression model underwent an examination for all specified assumptions. Figures 10 – 13 illustrate that the data and the model itself satisfied all assumptions.

*Figure 10 shows variables within the dataset that were highly correlated.*

*A blue squares with white text

Description automatically generated*

After identifying variables of high correlations, the data set was reduced by the variables that contributed least to the linear relationship.

*Figure 11 shows model residuals are normally distributed.*

*A graph of a graph

Description automatically generated*

When the residuals are normally distributed, it suggests that the linear regression model is capturing the underlying relationships in the data effectively. It implies that the model's assumptions are met, and the errors are random and not biased in any direction.

*Figure 12 shows Breusch-Pagan test for heteroscedasticity in the model.*

*A black text on a white background

Description automatically generated*

The Breusch-Pagan test is employed to detect heteroscedasticity in regression models. The test operates by comparing the null hypothesis, which assumes homoscedasticity (constant variance of residuals), against the alternative hypothesis, indicating the presence of heteroscedasticity. The test statistic follows a chi-square distribution and is computed from the residuals. When the p-value associated with the test statistic falls below a chosen significance level (0.05), the null hypothesis is rejected, signifying the presence of heteroscedasticity. Conversely, if the p-value exceeds the significance level, the null hypothesis is retained, suggesting no substantial evidence of heteroscedasticity in the regression model.

The studentized Breusch-Pagan test yielded a test statistic of 43.039 with 31 degrees of freedom and a p-value of 0.07369. Since the p-value surpasses the chosen significance level (e.g., 0.05), we cannot reject the null hypothesis of homoscedasticity. Consequently, there is no significant evidence of heteroscedasticity in the regression model, suggesting that the variance of the residuals likely remains constant across all levels of the independent variables. This result enhances confidence in the validity of the regression analysis as the model's assumptions are met.

**Chapter 5 Conclusion and Implications**

The Revenue Prediction Linear Regression Model was implemented to predict revenue and meet the organization's objectives effectively. This predictive capability enables the organization to make informed decisions regarding resource allocation, marketing strategies, and revenue forecasting with greater accuracy. Moreover, the model helped to confirm several pivotal suspicions regarding revenue dynamics within the organization. It provided valuable insights into market trends and competitive pressures and highlighted the importance of exploring variables such as setprc, equipment days, and months in greater detail to elucidate their influence on revenue generation. The outcomes of the project also shed light on the potential benefits and costs associated with data analytics initiatives. The benefits include improved decision-making, enhanced operational efficiency, and better resource utilization, which can ultimately lead to increased revenue and competitiveness in the industry. The successful implementation of the Revenue Prediction Linear Regression Model sets a precedent for future projects within the organization and the broader industry. It underscores the importance of leveraging data analytics techniques to derive actionable insights and drive business success.

In conclusion, the implementation of the Revenue Prediction Linear Regression Model represents a significant milestone for the organization, demonstrating the transformative power of data analytics in driving business outcomes. The organization is now poised to make more informed decisions, optimize resource allocation, and capitalize on emerging opportunities in the dynamic market landscape of Telehealth.

**Chapter 6 Recommendations**

Several recommendations could enhance the predictive power and practical utility of the Revenue Prediction Linear Regression Model. The first recommendation is to explore additional data sources that could provide deeper insights into revenue drivers. Specifically, including market trends and macroeconomic indicators can assess whether external factors outside competitors affect revenue. However, integrating external data sources may require data preprocessing and alignment with existing datasets to ensure compatibility and consistency. Investing or creating software that automates Extract Load and Transform is of great importance to the speed at which analyses are conducted.

Secondly, further consideration should be given to implementing advanced machine learning techniques such as ensemble or neural networks to improve revenue prediction accuracy. These techniques can capture complex nonlinear relationships and interactions among variables that linear regression may not adequately capture. However, adopting advanced techniques may necessitate additional computational resources and expertise in model development and validation.

Thirdly, dynamic modeling approaches that account for temporal dependencies and seasonality in revenue patterns could be investigated. Time series analysis techniques such as ARIMA or Prophet models could be explored to capture underlying trends and seasonality effects, thereby enhancing the model's predictive capabilities. However, dynamic modeling approaches may require data preprocessing and model architecture adjustments to accommodate time-dependent features and trends.

Lastly, implementing a continuous model monitoring and updating system is essential for maintaining predictive accuracy and relevance over time. Regular evaluation of model performance against new data and evolving business dynamics is necessary. Additionally, periodic retraining of the model with fresh data and recalibration of model parameters may be required to adapt to changing market conditions and business priorities.

These recommendations aim to enhance the predictive capabilities and practical utility of the Revenue Prediction Model, enabling organizations to make more informed decisions and optimize revenue generation strategies effectively in dynamic business environments.

**References**

*11.1 - Principal Component Analysis (PCA) Procedure | STAT 505*. (n.d.). PennState: Statistics Online Courses. https://online.stat.psu.edu/stat505/lesson/11/11.1

Alder, S. (2024, February 13). *HIPAA encryption Requirements*. HIPAA Journal. https://www.hipaajournal.com/hipaa-encryption-requirements/

Coleman, C. D. (2022). The importance of variable importance. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2212.03289

*Data analytics architecture: Data lake, big data, use cases | Definition*. (2023, December 11). Starburst. <https://www.starburst.io/learn/data-fundamentals/data-analytics-architecture/#:~:text=A%20data%20analytics%20architecture%20is%20a%20set%20of,architecture%20links%20analytics%20to%20a%20company%E2%80%99s%20business%20strategy>

Dray, M. J. D. E. B. B. J. R. N. J. F. L. E. B. E. R. J. F., Jr. (2021, March 1). *Advanced Encryption Standard (AES) | NIST*. NIST. https://www.nist.gov/publications/advanced-encryption-standard-aes

IBM. (n.d.). IBM SPSS Modeler documentation (Version 18.2.2): Deployment overview. Retrieved from https://www.ibm.com/docs/en/spss-modeler/18.2.2?topic=deployment-overview

*Linear Regression: Assumptions and limitations*. (2024, January 15). Quantitative Finance & Algo Trading Blog by QuantInsti. https://blog.quantinsti.com/linear-regression-assumptions-limitations/

Rights, O. F. C. (2022, October 20). *Summary of the HIPAA Security Rule*. HHS.gov. <https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html>

*Telehealth Market Size, Share & Trends Analysis Report by product type (Hardware, software, services), by delivery mode (On-Premise, web-based), by end-use, by disease area, by region, and segment Forecasts, 2024 - 2030*. (2023, December 1). https://www.grandviewresearch.com/industry-analysis/telehealth-market-report

*Validate your data*. (n.d.). https://docs.trifacta.com/Dataprep/en/trifacta-application/common-tasks/validation-tasks/validate-your-data.html#transformations-vs--data-quality-rules

Yashwanth, N. (2021, December 16). Evaluation metrics & Model Selection in Linear Regression. *Medium*. <https://towardsdatascience.com/evaluation-metrics-model-selection-in-linear-regression-73c7573208be>

Appendix

# Load necessary libraries

install.packages("corrplot")

install.packages("VIM")

install.packages("car")

library(dplyr)

library(Metrics) # For evaluation metrics

library(caret)

library(corrplot)

library(VIM)

library(car)

library(lmtest)

# Load train and test datsets

data\_set1 <- read.csv("C:/Users/loraa/OneDrive/Family/Desktop/Desktop/SNHU/DAT-690 Capstone in Data Analytics/Final Project/CustomerRevenue\_Data.csv")

new\_data <- read.csv("C:/Users/loraa/OneDrive/Family/Desktop/Desktop/SNHU/DAT-690 Capstone in Data Analytics/Final Project/CustomerRevenue\_Verify.csv")

#View data set structure

glimpse(data\_set1)

glimpse(new\_data)

#Variables to be rounded

# Select variables to round up

variables\_to\_round <- c("DROPVCE", "BLCKVCE", "UNANSVCE", "CUSTCARE", "THREEWAY", "OUTCALLS", "INCALLS", "PEAKVCE", "OPEAKVCE", "DROPBLK", "CALLFWDV", "CALLWAIT")

# Round up selected variables

data\_set1[, variables\_to\_round] <- lapply(data\_set1[, variables\_to\_round], function(x) ceiling(x))

new\_data[, variables\_to\_round] <- lapply(new\_data[, variables\_to\_round], function(x) ceiling(x))

# View the modified data\_set

glimpse(data\_set1)

glimpse(new\_data)

#remove object no longer needed

rm(variables\_to\_round)

#convert variables to numerical and remove categorical variables from both test and train dataset

data\_set1 <- data\_set1 %>% select(-CUSTOMER, -PhoneNumber) %>% mutate\_all(as.numeric)

new\_data <- new\_data %>% select(-CUSTOMER, -PhoneNumber) %>% mutate\_all(as.numeric)

#data\_set1 <- data\_set1[,3:39]

#new\_data <- new\_data[,3:38]

#data\_set1 <- mutate\_all(data\_set1, as.numeric)

#new\_data <- mutate\_all(new\_data, as.numeric)

# View the modified data\_set

glimpse(data\_set1)

glimpse(new\_data)

# Replace zeros with NA in SETPRC in both test and train dataset

data\_set1$SETPRC[data\_set1$SETPRC == 0] <- NA

new\_data$SETPRC[new\_data$SETPRC == 0] <- NA

#######################Missing Value Analysis

# Check for missing values using colSums

print(colSums(is.na(data\_set1)))

print(colSums(is.na(new\_data)))

#install.packages("VIM")

#library(VIM)

missing\_pattern <- aggr(data\_set1, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(data\_set1), cex.axis=.7, gap=3)

#what is the percentage of missing data in SETPRC for different values of CHANGER?

spineMiss(data\_set1[, c("CHANGER","SETPRC")])

#what is the percentage of missing data in SETPRC for different values of DIRECTAS?

spineMiss(data\_set1[, c("DIRECTAS", "SETPRC")])

spineMiss(data\_set1[, c("ROAM", "SETPRC")])

spineMiss(data\_set1[, c("MOU", "SETPRC")])

spineMiss(data\_set1[, c("OVERAGE", "SETPRC")])

spineMiss(data\_set1[, c("CHANGEM", "SETPRC")])

#Patterns in missingness

matrixplot(data\_set1, sortby = c('CHANGEM'))

matrixplot(data\_set1, sortby = c('CHANGER'))

matrixplot(data\_set1, sortby = c('MOU'))

matrixplot(data\_set1, sortby = c('DIRECTAS'))

matrixplot(data\_set1, sortby = c('OVERAGE'))

matrixplot(data\_set1, sortby = c('ROAM'))

matrixplot(data\_set1, sortby = c('SETPRC'))

matrixplot(data\_set1, sortby = c('REVENUE'))

matrixplot(data\_set1, sortby = c('DROPVCE'))

#impute missing values in the data\_set1 (train dataset) using K nearest neighbour.

# Impute missing values using k-nearest neighbors (KNN)

data\_set1 <- kNN(data\_set1, k = 5)

# Delete the first 36 columns

data\_set1 <- data\_set1[, -(38:74)]

#check that values were imputed

print(colSums(is.na(data\_set1)))

# Impute missing values in new\_data (test dataset) using k-nearest neighbors (KNN)

new\_data <- kNN(new\_data, k = 5)

# Delete the first 36 columns

new\_data <- new\_data[, -(37:72)]

#check that values were imputed

print(colSums(is.na(new\_data)))

##removing objects no longer needed

rm(missing\_pattern)

###Principal Component Analysis for dimension reduction

str(data\_set1)# check if all variables are numeric

mean(cor(data\_set1[,2:36])) #mean correlation of the dataset is less than 0.3 hence there are no strongly corelated variables and the PCA test can be avoided

pca\_result <- prcomp(data\_set1[,2:36])

summary(pca\_result)

loadings <- pca\_result$rotation

print(loadings)

PC = pca\_result$x

View(PC)

mean(cor(PC))

# Extract eigenvalues

eigenvalues <- (pca\_result$sdev)^2

# Plot the scree plot

plot(1:length(eigenvalues), eigenvalues, type = "b",

xlab = "Principal Component", ylab = "Eigenvalue",

main = "Scree Plot")

# Create a biplot

biplot(pca\_result, cex = 1.9)

##removing objects no longer needed

rm(loadings, PC,eigenvalues,pca\_result)

#######Correlation Analysis/assumption test 1: at least one linear relationship between predictor and target variable

# Calculate Pearson's correlation coefficients

correlation\_matrix <- cor(data\_set1[, -which(names(data\_set1) == "REVENUE")])

# Set a correlation threshold (adjust as needed)

correlation\_threshold <- 0.7

# Find features with high correlation with the target variable

highly\_correlated\_features <- names(which(abs(correlation\_matrix[, which(names(data\_set1) == "REVENUE")]) > correlation\_threshold))

# Display highly correlated features

print(highly\_correlated\_features)

# Subset the dataset with correlated variables

subset\_data <- data\_set1[, highly\_correlated\_features]

# Calculate the correlation matrix

correlation\_matrix <- cor(subset\_data) #Why weren't these detected by the PCA analysis in previous step?

# Plot the correlation matrix using corrplot

corrplot(correlation\_matrix, method = "color", type = "upper", order = "hclust", tl.col = "black", tl.srt = 45)

# Compute the correlation matrix of independent variables

correlation\_matrix\_indep <- cor(data\_set1[, -which(names(data\_set1) == "REVENUE")])

# Example correlation matrix (replace with your actual correlation matrix)

correlation\_matrix <- cor(data\_set1)

# Set correlation threshold

correlation\_threshold <- 0.8

# Find highly correlated variable pairs

highly\_correlated\_pairs <- which(abs(correlation\_matrix) > correlation\_threshold & upper.tri(correlation\_matrix, diag = FALSE), arr.ind = TRUE)

# Print highly correlated variable pairs

for (i in 1:nrow(highly\_correlated\_pairs)) {

row\_idx <- highly\_correlated\_pairs[i, 1]

col\_idx <- highly\_correlated\_pairs[i, 2]

var1 <- colnames(correlation\_matrix)[row\_idx]

var2 <- colnames(correlation\_matrix)[col\_idx]

cat("Variables", var1, "and", var2, "are highly correlated.\n")

}

#Remove correlated variables, keeping variable that was most statistically significant "MOU"

# library(dplyr)

data\_set1 <- data\_set1 %>%

select(-DROPVCE, -MOUREC, -PHONES, -NumMinStreamVideo) %>%

mutate\_all(as.numeric)

new\_data <- new\_data %>%

select(-DROPVCE, -MOUREC, -PHONES, -NumMinStreamVideo) %>%

mutate\_all(as.numeric)

##removing objects no longer needed

rm(correlation\_matrix,correlation\_matrix\_indep,highly\_correlated\_features,highly\_correlated\_pairs,

col\_idx,i,row\_idx,var1,var2, correlation\_threshold, subset\_data)

#######Data Modeling########

#split the dataset

# reproducible random sampling

set.seed(123)

# creating training data as 80% of the dataset

random\_sample <- createDataPartition(data\_set1$REVENUE,

p = 0.8, list = FALSE)

# generating training dataset

# from the random\_sample

training\_dataset <- data\_set1[random\_sample, ]

# generating testing dataset

# from rows which are not

# included in random\_sample

testing\_dataset <- data\_set1[-random\_sample, ]

# create linear regression model

statistical\_sig <- lm(REVENUE ~ ., data = training\_dataset)

summary(statistical\_sig)

##########Outlier Analysis##############

#create box plots for all variables in the dataset to see which data points are outliers.

boxplot(training\_dataset)

#test model before outliers are handle

statistical\_sig <- lm(REVENUE ~ ., data = training\_dataset)

summary(statistical\_sig)

#Replace outliers using cap/floor technique

# Define the function

adjust\_outliers <- function(x) {

qnt <- quantile(x, probs=c(.25, .75), na.rm = TRUE)

caps <- quantile(x, probs=c(.05, .95), na.rm = TRUE)

H <- 1.5 \* IQR(x, na.rm = TRUE)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

return(x)

}

# Apply the function to each column of the dataset

adjusted\_data <- lapply(training\_dataset, adjust\_outliers)

adjusted\_data <- as.data.frame(adjusted\_data)

# Apply the function to each column of the dataset

adjusted\_test\_data <- lapply(testing\_dataset, adjust\_outliers)

adjusted\_test\_data <- as.data.frame(adjusted\_test\_data)

# Apply the function to each column of the dataset

new\_data <- lapply(new\_data, adjust\_outliers)

new\_data <- as.data.frame(new\_data)

#Create boxplot to check if outliers were replaced properly

boxplot(adjusted\_data)

boxplot(adjusted\_test\_data)

boxplot(new\_data)

#model performance after outliers

statistical\_sig <- lm(REVENUE ~ ., data = adjusted\_data)

summary(statistical\_sig)

######Feature Engineering###########

#estimate the customer's lifetime value

adjusted\_data$CLV <- adjusted\_data$REVENUE \* adjusted\_data$MONTHS

adjusted\_test\_data$CLV <- adjusted\_test\_data$REVENUE \* adjusted\_test\_data$MONTHS

new\_data$CLV <- new\_data$REVENUE \* new\_data$MONTHS

View(new\_data)

View(adjusted\_data)

View(adjusted\_test\_data)

#remove churn variable

adjusted\_data <- adjusted\_data %>%

select(-CHURN) %>%

mutate\_all(as.numeric)

adjusted\_test\_data <- adjusted\_test\_data %>%

select(-CHURN) %>%

mutate\_all(as.numeric)

#model performance after CLV added

# create linear regression model

model <- lm(REVENUE ~ ., data = adjusted\_data)

summary(model)

# Make Predictions on Test data\_set1

predictions <- predict(model, newdata = adjusted\_test\_data)

predictions1 <- predict(model, newdata = new\_data)

# Evaluate Model Performance

mse <- mse(adjusted\_test\_data$REVENUE, predictions)

rmse <- rmse(adjusted\_test\_data$REVENUE, predictions)

mae <- mae(adjusted\_data$REVENUE, predictions)

r\_squared <- R2(adjusted\_test\_data$REVENUE, predictions)

# Display Evaluation Metrics

cat("Mean Squared Error (MSE):", mse, "\n")

cat("Root Mean Squared Error (RMSE):", rmse, "\n")

cat("Mean Absolute Error (MAE):", mae, "\n")

cat("R-squared:", r\_squared, "\n")

# Evaluate Model Performance1

mse <- mse(new\_data$REVENUE, predictions1)

rmse <- rmse(new\_data$REVENUE, predictions1)

mae <- mae(new\_data$REVENUE, predictions1)

r\_squared <- R2(new\_data$REVENUE, predictions1)

# Display Evaluation Metrics1

cat("Mean Squared Error (MSE):", mse, "\n")

cat("Root Mean Squared Error (RMSE):", rmse, "\n")

cat("Mean Absolute Error (MAE):", mae, "\n")

cat("R-squared:", r\_squared, "\n")

# Statistical Tests

anova\_result <- anova(model) # Analysis of Variance (ANOVA)

summary(model) # Display detailed summary of the linear regression model

###########Model with most impoprtant variables and k-fold cross validation#############

model\_new <- lm(REVENUE ~ DataUsageGB + CompetitivePackage + EQPDAYS + SETPRC, data = adjusted\_data)

summary(model\_new)

# Make Predictions on Test data\_set1

predictions <- predict(model\_new, newdata = adjusted\_test\_data)

# Evaluate Model Performance

mse <- mse(adjusted\_test\_data$REVENUE, predictions)

rmse <- rmse(adjusted\_test\_data$REVENUE, predictions)

mae <- mae(adjusted\_test\_data$REVENUE, predictions)

r\_squared <- R2(adjusted\_test\_data$REVENUE, predictions)

# Display Evaluation Metrics

cat("Mean Squared Error (MSE):", mse, "\n")

cat("Root Mean Squared Error (RMSE):", rmse, "\n")

cat("Mean Absolute Error (MAE):", mae, "\n")

cat("R-squared:", r\_squared, "\n")

#######Testing Assumptions############

#####Assumption 1: Linear relationship between target vraiable and predictor variables

# Calculate Pearson's correlation coefficients

correlation\_matrix <- cor(data\_set1[, -which(names(data\_set1) == "REVENUE")])

# Set a correlation threshold (adjust as needed)

correlation\_threshold <- 0.7

# Find features with high correlation with the target variable

highly\_correlated\_features <- names(which(abs(correlation\_matrix[, which(names(data\_set1) == "REVENUE")]) > correlation\_threshold))

# Display highly correlated features

print(highly\_correlated\_features)

#####Assumption 2: Homoscedasticity: The residuals have constant variance at every level of x.

# Extract residuals from the linear regression model

residuals <- residuals(model\_new)

# Ensure that both data sets have the same number of rows

n\_rows <- min(nrow(adjusted\_data), nrow(adjusted\_test\_data))

# Create a data\_set2aframe with predictions and residuals

residual\_df <- data.frame(

predictions = predictions[1:n\_rows],

residuals = residuals[1:n\_rows]

)

# Plot residuals vs. fitted values

plot(residual\_df$predictions, residual\_df$residuals,

main = "Residuals vs. Fitted Values",

xlab = "Fitted Values", ylab = "Residuals")

#####Assumption 3: Residuals are normally distributed.

# Plot a histogram of residuals

hist(residuals, main = "Histogram of Residuals", xlab = "Residuals")

# Plot a Q-Q plot of residuals

qqnorm(residuals)

qqline(residuals)

#####Assumption 4: Test if there is correlation in the residuals

# Install and load the 'car' package

# install.packages("car")

# library(car)

# Perform the Durbin-Watson test

durbinWatsonTest(model)

###Test Heteroscedasticity

# Load necessary libraries

# library(lmtest)

# Perform the Breusch-Pagan test for heteroscedasticity

bptest(model\_new)