MASTER OF SCIENCE IN APPLIED DATA SCIENCE

Portfolio Report

Satwik Belaldavar

Email: sbelalda@syr.edu

SU ID: 429747172

SATWIK MRITYUNJAY BELALDAVAR

Syracuse, NY • (315) 863-6513 • sbelalda@syr.edu • LinkedIn • Github

EDUCATION

Syracuse University, School of Information Studies, Syracuse, NY **August 2022 - May 2024**
M.S. Applied Data Science
Relevant Coursework: Introduction to Data Science, Data Admin Concepts and Database Mgmt, Scripting for Data Analysis, Applied Machine Leaning, Quantitative Reasoning, Business Analytics, Data Warehouse, Text Mining, Big Data Analytics, Visual Analytics Dashboard, Cloud Management

KLE Technological University, School of Computer Science and Engineering, Hubballi, India **August 2017 - May 2021**
B.E. Computer Science and Engineering
Relevant Coursework: Database Management System, Applied Statistics, Data Mining and Analysis

EXPERIENCE

Data Science Researcher, NEXIS Student Technology Lab, Syracuse, NY **February 2023 – Present**

- Develop an image captioning system with **CNN (VGG16 model) and LSTM** networks, analyzing over **8000** images from Flicker8k dataset to enhance model's ability to generate contextually accurate captions
- Implement **BLEU** scores for quality assessment, achieving a **BLEU-1** score of **55.43%**, indicates proficient caption matching at word level, with ongoing efforts to improve complex phrase generation

Associate Engineer, L&T Technology Services, Mysore, India **June 2021 - June 2022**

- Led a **6**-person team in transitioning websites to modern **SharePoint**, employing agile workflows and a content strategy to cut conversion time by **15%**
- Optimized costs by consolidating **2** applications merging data from different tables using **SQL and Python** for data analysis, cleaning, reducing redundancies with 'AppRationalizer'
- Collaborated with a **4**-person team to analyze over **10,000** files on department's SharePoint site using **PowerBI** dashboards, deleting **2,000** obsolete files and achieving a **20%** reduction in storage costs

PROJECTS

A Comprehensive Data-Driven Analysis for Enhancing University Operations **September 2023 - December 2023**

- Spearheaded modernization of data management for **2000-2500** students and **100** professors by adopting **Snowflake** as data warehousing platform, designing **ETL processes** with **dbt** to enhance academic data usability
- Developed **3** distinct dashboards using **PowerBI** to showcase regional engagement metrics, supporting advanced analytics and insightful reporting

Analysis of ChatGPT Impact on NLP Research Using Advanced Data Techniques **September 2023 - December 2023**

- Analyzed ChatGPT's impact on NLP through **Latent Dirichlet Allocation (LDA)** on over **5000** ACL Anthology papers, identifying key research trends and enhancing understanding of language models
- Evaluated **BART Large-CNN, GPT-4, and T-Summarization Model** for text summarization, achieving high accuracy, streamlining synthesis of complex academic content

Wholesale Management System **September 2022 - December 2022**

- Developed a user-friendly front-end using Microsoft **PowerApps**, enhancing operational efficiency for over **400** users
- Managed backend database development with **Azure Data Studio**, crafting **4** complex SQL queries and scripts for a system supporting transaction process for over **1000** daily operations

Uber Ride Request Analysis **September 2022 - December 2022**

- Analyzed **2000+** Uber ride requests using **Python** for data analysis, identifying peak demand times, optimizing ride allocation strategies to minimize cancellations and no-car scenarios
- Improved data usability by cleaning, restructuring a dataset with **346+** missing entries using **Python**, analyzed cancellation rates and demand-supply gaps with **pivot tables** to aid strategic decisions

CERTIFICATIONS/ TECHNICAL SKILLS

Programming Languages: Python, R, SQL
Tools: MS Excel, PowerBI, PowerApps, Azure, SharePoint, Docker, Tableau, Google Analytics, Snowflake, Github
Libraries: NLTK, TensorFlow, Keras, SciPy, Matplotlib, Scikit-learn, pandas, NumPy, Pytorch, Seaborn, Pyspark
AWS: S3, Glue, Lambda, Athena, Identity and Access Management (IAM), QuickSight, CloudWatch
Statistical Test: Hypothesis, T-Test, ANOVA, Chi-Square, Pearson Correlation, Principal Component Analysis (PCA)
ML Algorithm: Decision Trees, Random Forest, Support Vector Machines (SVM), Naive Bayes, KNN
Certifications: Microsoft Office Specialist Excel Associate 2019

Table of Contents

Introduction

The Applied Data Science program at Syracuse University equips students with the essential competencies required to excel as data professionals adept at resolving issues through data analysis. The program focuses on:

- Gathering, transforming, and storing data effectively.
- Mastering data visualization techniques to identify trends, patterns, and to effectively communicate insights.
- Applying both supervised and unsupervised machine learning methods to comprehend data and forecast outcomes.
- Presenting analytical results in a manner that is accessible to audiences with diverse levels of technical expertise.
- Formulating actionable recommendations based on data analysis for businesses, organizations, management, or shareholders.

The following projects demonstrate the skills I have learned in various classes of my program:

- IST 687 Introduction to Data Science
- IST 718 Big Data Analytics
- IST 722 Data Warehouse

IST 687 Introduction to Data Science

In this course, I gained practical experience in the entire data lifecycle, including collection, processing, transformation, management, and analysis. I acquired knowledge in applied statistics, data visualization, text mining, and machine learning to interpret and assess data. Additionally, I developed programming skills in R within the R-Studio environment.

Project Description

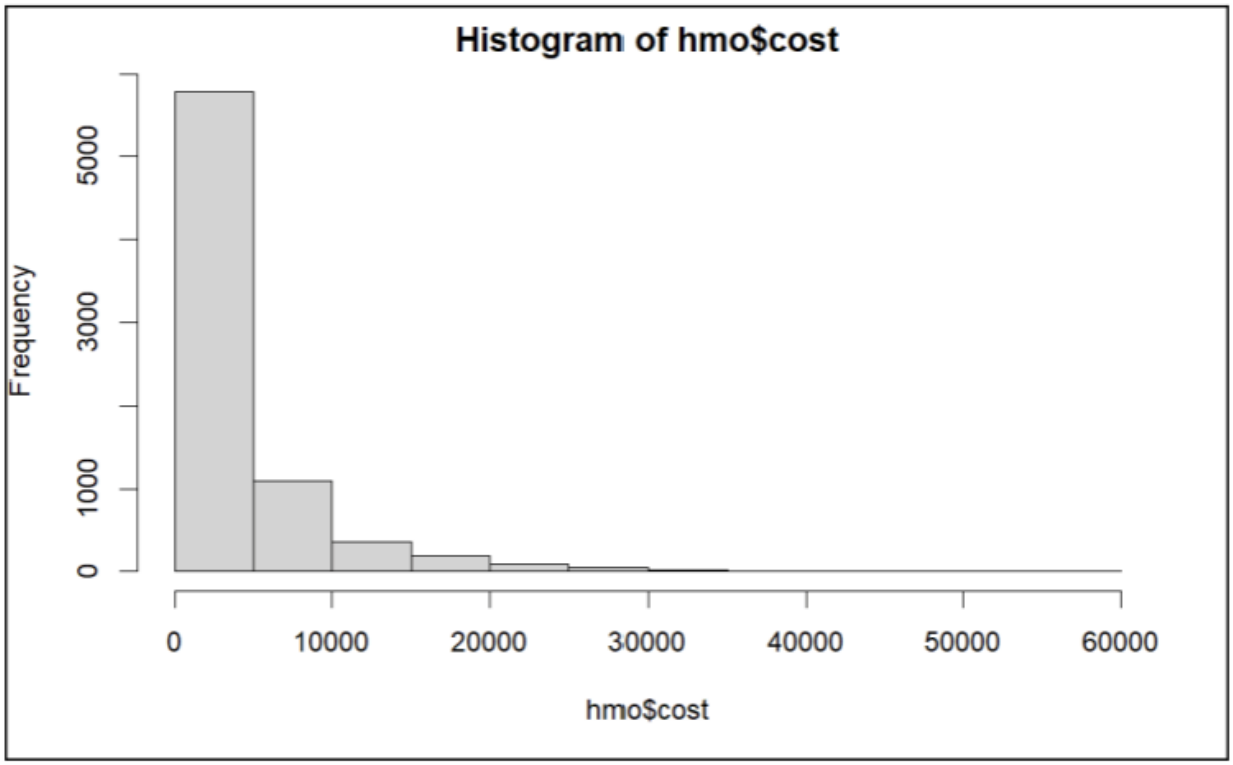
Goal: Analyze healthcare cost data to identify and predict high-cost individuals, providing the Health Management Organization with actionable insights and recommendations to reduce healthcare expenses.

About the data: The dataset for this project contains healthcare cost information from an HMO, where each row represents an individual. It encompasses variables such as age, location, lifestyle factors like exercise and smoking habits, BMI, medical history, and the total healthcare costs for the year. This comprehensive data set allows for an in-depth analysis of the factors influencing healthcare expenses.

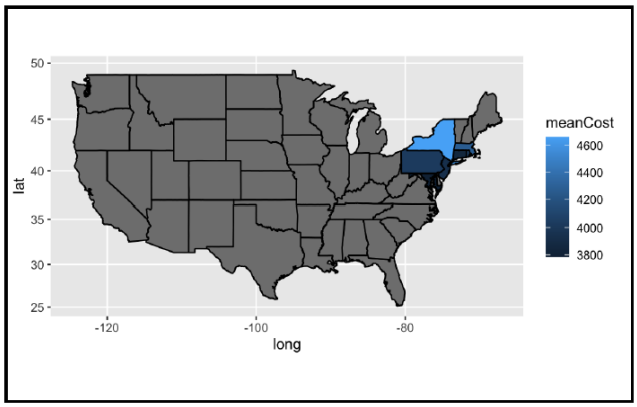
Step 1: Data Cleaning and Preparation

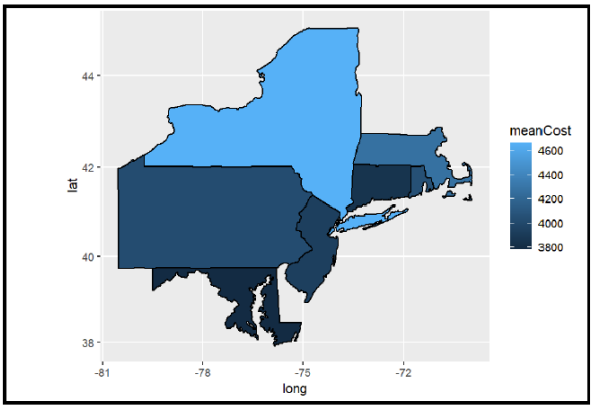
Our team scrutinized the dataset for any missing values and converted the necessary columns into categorical factors, making them suitable for inclusion in our machine learning models.

Step 2: Visualize the data



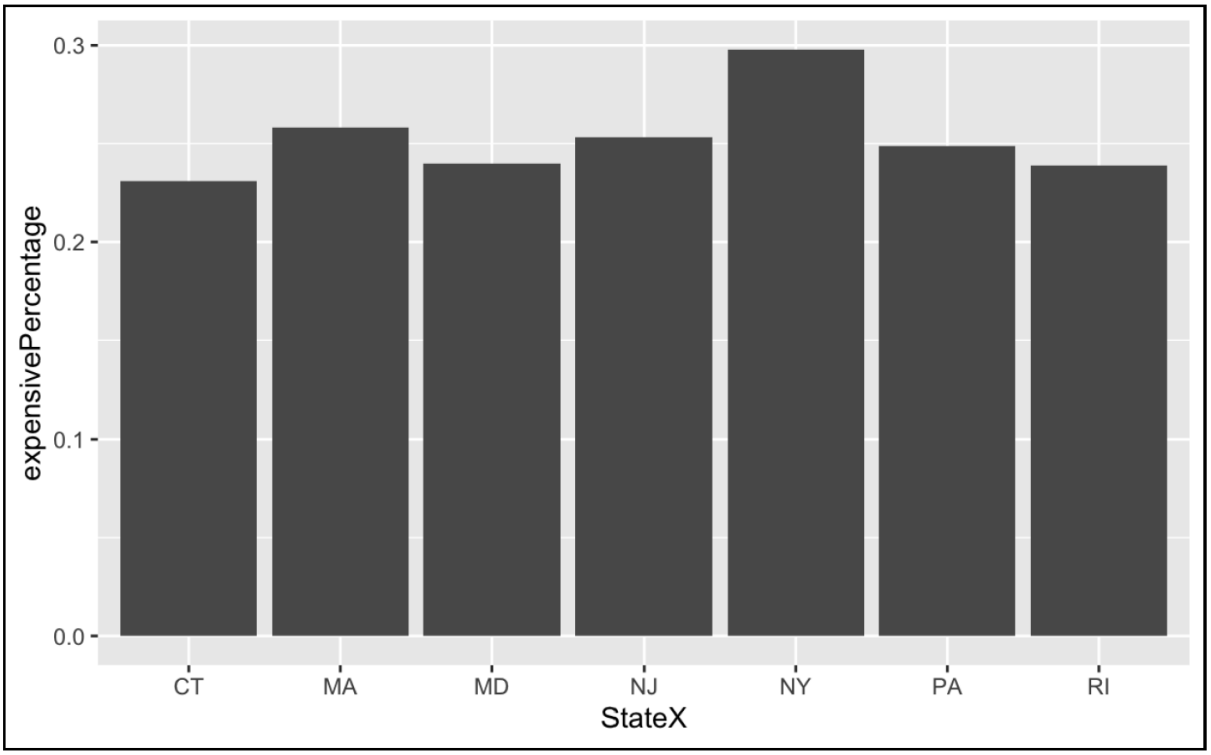
A simple histogram was constructed to visualize the distribution of healthcare costs, which confirmed the data's heavy rightward skew. Most costs are clustered between $0 and $5,000, yet there is a noticeable presence of outliers with costs exceeding $20,000.



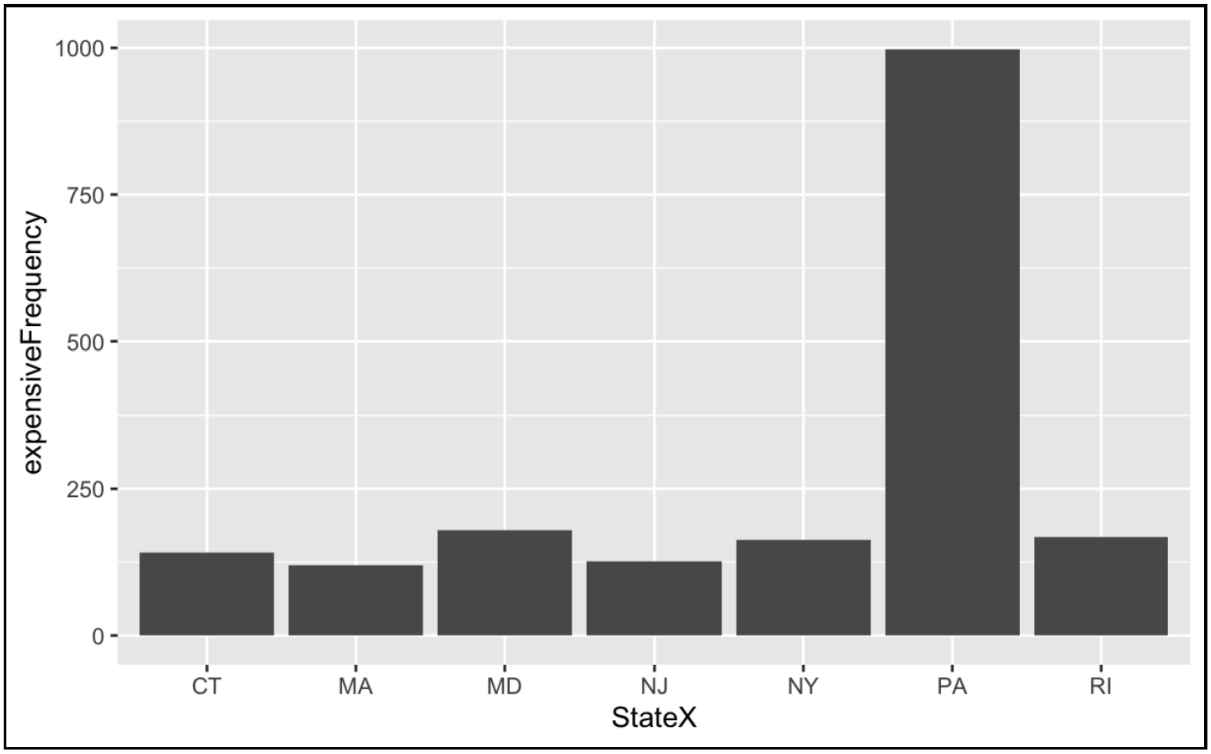


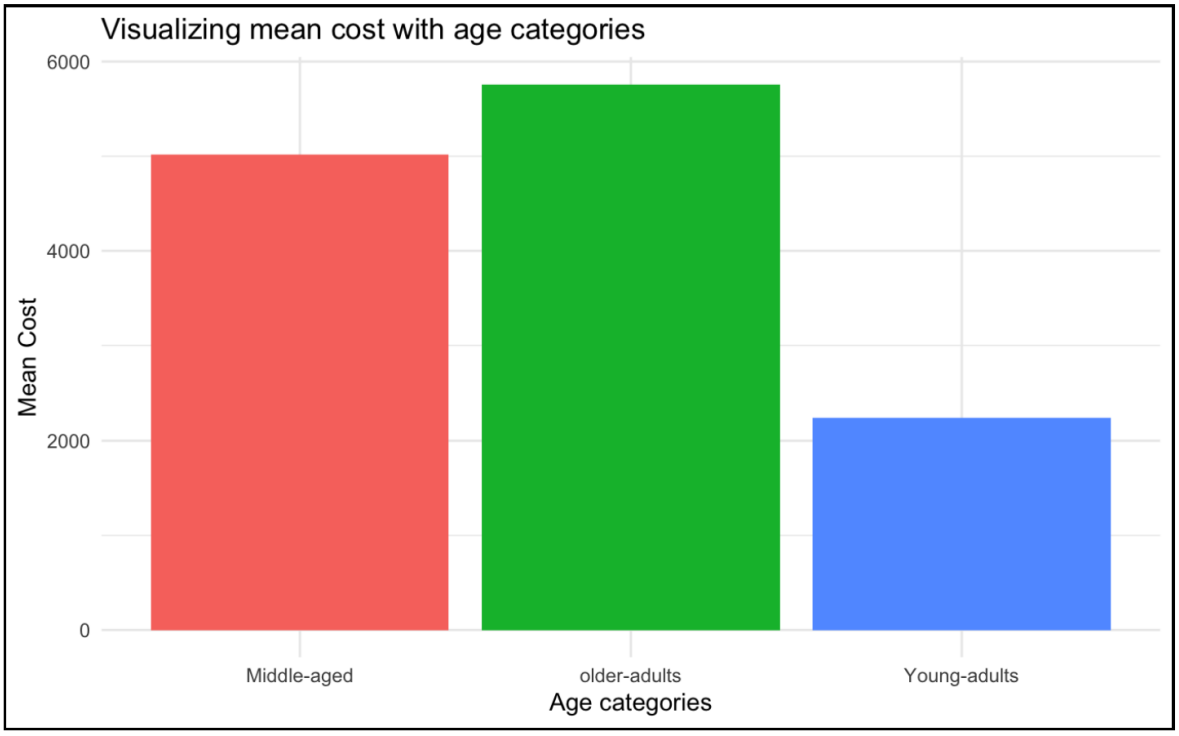
The dataset was segmented by state to produce two visualizations illustrating the average healthcare costs by region. These maps highlight New York as having notably higher average costs, with a discernible gradient across states, reflecting the geographical variance in healthcare spending.

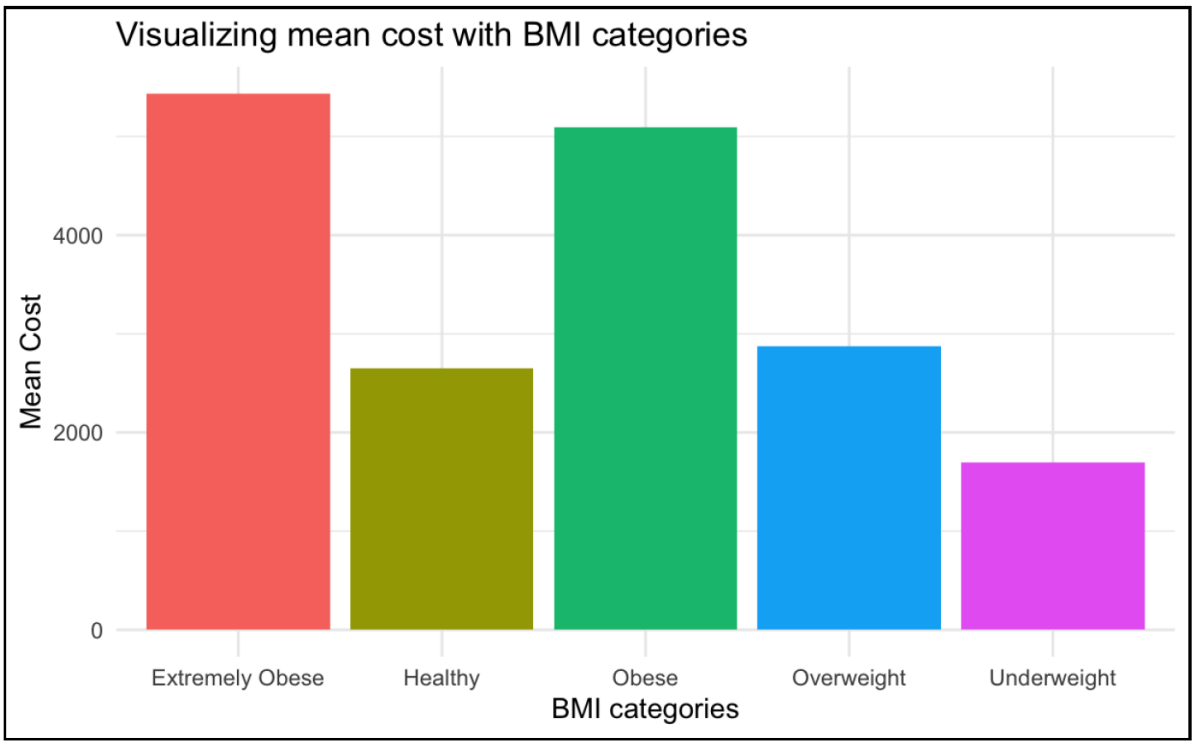
Visualizing the expensive percentage by state:



Visualizing the expensive frequency by state:




Visualizing mean cost with age categories
6000
4000
2000
0
Mean Cost
Middle-aged
older-adults
Young-adults
Age categories


Visualizing mean cost with BMI categories
4000
2000
0
Mean Cost
Extremely Obese
Healthy
Obese
Overweight
Underweight
BMI categories

Step 3: Create and test the models.

1. Linear Model

We began creating a model to solve the problem that we faced. First, we created a multiple linear regression model using all variables in the dataframe to model cost. We found that age, bmi, children, smoker, and being from New York were positive and statistically significant at the 0.05 level. This means that an increase in age, bmi, or children leads to an increase in insurance costs. Being a smoker or being from New York also leads to higher costs. We then ran a model with only statistically significant variables and the results are shown below.

```
hmo_1 <- select(hmo, -expensive)
modelAll_1 <- lm(cost ~., data = hmo_1)
summary(modelAll_1)

#model with significant predictors
modelSignificant <- lm(cost ~ age + bmi + children + smoker + location + education_level + exercise +
married + hypertension, data = hmo)
summary(modelSignificant)
```

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  -9101.452    259.363 -35.092  < 2e-16 ***
age                            102.399      2.629  38.952  < 2e-16 ***
bmi                            181.439      6.221  29.164  < 2e-16 ***
children                       233.250     30.444   7.662 2.06e-14 ***
smokeryes                     7665.446     93.437  82.039  < 2e-16 ***
locationMARYLAND              -129.468    175.706  -0.737 0.461238
locationMASSACHUSETTS            8.140    198.350   0.041 0.967268
locationNEW JERSEY             108.036    194.511   0.555 0.578621
locationNEW YORK               470.417    189.706   2.480 0.013170 *
locationPENNSYLVANIA            14.409    139.922   0.103 0.917982
locationRHODE ISLAND           118.118    178.143   0.663 0.507317
education_levelMaster          -97.611     95.102  -1.026 0.304741
education_levelNo College Degree  42.000    126.283   0.333 0.739453
education_levelPhD            -234.554    129.865  -1.806 0.070937 .
exerciseNot-Active            2261.702     85.619  26.416  < 2e-16 ***
marriedNot_Married             132.345     78.548   1.685 0.092051 .
hypertension                   341.360     92.739   3.681 0.000234 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3220 on 7565 degrees of freedom
Multiple R-squared:  0.574,     Adjusted R-squared:  0.5731
F-statistic: 637.1 on 16 and 7565 DF,  p-value: < 2.2e-16
```

The p-value is less than 0.05, we accepted this model. The Adjusted R-squared value for this was 0.5731. This means that 57.31% of the variation in health cost can be explained by the variables included in the model.

2. SVM Model

We then decided to create an SVM model, which is a supervised modeling technique. To do this, we used the createDataPartition () function from the caret package. We first tried splitting the data into test and training datasets with .2 and .8 partitions, but that model is not as accurate as the model we created when we split the data into test and training datasets with ⅔ of the data being used as the training data.\

```
library(caret)
library(kernlab)
hmo$expensive <- as.factor(hmo$expensive)

HMO <- select(hmo,-cost,-X)

set.seed(687)
trainList <- createDataPartition(y=HMO$expensive,p=.67,list=FALSE)
training <- HMO[trainList,]
testing <- HMO[-trainList,]
```

Next, we created an SVM model using all the variables with expensive being the variable of interest. This was created using the train () function. A second SVM model was made using the ksvm () function. Then, we predicted the test data with both models and created a Confusion matrix for each.

```
svm.model1 <- train(expensive ~ ., data = training,
method = "svmRadial",
trControl=trainControl(method = "none"),
preProcess = c("center", "scale"))
svm.model1

# another way to train the model
svm.model2 <- ksvm(expensive ~., data=training,
C=5,cross=3, prob.model = TRUE)
svm.model2

svmPred1 <- predict(svm.model1, newdata=testing)
confusionMatrix(svmPred1, testing$expensive)

svmPred2 <- predict(svm.model2, newdata=testing)
confusionMatrix(svmPred2, testing$expensive)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1825  267
         1   51  358

               Accuracy : 0.8729
                 95% CI : (0.8592, 0.8857)
    No Information Rate : 0.7501
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6167

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9728
            Specificity : 0.5728
         Pos Pred Value : 0.8724
         Neg Pred Value : 0.8753
             Prevalence : 0.7501
         Detection Rate : 0.7297
   Detection Prevalence : 0.8365
      Balanced Accuracy : 0.7728

       'Positive' Class : 0
```

(Model 1)

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1806  220
         1   70  405

               Accuracy : 0.884
                 95% CI : (0.8708, 0.8963)
    No Information Rate : 0.7501
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6638

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9627
            Specificity : 0.6480
         Pos Pred Value : 0.8914
         Neg Pred Value : 0.8526
             Prevalence : 0.7501
         Detection Rate : 0.7221
   Detection Prevalence : 0.8101
      Balanced Accuracy : 0.8053

       'Positive' Class : 0
```

(Model 2)

The P-Value for both models is less than 0.05, the no information rate is 0.7501. The accuracy of the first model was 0.8729 while the second model was 0.884. However, the sensitivity of the first model was 0.9728, higher than the sensitivity of the second model 0.9627. We then decided to use the first model to predict the test sample.

```r
testData <- read_csv("HMO_TEST_data_sample.csv")

testData$agecategory[18<=testData$age & testData$age<=34] <- "Young-adults"
testData$agecategory[35<=testData$age & testData$age<=50] <- "Middle-aged"
testData$agecategory[51<=testData$age & testData$age<=66] <- "older-adults"

testData$bmicategory[testData$bmi <= 18 ] <- "Underweight"
testData$bmicategory[testData$bmi >= 18 & testData$bmi < 25  ] <- "Healthy"
testData$bmicategory[testData$bmi >= 25 & testData$bmi < 30  ] <- "Overweight"
testData$bmicategory[testData$bmi >= 30 & testData$bmi < 40  ] <- "Obese"
testData$bmicategory[testData$bmi >= 40 & testData$bmi < 65  ] <- "Extremely Obese"

testData <- select(testData,-X)

testing2 <- select(testing,-expensive)
total <- rbind(testing2,testData)

svmPred3 <- predict(svm.model1, total)

svmPred3[2502:2521]
testData$expensive_svm <- svmPred3[2502:2521]
```

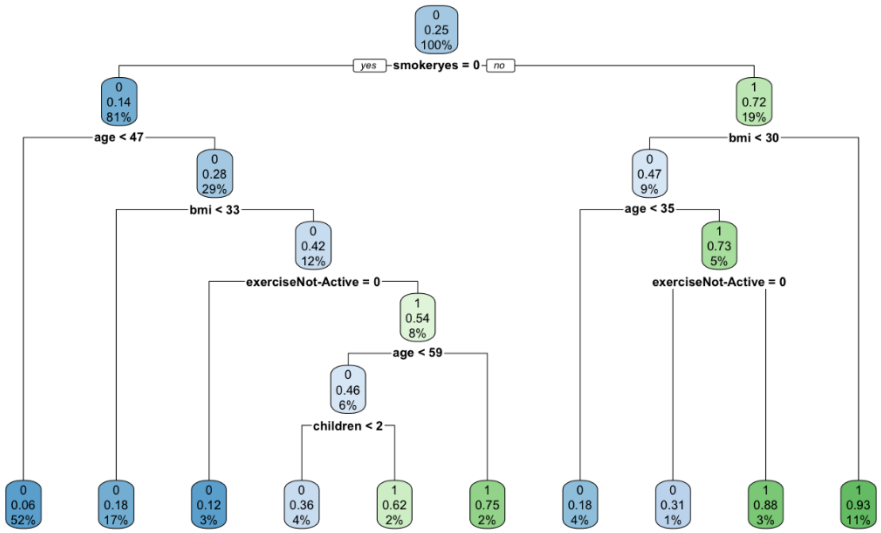
```
[1] 0 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0
```

3. Tree Model

We then generated a tree model.

```
library(e1071)
# train the model with rpart method
trctrl <- trainControl(method="repeatedcv", number=10)
model.rpart <- train(expensive ~ ., method = "rpart",
data = training,
trControl=trctrl,tuneLength = 50)
# getting the result of model.rpart
model.rpart

library(rpart.plot)
## Loading required package: rpart
library(rpart)
# getting the plot of model.rpart$finalModel
rpart.plot(model.rpart$finalModel)

predictValues <- predict(model.rpart,newdata=testing)
# getting the confusion matrix
# looking at the accuracy, no information rate and the p-value
confusionMatrix(predictValues, testing$expensive)

predictValues2 <- predict(model.rpart,newdata=testData)
# getting the confusion matrix
# looking at the accuracy, no information rate and the p-value
predictValues2
testData$expensive_tree <- predictValues2
```

```
          Reference
Prediction    0    1
         0 1817  211
         1   59  414

               Accuracy : 0.892
                 95% CI : (0.8792, 0.9039)
    No Information Rate : 0.7501
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6866

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9686
            Specificity : 0.6624
         Pos Pred Value : 0.8960
         Neg Pred Value : 0.8753
             Prevalence : 0.7501
         Detection Rate : 0.7265
   Detection Prevalence : 0.8109
      Balanced Accuracy : 0.8155

       'Positive' Class : 0
```



Compared with the SVM models, the Accuracy, the Pos Pred Value, and the Neg Pred Value of the Tree model were a little higher. However, the Sensitivity was lower than the first SVM Model.

We used the tree model to predict the test sample.

```
 [1] 0 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0
Levels: 0 1
```

For a test sample of 20 people, the SVM model and the tree model were used to predict whether these people were expensive. Though the first SVM model is our best model (because it has the highest sensitivity), we got the same results from the 2 models.

expensive_svm	expensive_tree
0	0
0	0
1	1
1	1
1	1
1	1
1	1
1	1
0	0
0	0
1	1
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0

Step 5: Association Rules

Next, we used association rules mining to predict whether someone would be expensive or not.

```
library(arules)
library(arulesViz)
hmo_new <- data.frame(X=as.factor(hmo$X),
                      agecategory=as.factor(hmo$agecategory),
                      bmicategory=as.factor(hmo$bmicategory),
                      children=as.factor(hmo$children),
                      smoker=as.factor(hmo$smoker),
                      location=as.factor(hmo$location),
                      location_type=as.factor(hmo$location_type),
                      education_level=as.factor(hmo$education_level),
                      yearly_physical=as.factor(hmo$yearly_physical),
                      exercise=as.factor(hmo$exercise),
                      married=as.factor(hmo$married),
                      hypertension=as.factor(hmo$hypertension),
                      gender=as.factor(hmo$gender),
                      expensive=as.factor(hmo$expensive)
)
hmoX <- as(hmo_new, "transactions")
#itemFrequency(hmoX)
itemFrequencyPlot(hmoX,topN=20)
inspect(hmoX[1:10])



ruleset <- apriori(hmoX,
                   parameter = list(support = 0.05,confidence = 0.83),
                   control=list(verbose=F),
 appearance=list(default="lhs",rhs=("expensive=1")))
summary(ruleset)
inspectDT(ruleset)
```

We generated 16 rulesets, and these were the sets of rules with the highest lift:

	LHS	RHS	support	confidence	coverage	lift	count
	All	All	All	All	All	All	All
[11]	{bmicategory=Obese,smoker=yes,yearly_physical=No,exercise=Not-Active}	{expensive=1}	0.051	0.997	0.051	3.991	383.000
[7]	{bmicategory=Obese,smoker=yes,exercise=Not-Active}	{expensive=1}	0.066	0.996	0.066	3.985	500.000
[12]	{bmicategory=Obese,smoker=yes,exercise=Not-Active,hypertension=0}	{expensive=1}	0.052	0.995	0.052	3.981	394.000
[5]	{bmicategory=Obese,smoker=yes,married=Married}	{expensive=1}	0.057	0.946	0.061	3.783	434.000
[2]	{bmicategory=Obese,smoker=yes}	{expensive=1}	0.084	0.940	0.090	3.760	639.000
[4]	{bmicategory=Obese,smoker=yes,education_level=Bachelor}	{expensive=1}	0.053	0.937	0.056	3.748	400.000
[3]	{bmicategory=Obese,smoker=yes,gender=male}	{expensive=1}	0.055	0.937	0.058	3.748	414.000
[8]	{bmicategory=Obese,smoker=yes,yearly_physical=No}	{expensive=1}	0.064	0.934	0.068	3.739	485.000
[9]	{bmicategory=Obese,smoker=yes,hypertension=0}	{expensive=1}	0.065	0.934	0.070	3.736	494.000

If a person is a smoker, with the BMI >= 30 & < 40, and at the same time neither have yearly physical nor active exercise, this person has a higher probability of being expensive in health care cost.

Conclusion

Upon visualizing the initial data, we developed a multiple linear regression model incorporating all available variables to estimate healthcare costs. The analysis revealed that factors such as age, BMI, number of children, smoking status, and residency in New York significantly contribute to increased insurance costs at a 0.05 significance level. An uptick in any of these variables was associated with a rise in costs, with smoking and New York residency being particularly impactful.

Consequently, a refined model was constructed with only these significant predictors, leading to a more streamlined analysis of cost drivers.

Skills learned from this project:

- Loaded, cleansed, and prepped the dataset for comprehensive analysis.
- Developed insightful visual representations from the dataset to identify key patterns.
- Trained and evaluated Linear and SVM models, assessing the influence of different variables.
- Formulated strategic recommendations based on the outcomes of the data analysis.

IST 718 Big Data Analytics

This course offers a deep dive into Python and its associated libraries, focusing on real-time, large-scale data processing. Students learn to navigate big data characterized by volume, velocity, variety, and more, utilizing platforms like PySpark and tools such as Google Colab. And it promises to equip students with the skills to leverage AI and ML integration, understand edge computing, and implement multi-cloud strategies for varied big data applications, from optimizing supply chains to enhancing customer insights.

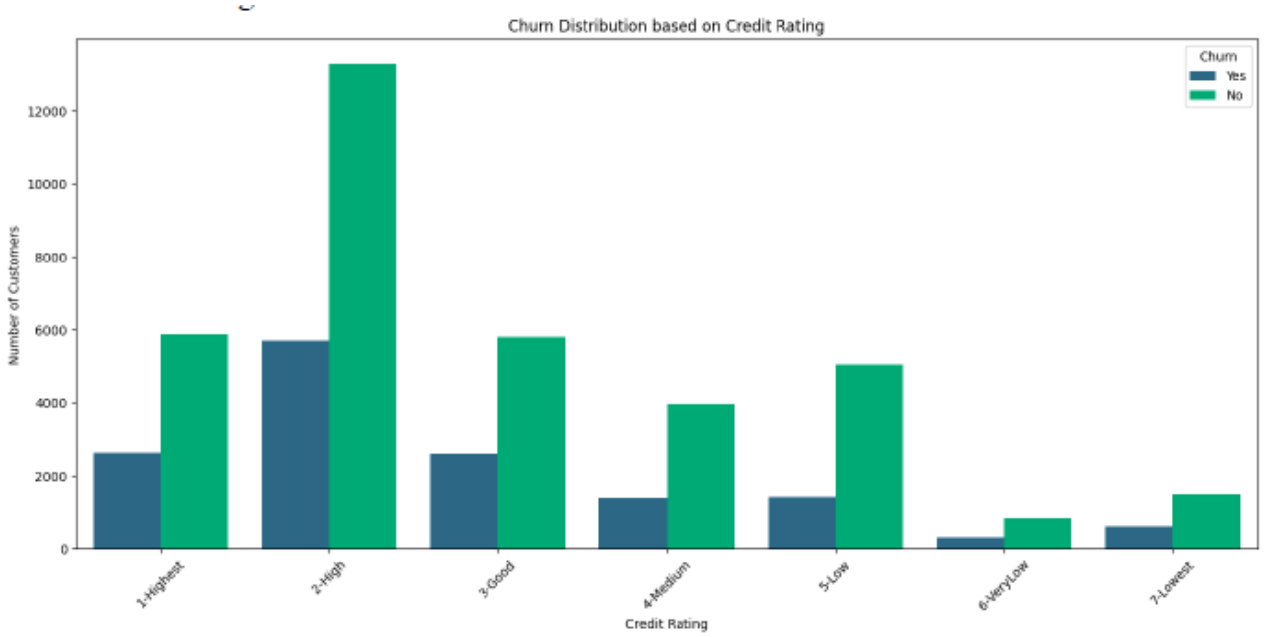
Project Description

Goal: To develop a model to predict and analyze customer churn in the telecom sector, aiming to enhance retention strategies by identifying customers at high risk of contract cancellation. The initiative is designed to enable targeted retention campaigns, offering special incentives to those identified customers to reduce churn rates and understand the underlying reasons for their potential departure.

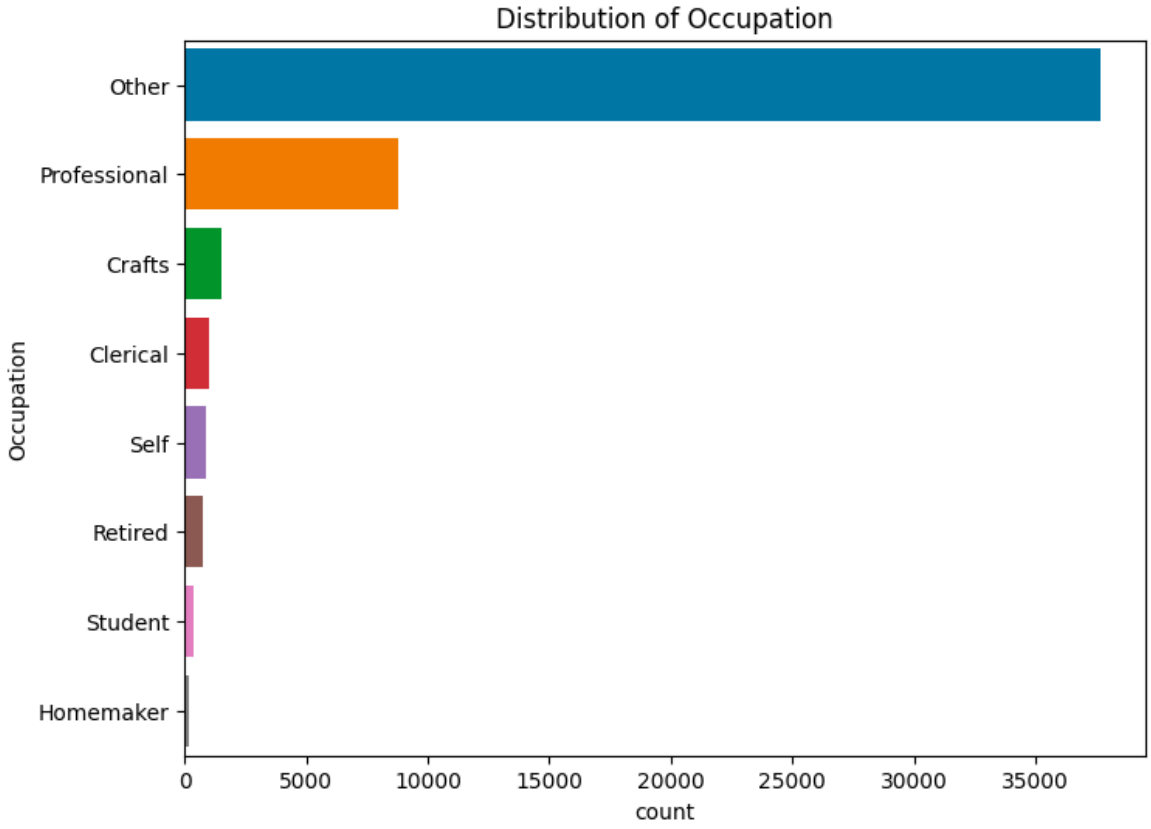
About the data: The Cell2Cell dataset, sourced from the Teradata Center for Customer Relationship Management at Duke University, focuses on customer churn within the telecom sector. It includes 51,047 entries and 58 attributes, detailing various aspects such as customer satisfaction, behavior, and profitability. This comprehensive dataset provides a foundation for analyzing and predicting customer churn, enabling analysts to identify potential churners and understand the factors contributing to their decision to switch service providers.

Step 1: Data Visualization

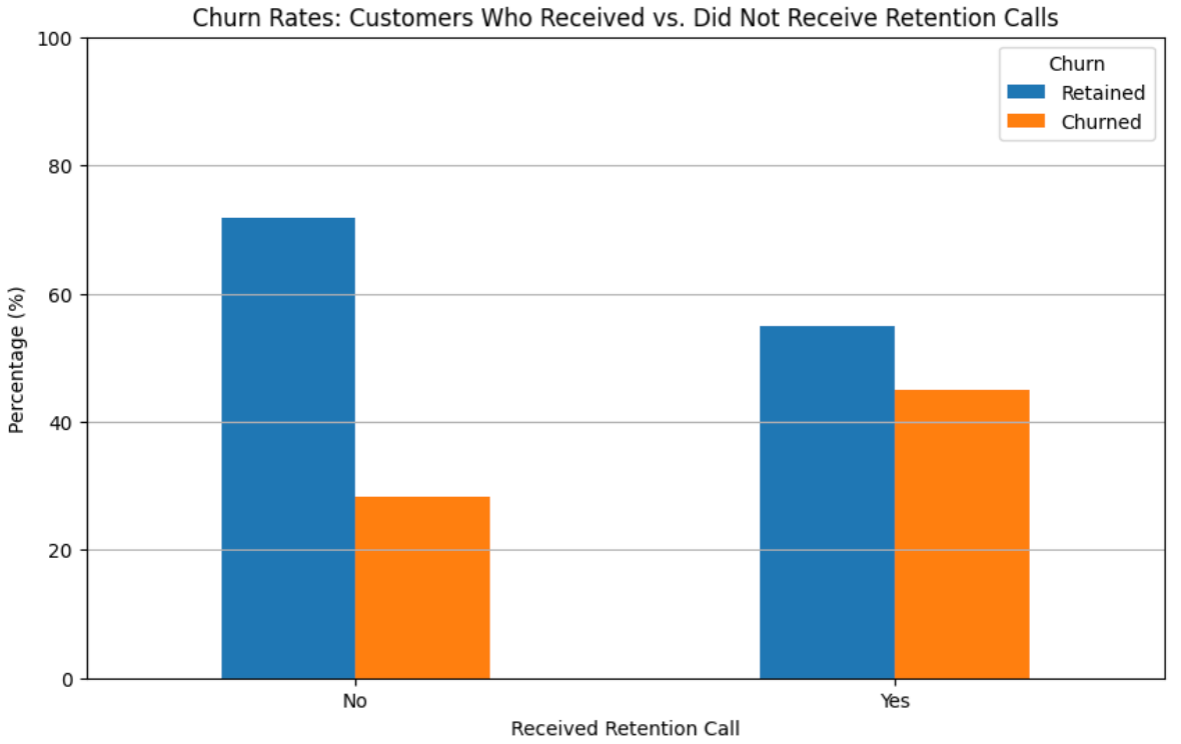
1. The visualizations suggest that customers' credit ratings significantly impact churn, with majority of retained customers being in top 3 rating categories. This insight directs us to consider credit rating as a key driver of customer churn, possibly due to misalignment between service offerings and the financial expectations or capabilities of various customer segments.



2. The occupation distribution chart hints at a potential relationship between customers' occupations and their service usage patterns. This can inform targeted improvements in customer service and product offerings, especially to the professional and other high frequency segments shown in the data.



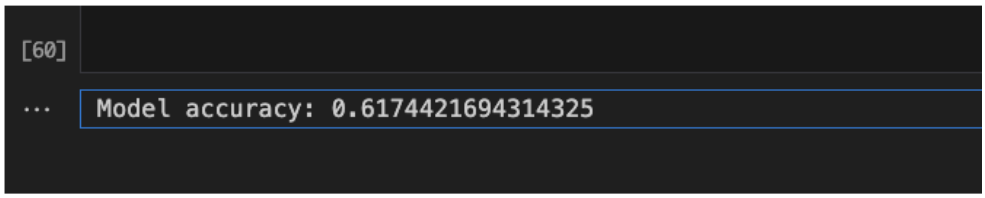
3. We find that retention calls play a vital role in mitigating customer churn. However, the persistence of churn despite these calls and the retention of many customers without such intervention points to other factors influencing customer loyalty. This suggests that while retention calls are useful, there's room to enhance their effectiveness and investigate other loyalty drivers.



Step2: Predictions and results

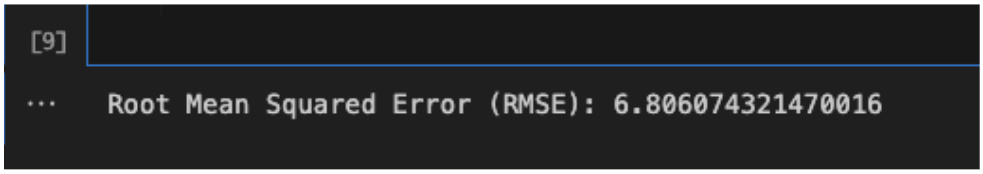
1. Customer Churn Prediction

   The 'Customer Churn Prediction' report segment details a predictive model using a Random Forest Classifier to forecast customer attrition, with an achieved accuracy of 61.74%. The model, trained on 80% of the dataset with preprocessing like binary encoding and null value handling, helps identify customers likely to discontinue service. This prediction enables targeted actions to enhance retention, which is crucial for maintaining revenue and customer base. The moderate accuracy indicates potential for model optimization to improve prediction reliability.



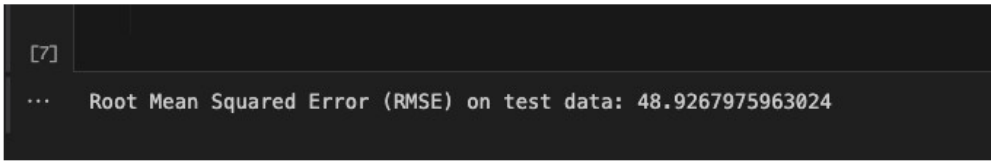
2. Monthly Revenue Forecasting

   The "Monthly Revenue Forecasting" report segment describes a predictive model constructed using a Random Forest Regressor to estimate future revenue, achieving a Root Mean Squared Error (RMSE) of 6.80. The model was trained on 80% of the data, which underwent preprocessing such as conversion of string data to float and handling null values to ensure quality inputs. This process of forecasting monthly revenue is vital for the business to make strategic decisions, allocate resources effectively, and understand market trends. The RMSE indicates the average difference between the revenues predicted by the model and the actual figures, providing a quantitative measure of the model's accuracy. The obtained RMSE suggests a degree of precision that could be improved through further model refinement, aiming to enhance the reliability of the revenue forecasts.



3. Overage Minutes Prediction

   The "Overage Minutes Prediction" report segment outlines a predictive model using a Random Forest Regressor to estimate the number of overage minutes — time exceeding the plan's allotted minutes — with a Root Mean Squared Error (RMSE) of 48.92% on the test data. Trained on 80% of the dataset and preprocessed to handle categorical variables and null values, the model is designed to forecast the additional minutes customers are likely to use. Understanding overage patterns is beneficial for optimizing plan structures and potentially increasing revenue. The given RMSE offers insight into the model's prediction accuracy and highlights opportunities for further tuning to enhance forecast precision.



Conclusion

Our project achieved its prediction and inference goals with moderate success. The churn prediction model, using a Random Forest Classifier, attained a 61.74% accuracy, suggesting a foundation for targeted retention actions but also highlighting room for optimization. Revenue forecasting with a Random Forest Regressor yielded an RMSE of 6.80, indicating a reasonable prediction accuracy, while overage minutes prediction achieved an RMSE of 48.92%, pointing to further opportunities for model tuning. These results show a good starting point for practical applications and future enhancements in churn prediction and revenue forecasting within the telecom industry.

IST 722 Data Warehouse

This course provides a deep understanding of data warehouse architectures, tools, and dimensional modeling, enhancing their ability to integrate data and apply business intelligence solutions. With a strong emphasis on practical, hands-on experience using modern technologies, the course prepares us to effectively tackle real-world data challenges.

Project Description

Goal: Our project aimed to gain a deep insight into university operations through a data-driven approach. We explored online and offline courses, profiles of instructors and students, departmental structures, and external data from S3 containing course ratings and reviews. Choosing Snowflake as our data warehousing platform, we laid the foundation for a modern data management system.

Step 1: Data Profiling

Table	Type	Row Count	Business Key	One Row Is
Online course	Master data	4	Course_url	Course URL
Onsite_course	Master data	6	None (PK's used)	a course
course	Master data	10	Course_title	A course
Course_instructor	Business process	9	None (pk used)	A course taught by a professor
Person	Master data	34	None (pk used)	Person details
Student_grade	Business process	40	None	Avg grade of each student
department	Master data	4	Department name	Each row is a department information
Office_assignment	Master data	9	None	One row is office assigned to a professor
Course_evaluations.json	Business process	30	None	Course review by student for each courses they took

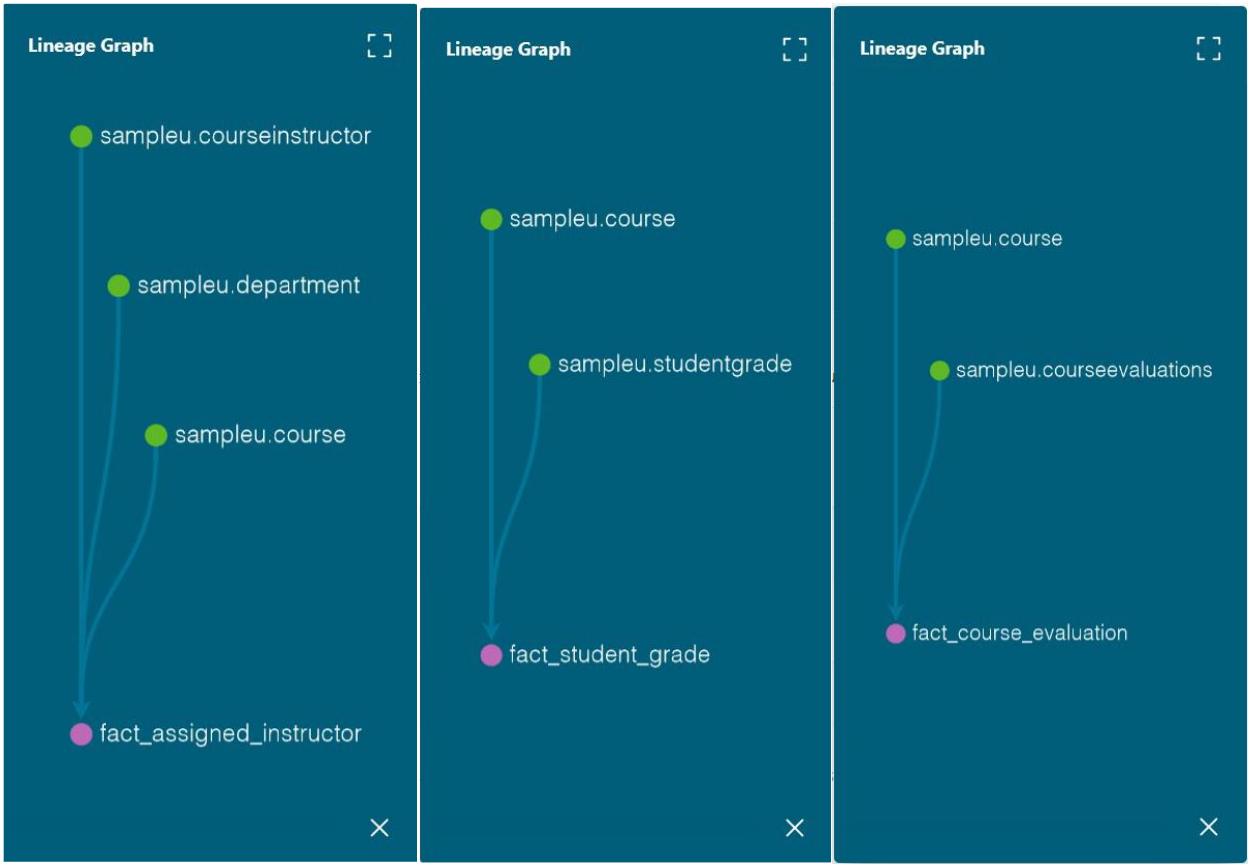
Step 2: Data Extraction and Loading (ELT Process)

Our primary focus was curating data from various sources, including course details, instructor and student profiles, departmental hierarchies, and external feedback from S3. Snowflake facilitated a streamlined Extraction and load process; the result was stagged raw data in snowflake for further processing.

Step 3: Transformations and Analytics (dbt)

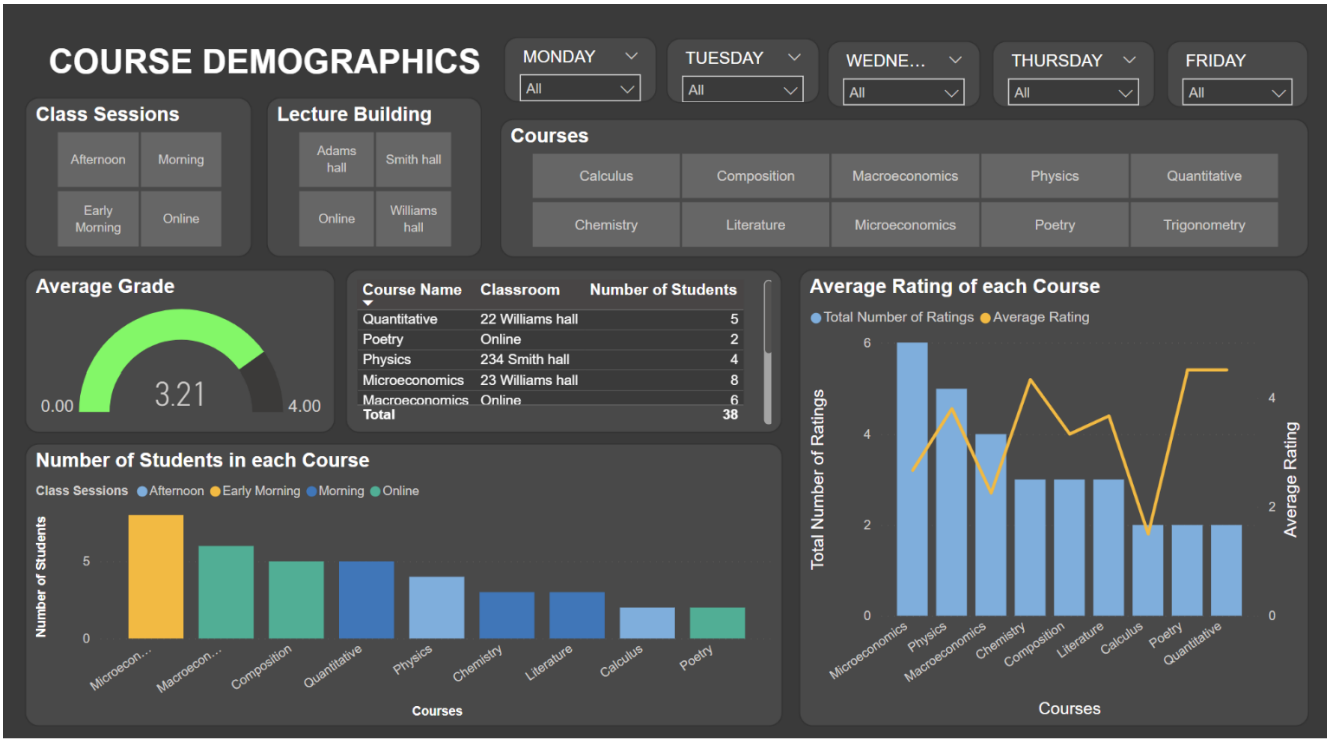
Within dbt, raw data underwent transformative processes, introducing necessary columns and intricate transformations. We calculated average class grade, GPA with the help of credits of a course and grade of student. We also categorized the lecture time column such that it will tell us the time of the day lecture is in(early morning, morning, afternoon). We combined online and onsite courses into the dim_course and introduced the lecture_session column which will define the time of the day for classes as stated above. After that we separated students and instructors

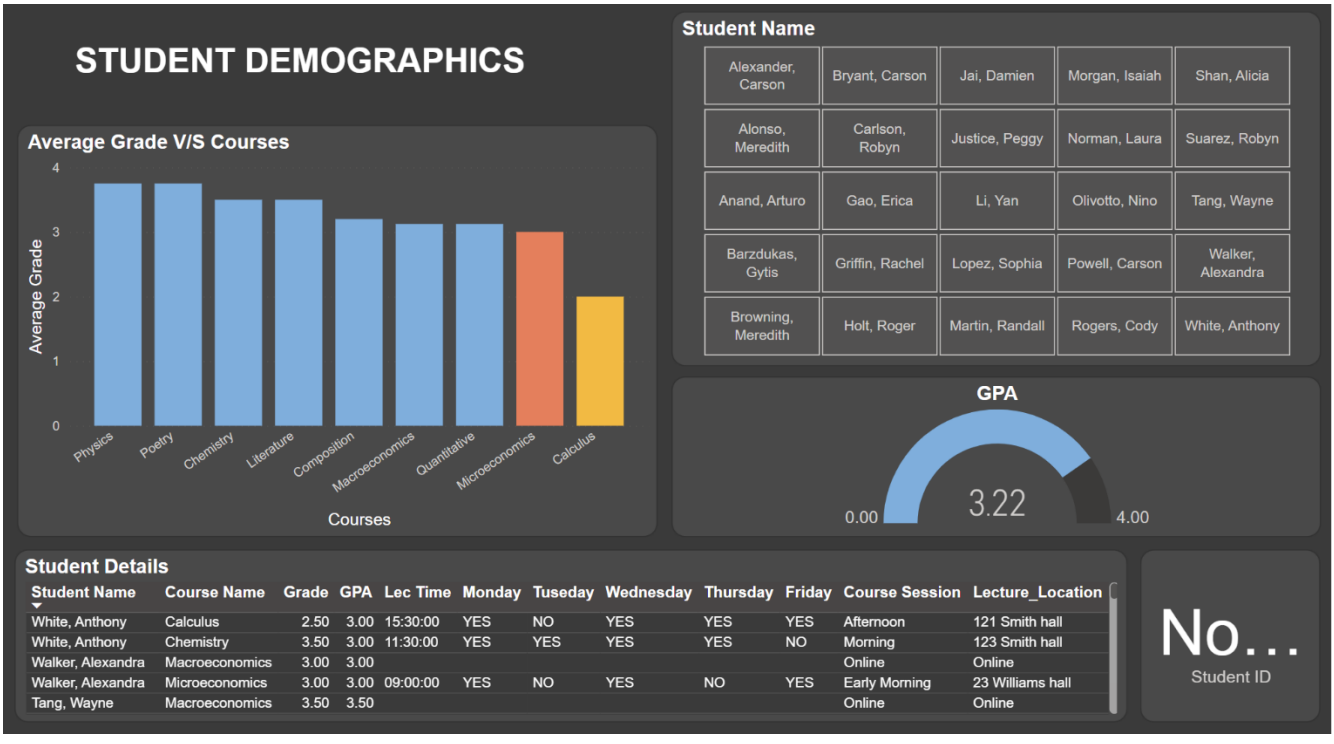
from person table creating two different dimensions for easy and efficient access in future. This refined dataset was prepared for advanced analytics and migrated into analytics database for exploration and PowerBI report generation.

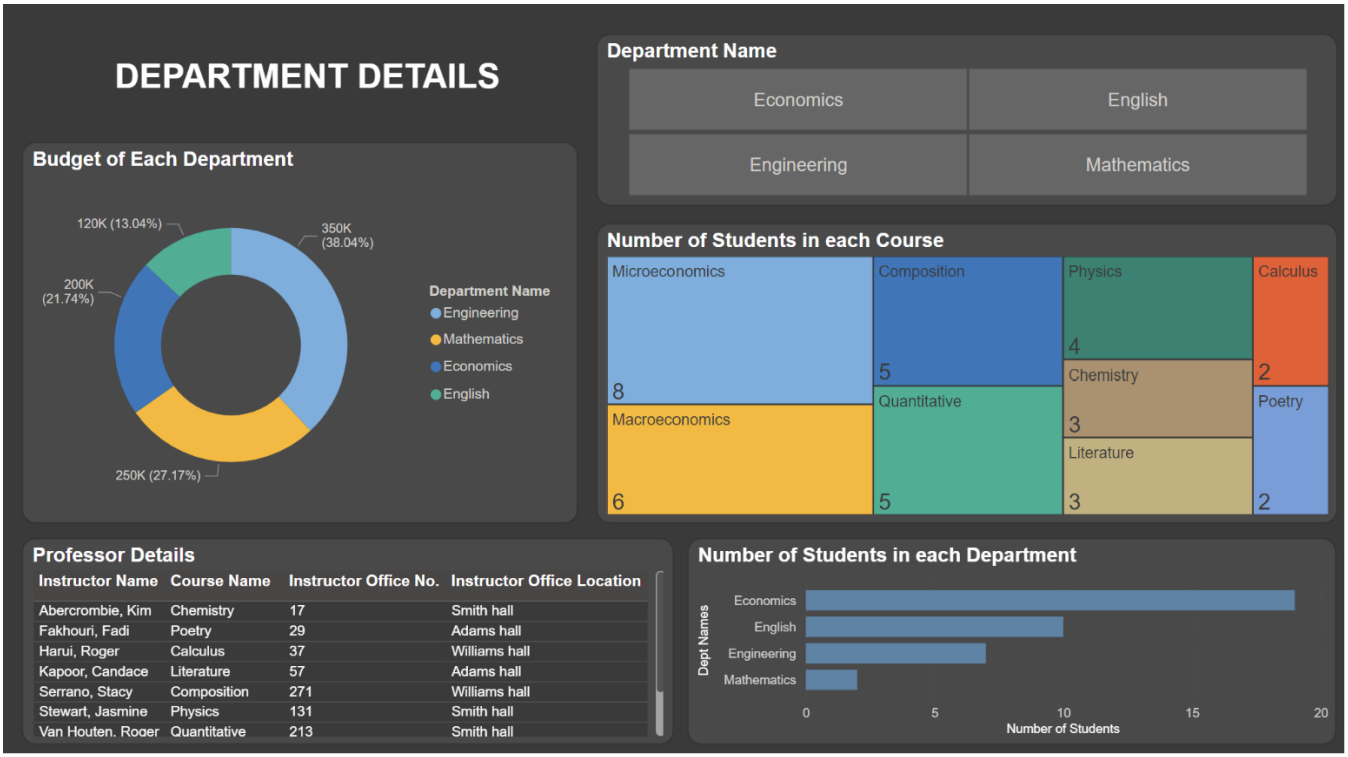


Step 4: Visualization and Analysis (Power BI)

The synergy between dbt and Power BI marked the visualization and analysis phase. The integration allowed us to translate meticulously prepared data into insightful visualizations. For instance, bar graphs on course demographics page provided insights into total course ratings and average ratings of the courses, and lecture sessions. Student performance analysis delved into subject-wise grades and overall academic GPA, linked to individual student IDs. Notably, personalized visualizations for each student illustrated their chosen courses, corresponding GPAs, and course schedules. This individualized perspective not only aids in monitoring academic progress but also enables the design of targeted support and intervention strategies based on each student's unique journey. Such dashboard would be very effective for college academic counselors who can see all the details about student just by selecting student name or ID.


COURSE DEMOGRAPHICS
MONDAY
All
TUESDAY
All
WEDNE...
All
THURSDAY
All
FRIDAY
All
Class Sessions
Afternoon
Morning
Early Morning
Online
Lecture Building
Adams hall
Smith hall
Online
Williams hall
Courses
Calculus
Composition
Macroeconomics
Physics
Quantitative
Chemistry
Literature
Microeconomics
Poetry
Trigonometry
Average Grade
0.00
3.21
4.00
Course Name
Classroom
Number of Students
Quantitative
22 Williams hall
5
Poetry
Online
2
Physics
234 Smith hall
4
Microeconomics
23 Williams hall
8
Macroeconomics
Online
6
Total
38
Average Rating of each Course
Total Number of Ratings
Average Rating
Total Number of Ratings
Average Rating
Microeconomics
Physics
Macroeconomics
Chemistry
Composition
Literature
Calculus
Poetry
Quantitative
Courses
Number of Students in each Course
Class Sessions
Afternoon
Early Morning
Morning
Online
Number of Students
Microecon...
Macroecon...
Composition
Quantitative
Physics
Chemistry
Literature
Calculus
Poetry
Courses


STUDENT DEMOGRAPHICS
Student Name
Alexander, Carson
Bryant, Carson
Jai, Damien
Morgan, Isaiah
Shan, Alicia
Alonso, Meredith
Carlson, Robyn
Justice, Peggy
Norman, Laura
Suarez, Robyn
Anand, Arturo
Gao, Erica
Li, Yan
Olivotto, Nino
Tang, Wayne
Barzdukas, Gytis
Griffin, Rachel
Lopez, Sophia
Powell, Carson
Walker, Alexandra
Browning, Meredith
Holt, Roger
Martin, Randall
Rogers, Cody
White, Anthony
Average Grade V/S Courses
Average Grade
Physics
Poetry
Chemistry
Literature
Composition
Macroeconomics
Quantitative
Microeconomics
Calculus
Courses
GPA
0.00
3.22
4.00
Student Details
Student Name
Course Name
Grade
GPA
Lec Time
Monday
Tuseday
Wednesday
Thursday
Friday
Course Session
Lecture_Location
White, Anthony
Calculus
2.50
3.00
15:30:00
YES
NO
YES
YES
YES
Afternoon
121 Smith hall
White, Anthony
Chemistry
3.50
3.00
11:30:00
YES
YES
YES
YES
NO
Morning
123 Smith hall
Walker, Alexandra
Macroeconomics
3.00
3.00
Online
Online
Walker, Alexandra
Microeconomics
3.00
3.00
09:00:00
YES
NO
YES
NO
YES
Early Morning
23 Williams hall
Tang, Wayne
Macroeconomics
3.50
3.50
Online
Online
No...
Student ID



Conclusion

By combining Snowflake, dbt, and Power BI, our project converged technological tools, providing a holistic understanding of university dynamics. We learned fundamentals behind data warehousing and with this project we were able to create a complete data warehouse workflow. Visualizations acted as windows into course scheduling and student performance, offering actionable insights. This project serves as a foundational platform for ongoing enhancements and advanced analytics, contributing significantly to informed decision-making within the university.