THOMPSON RIVERS UNIVERSITY

Machine Learning and Patient Partner Engagement to Predict the Usage of Home Care Services

By

Robin Teotia

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science in Data Science

KAMLOOPS, BRITISH COLUMBIA

[April, 2022]

Supervisors

Dr. Piper Jackson

Dr. Jabed Tomal

ABSTRACT

This research is a comparative analysis of the application of different machine-learning methods to health care data to predict home care usage in consultation with patient partner involvement. The data used are from the interRAI Home Care assessment after instrument, collected in central British Columbia, Canada. The original data set used contains 837,536 records, gathered from 2010 to 2019, and 423 attributes. The model is developed for predicting the average hours per day usage of home care services in the three weeks following an assessment using different regression and classification methods. For regression, I used multiple linear model, lasso, ridge, decision tree, and ensemble methods, where the last appeared as the most promising. For classification, I used KNN, logistic regression, decision tree, and ensemble methods. Apart from the machine learning algorithms, both patient partners and health care experts participated and provided feedback regarding home care practices and issues. These formed essential elements in designing the research questions, selecting variables, and improving the models. The ensemble methods, namely Random Forests and Bagged trees, are found promising for both regression and classification problems. The Random Forests has achieved the largest R^2 (0.53) in predicting the average hours per day. For classification, the largest accuracy and ROC AUC scores are 0.96 and 0.97 respectively, obtained from the Random Forest and Bagging algorithms.

Key Words: machine learning; healthcare; ensemble methods; Random Forests s; k-nearest neighbors; patient-oriented research; feature selection.

ACKNOWLEDGEMENTS

I am very grateful to Dr. Piper Jackson, Dr. Shannon Freeman, Dr. Jabed Tomal, and Dr. Yan Yan for their invaluable guidance towards the development of this thesis. I would like to thank the BC Support Unit for funding this research, and for investing in advancing data science using a patient-oriented research approach in British Columbia. I would also like to thank Holly Buhler at Interior Health for her support and guidance in working with this data. Finally, I am very grateful for the many contributions and generous feedback about the predictive model from our patient partners: Brent Baker, Ivy Muturi, S. Carl Zanon, Susan Prior, and Grace D. Kramer.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's world, the health problems that humanity is experiencing require immediate care. Health care is a vital area of society. With the advancement of technology and research, healthcare institutions are producing massive amounts of data, requiring proper data management and analysis.

1.1 Health Care Data Management and Analysis

Data collection has become a vital part of every private or public organization. The health care industry is recording data in terms of medical reports, medical history, and medical results of the patients [Dash et al., 2019]. In order to address the health crisis, a proper analysis of the data is needed. This huge amount of data is an untapped wealth of information in

health science that can potentially be harnessed by using machine learning (ML) algorithms by detecting patterns and forecasting. It is essential to draw accurate inferences and information from the analysis of the available data through machine learning to address critical health issues. It is also important to be able to present findings to health authorities and policy makers so that they can find the basis to formulate new policies and deploy new systems and devices in order to ensure the good health of society.

As the Canadian population ages, the health care system will be expected to serve increased demands and expectations, including a higher prevalence of persons living with chronic conditions and a common expectation to support Canadians to live at home as long as possible, the research is mentioned in the report Canadian Institute for Health Information [2017]. In 2015-2016, 6.4% of Canadians (881,800 Canadian households) reported one or more persons in their home had received home care services in the previous year, most often nursing services and personal/home supports, stated by Gilmour [2019]. With the growing number of individuals requiring care and support from community-based home care services and supports, it has become paramount to ensure that resources allocated are able to support Canadians to live and age in the right place at the right time. To develop evidence-based solutions to challenges in the health system and to inform more accurate forecasting of health service demands and utilization, proper organization and uptake of data are needed.

1.2 Project Goal and Patient Partners Contribution

The research forms a part of the goal of Canada's Strategy for Patient-Oriented Research (SPOR) [Canadian Institutes of Health Research, 2019]. Under this, there is active participation of patient partners, researchers, domain experts, healthcare providers, and decision-makers working together to develop a sustainable and accessible healthcare system and bring positive health changes into society. Mentioned in Canadian Institutes of Health Research [2019], the goals of patient-oriented research are:

1. To improve health.
2. To enhance access to the health care system.
3. To ensure the right treatment at right time.
4. To be an active informer of health care.
5. To make pragmatic efforts and contributions to make the Canadian health system effective.

Active patient participation in studies will strengthen the significance of the research and its implementation into policy and practice, resulting in improved and efficient health services and products and eventually boosting Canadians' quality of life and enhancing the Canadian healthcare system. Researchers should work in direct consultation with patient partners, domain experts, researchers, and health care experts in order to ensure they are going in the right direction, which is imperative to accomplish the goals of their project.

1.3 Machine Learning with the Health Data

This research is an exploration of how advanced data science methods can be used to improve health care decision making in our home region, central British Columbia. In particular, I am focusing on predicting transitions in the healthcare needs of older adults in our communities so that healthcare resources will be ready for them when they are needed. Machine learning is a group of statistical methods that use computation to build models from large amounts of data to predict a response variable of interest. Because machine learning methods are capable of providing results that are highly tailored to the characteristics of individual examples (in this case, health records), I expect that they will be useful for predicting the needs of the highly diverse populace.

In this research, Home and Community Care Resident Assessment Instrument (RAI-HC) health care data are used in collaboration with Resident Assessment Instrument (RAI) home care service use data. Patients receiving home health care are authorized with a specified number of hours by the respective health institution, but patients often receive fewer or more hours according to the services or help they need. So the conundrum is finding the perfect allocation of hours that patients need, according to the help and services required as per their health conditions. In order to answer and resolve the above-mentioned issues, I have formulated the following research questions.

1.3.1 Research Questions

1. What is the promising method to predict the number of hours per day on average a client needs in home care?
2. What are good classification methods to classify and dichotomize the number of hours per day on average a client needs in home care?
3. What are the significant features to predict the usage of home care services for a client in the near future?

These research questions have been improved and refined after a thorough consultation with patient partners and domain experts. Keeping these questions in mind and with patient partners' and domain experts' advice, three weeks was identified as the critical time period following an assessment for any issues captured in that assessment to impact home care service use. Thus, the experiments described in this research predict home care service use in the three-week period following a home care assessment. The target is formulated to be the average hours per day used of each service from the start date of the assessment.

Figure 1.1 shows the flow chart of the process followed throughout the experiment. It shows how the data was prepared initially, regression, and classification algorithms were applied following different cross-validation techniques, under different classification scenarios.

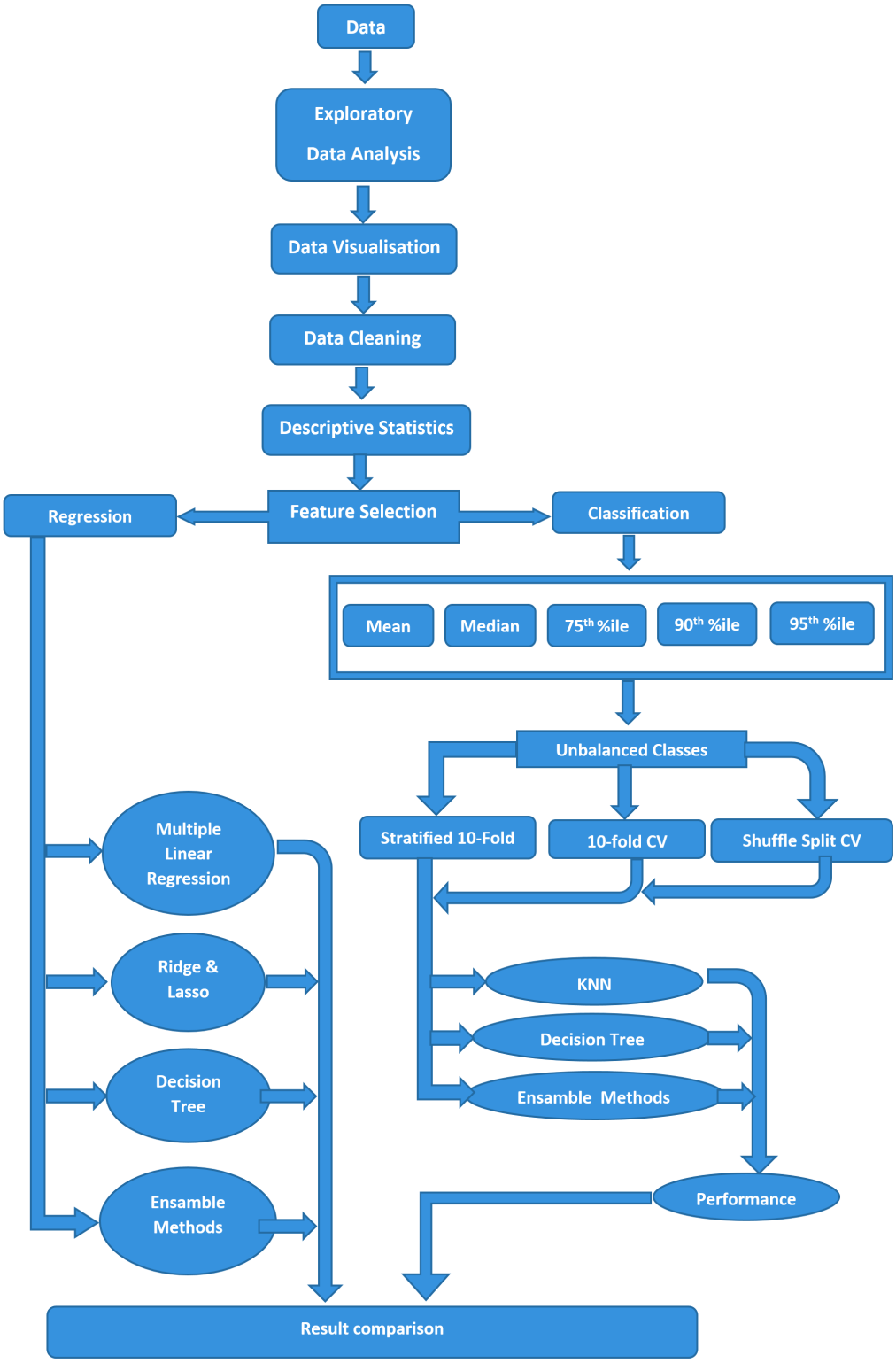

Figure 1.1: Flow chart of the proposed methodologies.

Further, it illustrates the sequence of the different machine learning algorithms and the comparison of results.

The response variable was predicted by using different machine learning regression algorithms like multiple linear regression, Ridge, Lasso, and ensemble techniques. Ensemble methods gave comparatively good results, and in particular, the Random Forests and Bagged Trees provided very promising results.

For classification, the target was dichotomized on the basis of the average, median, 75^{th}, 90^{th}, and 95^{th} percentile values and then different machine learning classification algorithms were deployed to calculate the accuracy of the classification and area under the ROC curve. Among all the applied algorithms, the ensemble methods were again promising to show good classification results. When the mean was used to dichotomize the target and to make classes, the Random Forests gave an accuracy of 0.84 using the 10-fold cross-validation. For higher quantile values, the 90^{th} and 95^{th} percentiles were used to dichotomize the target, it was noticed that the accuracy was greatly enhanced. The Bagged Trees and Random Forests produced promising results with accuracies of 0.92 and 0.95 for 90^{th} percentile and 95^{th} percentiles respectively. For 95^{th} percentiles classification, both Random Forests and the Bagged Trees have the largest ROC AUC score of 0.97. From this, it was concluded that higher quantile values with the interRAI home care assessment (RAI-HC) data set generated comparatively good results when compared to centered values such as mean and median.

This thesis is structured as follows: the next chapter provides some background information on the methods applied in this research and other

relevant concepts. Next, I share information about the data and how it was organized. The adopted approaches for data preparation and experiments are then discussed in a chapter on methodology. Following that, I present my research findings and then discuss some of my observations, discoveries, challenges, and limitations. The thesis report is completed with a conclusion; references and program code are also included at the end.

Chapter 2

Background

The chapter explains the important terms such as home care and patient partner engagement in patient-oriented research. It also covers the past research in the field of machine learning concerning health care. Different machine learning methods, which form the basis of this research, are also explained in this chapter.

2.1 Home Care

Home care incorporates a range of services, such as home assistance, that assist clients in remaining as self-sufficient as possible in their own homes. Community health workers provide home support services to clients who need personal assistance with the activities of daily living (ADL) [Ministry of Health, 2017]. These ADLs include:

- mobilization,
- nutrition,
- baths,
- lifts, and
- grooming and toileting.

Ministry of Health [2017] states that when needed, home support services may include safety maintenance chores in addition to personal assistance. Clean-up, laundry or clothing, and food preparation are examples of these activities. Furthermore, health care workers may be assigned by health care experts to do specialized nursing and rehabilitative activities.

2.2 Health Care and Machine Learning

As a core component of health service provision, there is growing interest in home care as an application field for machine learning, as can be seen in recent scientific literature. Zhu et al. [2014] present three examples (using KNN, lasso regression, and Random Forests) to demonstrate how machine learning can fulfill multiple roles in home care research and clinical decision making, producing both predictions as well as explanatory insights. Cheng et al. [2015] applied lasso regression and Random Forests to find characteristics represented in RAI-HC data that have the potential for predicting the need for rehabilitation services. Jones et al. [2018] worked on predicting emergency department and hospital use by applying ensemble

methods and neural networks with multiple health databases. In their case, gradient boosting was the most effective method. Finally, Veyron et al. [2019] used several machine learning methods on functional status data collected by home care aides to predict emergency department visits, finding that Random Forests was the most effective method. Notably, these last two studies considered both regression and classification methods for predicting the outcomes of interest. As a group, these projects demonstrate that many different machine learning methods have the potential for effective use in home care, and the consequent need to consider and compare multiple methods when developing a new study.

Mohammed et al. [2020] showed how to deal with imbalanced classes in the data set. They applied resampling algorithms on publically available imbalanced data from Kaggle. Their main discovery was that oversampling outperforms undersampling for different classifiers and results in better scores in distinct evaluation matrices, the reason behind it is that the methodology of undersampling leads to the information loss [Mohammed et al., 2020]. In Jeatrakul and Wong [2009], a comparison of different neural network approaches is presented for binary classification. Back propagation neural networks (BPNN), radial basis function neural networks (RBFNN), general regression neural networks (GRNN), probabilistic neural networks (PNN), and complementary neural networks (CMTNN) were the five techniques that were used to compare the classification performance. The comparison is relying on three benchmark data sets taken from the University of California at Irvine's machine learning repository. When compared to strategies used to solve binary classification issues, the results demonstrate that CMTNN generally produces better classification results.

This study is a comparison of various standard machine learning methods and related techniques, examining how they could be applied to consider a real-world problem using complex health data. This included both classification as well as regression algorithms. I will provide a short description of the methods here.

2.3 K-Nearest-Neighbors

It is a machine learning algorithm for regression and classification. The rationale behind KNN is to use distance to locate the most likely value of the target feature. For the motive, the k closest neighbours are found to the example under consideration. In classification, the most popular class among those neighbours is deemed as the predicted class. It is a non-parametric strategy and does not use any parametric distribution. It is a type of classification that entirely relies upon instances, meaning it only takes the available data into consideration with no generalization occurring. It is also known as a lazy learning algorithm since all of the steps and processes of the algorithm are performed during the query (in principle). It means the algorithm does not mandate any preprocessing. It usually works well with data having low dimensionality but its efficacy decreases when the dimensions of the data are large. In cases when the dimensions are large, principal component analysis (PCA) may resolve the issue [Jain, 2021].

KNN uses different metrics to measure distance, such as Euclidean Distance, Manhattan Distance, Chebyshev Distance, Minkowski Distance, and Mahalanobis Distance. For this research, the distance metric used was Euclidean for all of the experiments. Euclidean distance is given by the

formula,

$$D(x, y) = \sqrt{\sum_i |x_i - y_i|^2}.$$

The value of K is optimized using distinct cross-validations.

2.4 Decision Tree and Ensemble Learning

Decision trees are dominant and influential tools for classification and regression. It is a popular machine learning algorithm that is simple and easy to implement. It is also known as a common algorithm and is well-known by the name Classification and Regression Trees (CART). The algorithm has the potential to deal with the data possessing high dimensionality and can work with numerical as well as categorical data. Decision tree models are very easy to comprehend and interpret, hence they are extremely useful and congruous for exploratory data discovery for information. In the process of decision tree formation, recursive division of the data takes place, until the target class variable in each division is as homogeneous as possible.

The performance of decision trees can be enhanced with suitable attribute selection. So, variable selection plays an important role in decision tree construction. A simple decision tree may be unable to classify or predict the target at a depth of 3, 4, or 5. Increasing this depth or using a different combination of trees altogether might give better results. Hence, ensemble methods are other techniques that came into the role.

In the paradigm of **ensemble learning**, numerous estimators are trained to foresee the performance of the models, but the accuracies of

these models are not necessarily good enough on their own. These estimators are classified as weak estimators. However, when these estimators are collectively trained on a model and used together as a group, it generates a robust model with outstanding performance. So, with the technique of ensemble methods, multiple estimators are used to establish a better predictor by aggregating distinct models in a way to make one model with better performance.

There are two essential aspects to be considered: low variance and low bias model for promising prediction. In addition to that, the choice of degree of freedom must be done by assessing the two cases, first deterring high variance and second, maintaining the robustness of the model [Rocca, 2021]. The **bias-variance trade-off** of a model is resolved with the help of ensemble methods. Under ensemble methods, there exist two major types of meta-algorithms that aim at combining weak learners: averaging and boosting.

To serve our purpose and come up with better accuracy, both averaging and boosting strategies are used. In averaging, construction and aggregation of multiple models or classifiers take place. Bagged Trees and Random Forests fall into this category. The methods of Adaboost and Gradient Boosting come under boosting, in which multiple weak models are used collectively and iteratively to generate an improved model. The accuracy scores are recorded to measure the performance.

2.5 Multiple Linear Regression

Multiple linear regression was the first machine learning regression algorithm applied to predict the response variable, the average hours per day of services used by home care clients. The matrix equation of multiple linear equations is:

$$Y = X\beta + \epsilon.$$

Here, Y represents the response vector, X represents the design matrix (containing vectors of predictors along with a column vector of 1s, to accommodate the intercept term) and ϵ represents the error vector. The assumption of the error vector is that it is normally distributed with a mean 0 and a constant variance σ^2 [Abraham and Ledolter, 2006]. Errors are independent and identically distributed, meaning that ϵ_i and ϵ_j have $\text{cor}(\epsilon_i, \epsilon_j) = 0$, for all $i \neq j$. This assumption implies that Y has a normal distribution with mean $X\beta$ and the constant variance σ^2. Y is independent and identically distributed, that is, for any y_i and y_j in Y, the $\text{cor}(y_i, y_j) = 0$, where $i \neq j$ [Abraham and Ledolter, 2006]. The above multiple linear equation can also be written as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_p x_p + \epsilon.$$

Here, β_0 is the intercept and the other β_j are the slopes. The main task is to find these coefficients and put them into this equation to predict the response. The independency of the predictor variables is imperative, meaning that they should not be correlated. That is, one predictor variable can not be written as a linear combination of the other variables.

To check whether the algorithm is better fitted or not, these steps are taken: first, examining the residual vs. fitted plot, and secondly, checking

the normal QQ-plot.

Residual vs. Fitted Plot: In multiple linear regression, the residuals are plotted against the predicted or fitted value $X\hat{\beta}$. For the model to be a good fit, the residuals should not show any discerning patterns against the fitted values [James et al., 2013]. In other words, the error should have the same unknown variance, which is also called a homoscedasticity assumption. The characteristics of good residual vs. fitted plots explained by Cran.R [2021] are:

1. The residuals are randomly distributed around the 0 line, showing a linear relationship.
2. The residuals constitute a roughly horizontal zone around the 0 line, suggesting homogeneity of error variance.
3. There should be no outliers in the residuals.

If there exists a pattern in the residuals, it implies a problem with the linear model. If the residual vs. the fitted plot indicates that there exists a visible pattern or a non-linear association in the data, then a simple approach is to apply a non-linear transformation, including $\log x$, $\sqrt{x}$ and x^2 [James et al., 2013].

Normality of the residuals: The residuals should be normally distributed. The QQ-plot of the residual is a good way to check for any violation of the normality assumption. In this plot, the residuals should fall close to the expected normal quantiles. If the residuals lie away from the diagonal line, there is a violation of the normality assumptions. Histograms are also used to check the normality of the residuals. If the curve drawn

with the help of a histogram follows a normal distribution, the residuals follow the assumptions of normality. The R^2 and the mean squared error are calculated to check the performance of the regression model. The R^2 is given by the formula:

$$R^2 = 1 - \frac{RSS}{TSS}.$$

Where RSS represents the residual sum of squares and TSS represents the total sum of squares. With the addition of variables in the model, the R^2 value tends to increase. To overcome this, the adjusted R^2 is calculated. In addition to that, the mean squared error (MSE) is also calculated to evaluate a model.

2.6 Ridge and Lasso

Ridge and Lasso are popular regression methods to shrink the coefficients of the variables. Ridge regression imposes a penalty on the magnitude of the regression coefficients. As a result, they are forced to decrease. The task is to minimize a penalized residual sum of squares,

$$\hat{\beta}_{ridge} = \underset{\beta}{argmin}\{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{i=1}^{p} \beta_j^2\}.$$

The Lasso regression is analogous to ridge regression but there exists a crucial difference. In ridge, the loss is $\sum_{i=1}^{p} \beta_j^2$; whereas the penalty term in lasso is $\sum_{i=1}^{p} |\beta_j|$. The lasso equation is given as

$$\hat{\beta}_{lasso} = \underset{\beta}{argmin}\{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{i=1}^{p} |\beta_j|\}.$$

Here, n is the number of data values, and p is the number of predictors.

In Lasso, some of the predictors may be penalized to zero. Therefore, Lasso reduces the size of the set of variables [Hastie et al., 2001].

2.7 Cross-Validation (CV)

In modern statistics, data partitioning is indispensable for evaluating a model. The process involves recursively partitioning data into training and test samples. Furthermore, fitting a model on the training sample and evaluating it on the test sample. The main goal is to evaluate the test error rate, which is only possible if there is test data. Therefore, a reasonable solution is to calculate the test error rate from the available training error rate. But in general, the training error tends to underestimate the test error.

This problem is addressed by using a validation set. Under this approach, the observations in a data set are randomly divided into a training set and a validation set. The model is then fitted on the training set, which is used to predict the response in the validation set. From the validation set, the error is eventually calculated. The resulting validation set provides a reasonable estimate of the test error rate.

The validation set approach has 2 drawbacks:

1. The observations used in the training set are different from those in the validation set. As a result, the validation error rate can vary largely.

2. The model is trained only on a subset of observations, i.e., fewer

observations, and may perform poorly. This may result in overestimating the test error rate.

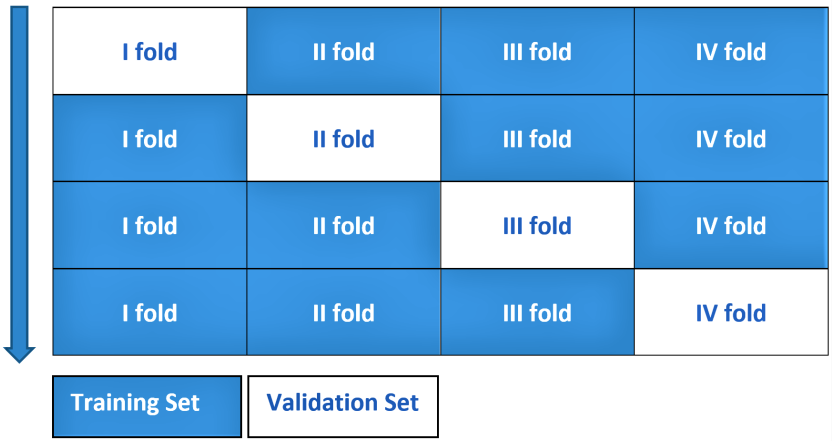

Figure 2.1: K-fold cross-validation (k = 4).

These issues are resolved by adopting the methodology of cross-validation. In this work, three methods of cross-validation were used: K-fold cross-validation, Shuffle split cross-validation and Stratified k-fold cross-validation.

In K-fold cross-validation, the entire set of observations is randomly divided into k parts (or folds) of equal size. Figure 2.1 shows how the process of 4-fold cross-validation is executed.

When the first fold is considered as the validation set, the model is fitted on the remaining k-1 training folds. The trained model is then used to calculate the MSE on the held-out fold (i.e., the first fold). In the second iteration, the second fold is used for validation when the remaining folds are used for training. The process continues for a total of k times. A key aspect to note is that in each iteration, $(1/k)^{th}$ portion of the data is used for

validation, and $(k - 1/k)^{th}$ proportion is used for training. The process provides k estimates of the test error such as MSE_1, MSE_2, ..., MSE_k. The cross-validation estimate [James et al., 2013] is given by:

$$CV_{k-fold} = \frac{\sum_i^k (MSE_i)}{k}.$$

The process of cross-validation involves randomness in splitting the data values, which is essential to evaluate the model performance and testing. James et al. [2013] shows that CV is one of the best techniques to prevent over-fitting, especially when the data size is small. CV also offers several approaches to handle the issue of imbalanced classes.

Shuffle split cross-validation method draws training and test sets randomly instead of forming folds. The technique is very useful for large data sets and often appears computationally feasible. It is also known as Monte Carlo cross-validation [James et al., 2013]. The number of iterations performed is decided by the experimenter, and the results are averaged over the number of iterations. The percentage of data in the training and test splits is independent of the number of iterations. This CV is not particularly helpful while working with an imbalanced data set.

Stratified k-fold cross-validation technique is useful for rare-class classification. The stratified CV maintains target class proportions. This technique ensures that the proportion of classes is balanced in each fold. In each fold, it maintains the distribution (mean, variance, and among others) of the original data and is beneficial over k-fold cross-validation [Singh, 2022]. Stratified CV is very helpful for imbalance class classification. Figure 2.2 illustrates the stratified k-fold cross-validation.

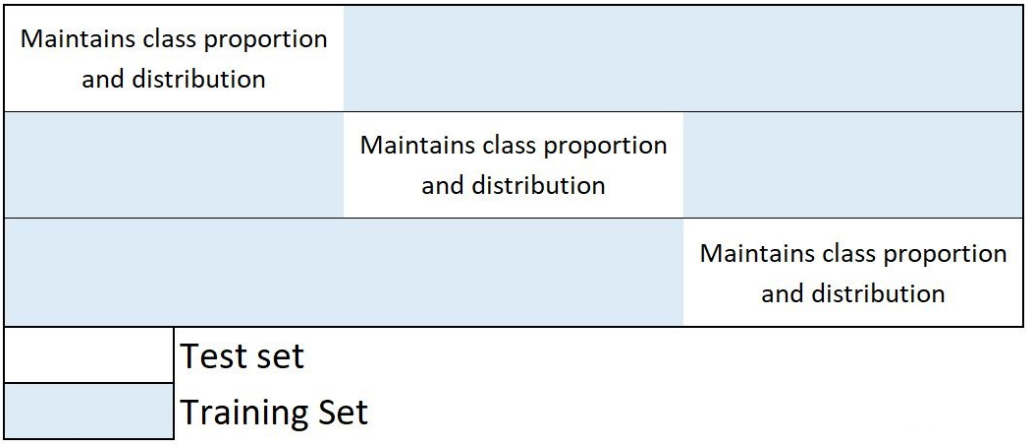

Figure 2.2: Stratified k-fold cross-validation.

2.8 Quantiles and Percentiles

We used different quantiles to split the data for classification. For the K^{th} percentile, the lower portion contains k% of the data while the upper portion contains the rest of the data (100-k)% [Triangles, 2015].

Tackling the imbalanced class problem is of importance while training a machine learning algorithm. The classification problem is considered to be imbalanced when the distribution of the training data is skewed. In such conditions, a classifier is usually biased towards the majority class, and the machine learning algorithms can fail to detect the minority class.

2.9 Random Sampling

Random sampling is another technique to tackle imbalanced class problems. Two strategies are undersampling and oversampling. Figure 2.3 illustrates the techniques of oversampling and undersampling.
Undersampling is the process in which the samples of the majority class are reduced.

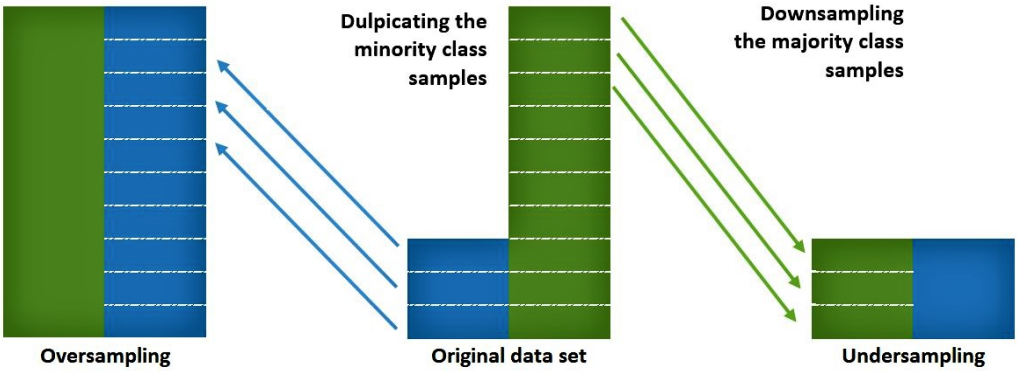

Figure 2.3: Over-sampling and under-sampling.

Oversampling is the process in which the samples of the minority class are duplicated. This method also has some weaknesses, as oversampling can cause overfitting, while undersampling can result in the loss of information [Kumar, 2020]. This random sampling is also known as the naïve technique because when it works, it does not have any assumption over the data [Kumar, 2020].

2.10 Evaluation Metrics

To evaluate the performance of the model for classification, confusion

matrix, and ROC AUC curve are often used. For regression, the R^2 value and mean squared error are usually calculated for evaluation.

2.10.1 Confusion Matrix

A Confusion Matrix is used to evaluate the results of binary classification. The structure and the functionality of the confusion matrix are elaborated in Figure 2.4.

A confusion matrix provides a good explanation between the predicted values and the actual values. It is also known as the error matrix. Its layout helps in depicting the performance of the applied machine learning algorithm. It has two dimensions: the actual value of an example and the predicted value by the algorithm. For binary classification, this results in four possibilities: an example is either positive or negative, and it can either be correctly or incorrectly classified. The positive cases are represented as P, whereas the negative cases are given by N. The total population is defined as P+N. Here, the columns define the actual condition, whereas the rows explain the predicted condition. These are the metrics that are calculated from the confusion matrix.

True Positive (TP): an algorithm predicts a positive result, which is actually positive. It means that the test correctly detected the presence of a trait or condition.

True Negative (TN): an algorithm predicts a negative result, which is actually negative. It means that the test correctly detected the absence of a trait or condition.

False Positive (FP): an algorithm predicts a positive result that is

actually negative. It means that the test falsely detected the presence of a trait or condition.

False Negative (FN): an algorithm predicts a negative result that is actually positive. It means that the test falsely detected the absence of a trait or condition.

		ACTUAL VALUES	
		POSITIVE(1)	NEGATIVE(0)
PREDICTED VALUES	POSITIVE(1)	TRUE POSITIVE	FALSE POSITIVE TYPE I ERROR
	NEGATIVE(0)	FALSE NEGATIVE TYPE II ERROR	TRUE NEGATIVE

Figure 2.4: Confusion matrix for the binary classification problem.

A false positive is also known as type I error, which is defined as rejecting true null hypothesis. A false negative is also known as type II error, which is defined as accepting a false null hypothesis. There are also some further statistical metrics which are used to evaluate the performance of a classifier [Fawcett, 2006]:

Sensitivity/recall/true positive rate (TPR) is the proportion of the positive class that was correctly classified.

$$\frac{TP}{P} = \frac{TP}{TP + FN}.$$

Specificity/ selectivity/ true negative rate (TNR) is the proportion of the negative class that was correctly classified.

$$\frac{TN}{N} = \frac{TN}{TN + FP}.$$

Precision or positive predictive value (PPV) is the ratio of correctly predicted positive classes to the total predicted positive classes.

$$\frac{TP}{TP + FP}.$$

Accuracy(ACC) is the ratio of correct predictions to the total number of predictions.

$$\frac{TP + TN}{P + N}.$$

F1-Score is the harmonic mean of precision and sensitivity.

$$\frac{2Precision * Sensitivity}{Precision + Sensitivity}.$$

2.10.2 ROC AUC Curve

The ROC AUC curve is an assessment metric for classification at various discrimination thresholds [Lars et al., 2011]. Receiver Operator Characteristic (ROC) is the probability curve, and AUC is the area under the ROC curve that estimates the degree of separability. It indicates how well the model can distinguish the two classes. The higher the AUC, the better the model [Narkhede, 2022].

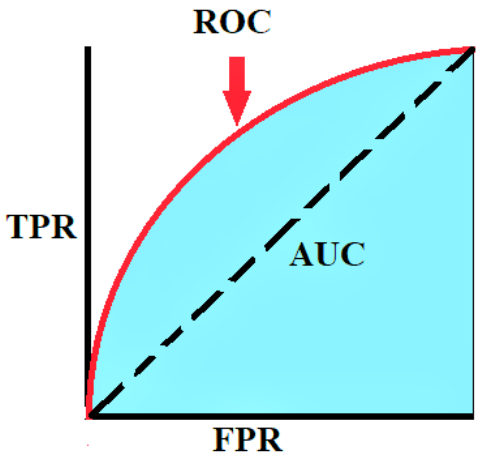

Figure 2.5: ROC AUC curve. The x-axis represents the true positive rate (TPR) while the y-axis denotes the false positive rate (FPR). The blue area under the red curve (ROC) is AUC.

If the AUC is 1, it means the model is excellent and able to classify classes perfectly. A poor model has an AUC score close to 0. A 0 AUC score reciprocates the results, it predicts class 1 as class 2 and vice-versa. A model that has a 0.5 AUC score is not capable of classifying the classes. To calculate AUC, the True Positive Rate (TPR)/Sensitivity is plotted against the False Positive Rate (FPR)/(1-Specificity), where TPR is plotted on the y-axis and the FPR is plotted on the x-axis (Figure 2.5).

2.10.3 R^2 and MSE

The coefficient of determination is denoted by R^2. The R^2 value is the statistical measure that computes the percentage of variance in the

dependent variable around the mean that is explained by the model.

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}.$$

Here, y_i is the response value and $\hat{y}_i$ is the predicted value. $\bar{y}$ is the mean of the response variable and n is the number of data points [Lars et al., 2011].

The Mean squared error (MSE) provides a measurement of how far the predictions are from the actual values on average.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

Both R^2 and MSE are used to check the performance of the regression model.

Chapter 3

Data

I used home care data collected from 2010 to 2019 in the Interior Health Region of British Columbia, Canada. This includes two data sets. The first is InterRAI Home Care assessment data (RAI-HC), which is used by health professionals to record the current status of a home care client. The portion of the data shared consists of 837,536 records, each with 423 variables, which are mostly categorical in nature. The variables cover many aspects of a home care client's characteristics and statuses, such as health conditions, activities of daily living (ADL), independent activities of daily living (IADL), availability of both formal and informal caregiving, mood, and socialization. The second data set is Service Data which covers home care usage by clients. It has 8 attributes. This project was a collaboration with researchers at UNBC. The UNBC Research Ethics Board (REB) determined that REB approval was not necessary for this project because it was entirely secondary data that had been previously linked and de-identified by the health authorities before storing it on a secure UNBC

server.

The primary values of interest to my work are (1) hours of service allocated to a client after assessment, and (2) actual hours of service used by a client. While a perfect assessment would result in these two numbers always being the same for a given client, in practice they can differ greatly depending on whether a client's actual needs over time were greater or lesser than when assessed. For this particular phase of the project, I focused on (2), the actual hours of service used.

Based on input from healthcare experts and patient partners, I identified the three weeks following an assessment as the critical time period of interest for prediction. For each assessment, I calculated the average hours per day of home care service use following the assessment. If a course of home care service began before the assessment and/or was completed after the three-week time limit, only the portion that fell within the three-week time period was included. The mean value for all assessments was 0.79 hours per day. For regression methods, predicting such a value in the target feature was the goal. For classification, the task was to split the service use into two categories: high service users and low service users.

3.1 Basis for Splitting the Target

There are multiple bases used to split the target into categories, such as the mean, median, 75^{th}, 90^{th}, and 95^{th} percentiles.

High: assessment of home care clients above the threshold, i.e., their average hours of home care service use per day was greater than or equal to

the threshold.

Low: assessment of home care clients below the threshold, i.e., the remainder of the records that did not fall into the High category.

When the mean was used to dichotomize the target, It did not show a balanced distribution (Figure 3.1). Among the two classes, the ratio of Low to High was approximately 5:3. So cross-validation and balancing of the classes were employed before training the model.

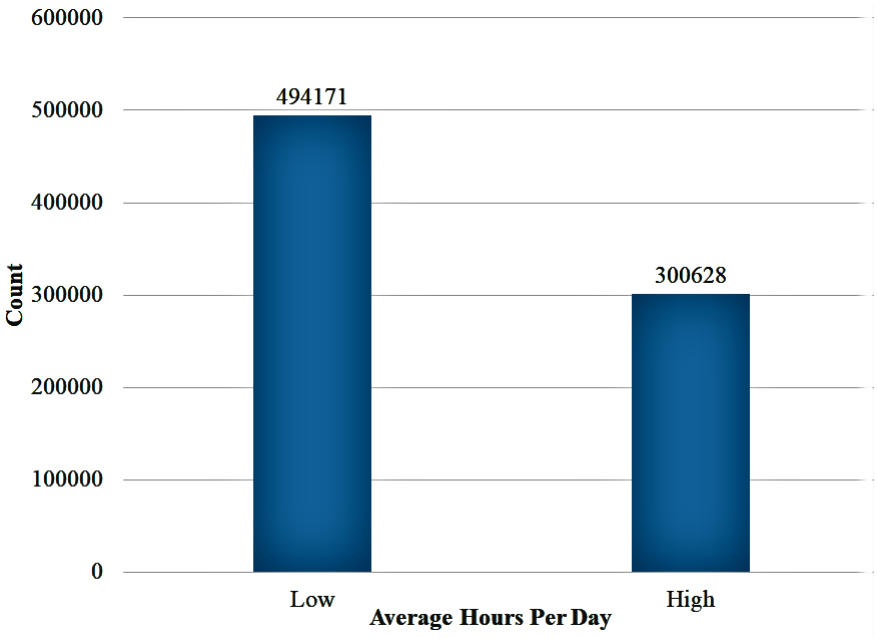

Figure 3.1: Formation of classes when the response variable "Average Hours Per Day" is dichotomized using mean ("Low" is below the mean; "High" is above the mean).

The boxplot in Figure 3.2 gives the information about the target and the respective outlier values. After plotting the histogram (3.3), it was noticed that the data is right-skewed and most of the hours in the response

variable are having very small quantitative values. Thus, almost all of the records lie close to the left corner of the histogram.

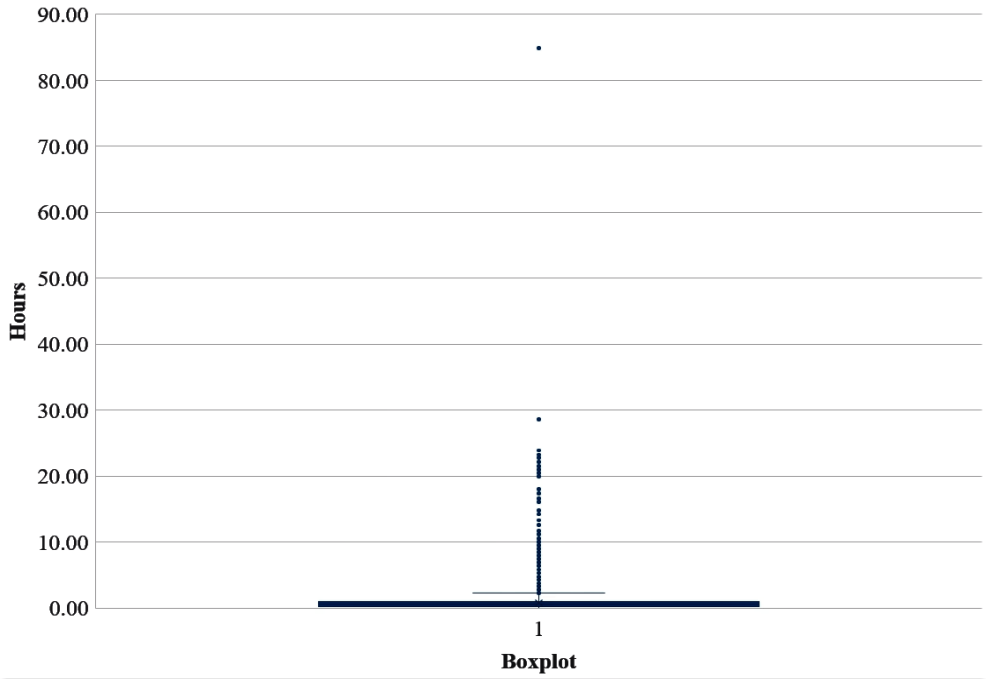

Figure 3.2: Boxplot of the response variable (Average Hours Per Day). Note the outliers which are more than 24 hours (per day).

After reformulating the histogram to focus on the small counts of hours on the x-axis as shown in figure 3.4, I observed that the response variable had values overwhelmingly below 5 hours.

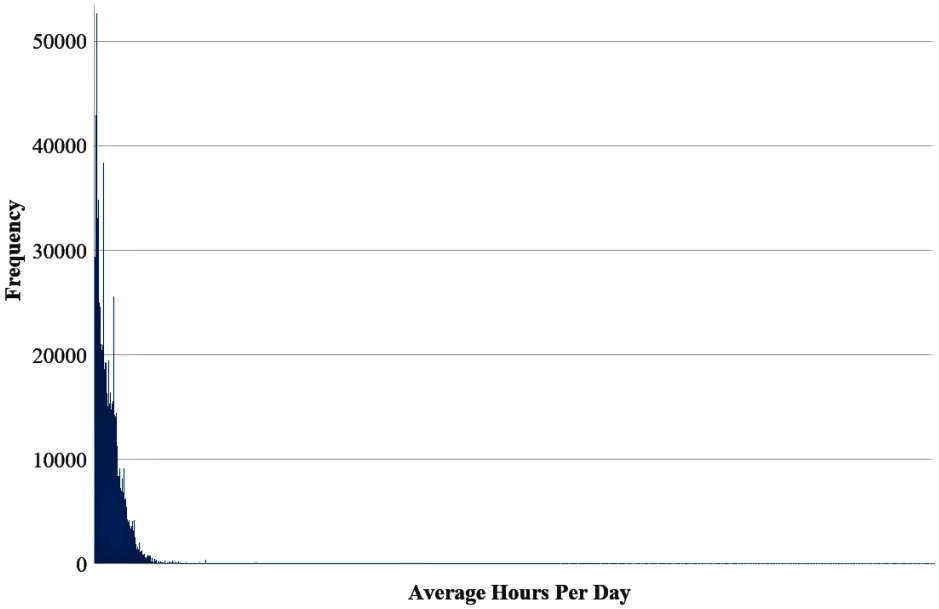

Figure 3.3: Histogram of the response variable (Average Hours Per Day).

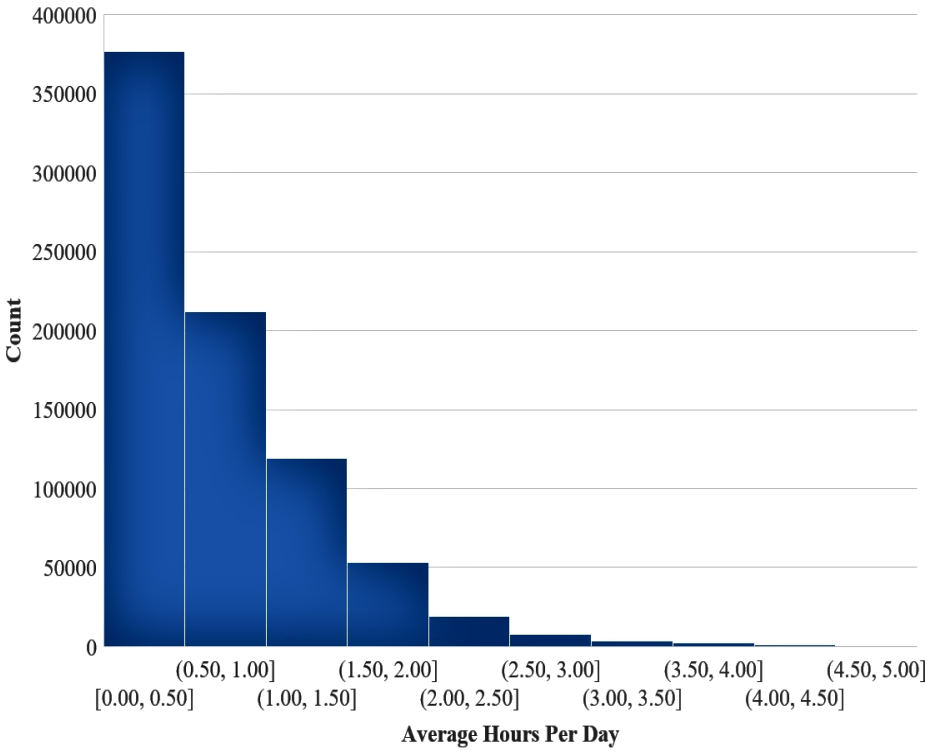

Figure 3.4: Histogram of the response variable after zoomed in.

The table 3.1 shows the information of the target, the mean value of the response variable is 0.79 and the median value is 0.55. And it is evident from the skewness value of 14.01 that the distribution is right skewed. The largest value is 84.92 average hours per day which is clearly an error (because there are only 24 hours in a day). I deleted 5 rows with average hours of uses of home care services above 24 hours.

Table 3.1: Summary of the response variable

Mean	0.79
Standard Error	0.00
Median	0.55
Mode	0.50
Standard Deviation	1.17
Sample Variance	1.37
Kurtosis	485.36
Skewness	14.01
Range	84.92
Minimum	0.00
Maximum	84.92
Sum	125635.87
count	159693
Largest(1)	84.92
Smallest(1)	0.00
Confidence Level(95.0%)	0.01

When the target was divided into classes by using 90^{th} percentile value (1.55 average hours per day), the class variable did not have a balanced

distribution.

The pattern of the targeted bins is visualized in Figure 3.5 is on the basis of the 90th percentile (1.59 Avg. Hrs./Day). Among the two classes, the ratio of Low to High was approximately 7:1. So cross-validation and balancing of the classes were used before training the model. Similarly, the 95th percentile (2.0 Avg. Hrs/Day) was used to split the class and follow the same process.

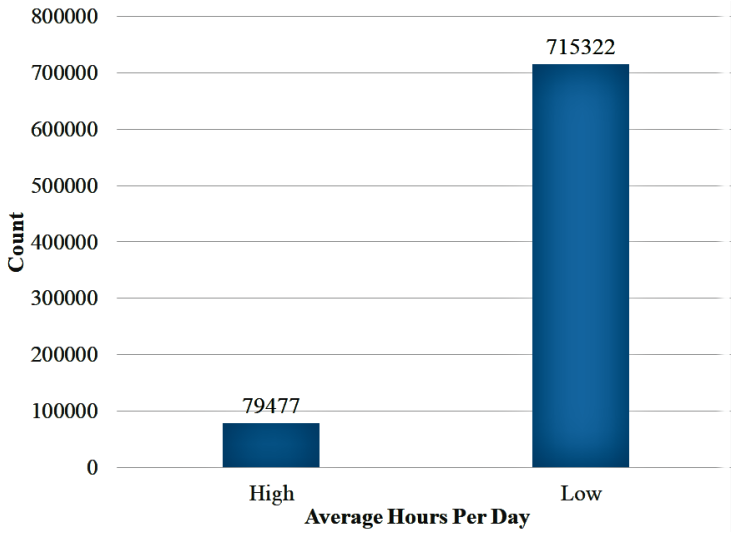

Figure 3.5: Formation of classes when the response variable "Average Hours Per Day" is dichotomized using 90^{th} percentile ("Low" is below the 90^{th} percentile; "High" is above the 90^{th} percentile).

3.2 Data Cleaning and Preparation

The Data set has 132 description variables out of 423 given variables.

291 features are left after the removal of 132 description variables. Finally, 61 variables are selected through the Recursive Feature Elimination method with Random Forests, correlation technique, and the regularisation method of Lasso. The Flow chart 3.6 details the information about the data cleaning and preparation.

The initial number of rows in the data is 837,536. After removing the null value, the resulting rows are 794,804. There exist 5 outliers in the response variable "Average hours Per Day". After removing these outliers, the final number of rows are 794,799.

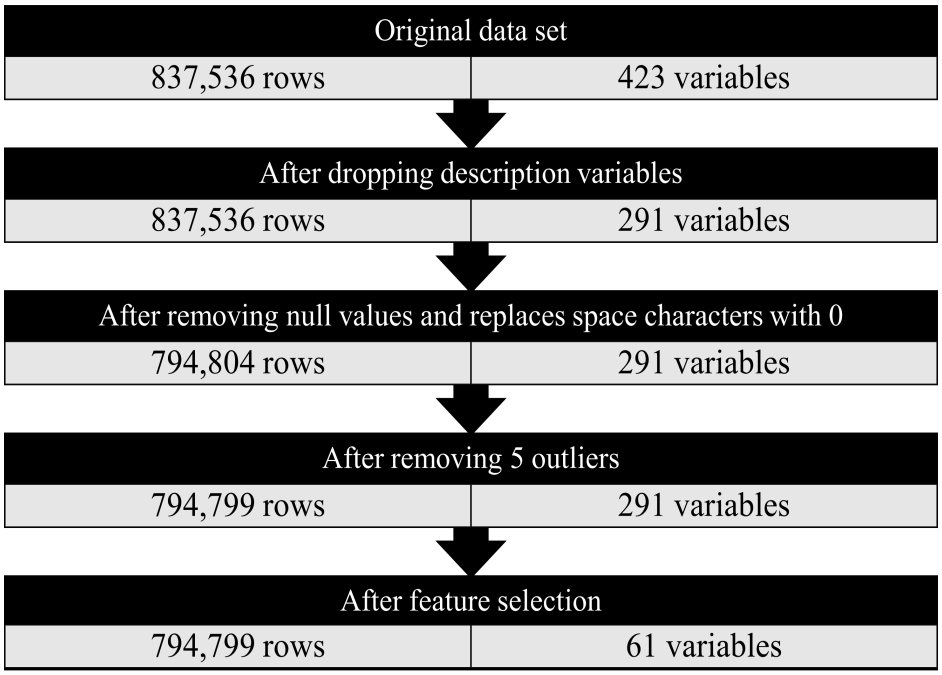

Figure 3.6: Flow chart of the data cleaning and preparation.

Chapter 4

Methodology

In this chapter, connection to the database, exploratory data analysis, and the relevant machine learning methodologies are explained.

4.1 Environment Setup for Acquiring and Processing Data

To accommodate the large data and computation, the main programming language that was considered was Python. Python is a free and open-source language. Its simplicity does not limit its functional abilities. It is a high-level programming language with an enormous existing community. In this project, the Python libraries such as Numpy, Pandas, Matplotlib, Keras, Scikit-Learn, and others were used. In the project, the database was accessed through Microsoft Structured Query Language (MS-SQL). MS-SQL was used to maintain, modify, and

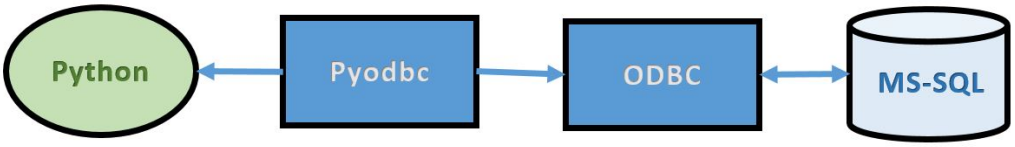

Figure 4.1: Connection of Python with MS-SQL.

manipulate the relational database.

MS-SQL was used to query the various database tables, manipulate the data, and join tables. After this, the main task was to connect the Python code with MS-SQL to access the data for computation and analysis. The task was completed by using the pyodbc module. Pyodbc is a free Python package that makes it easy to connect to ODBC databases (Figure 4.1). It provides a quick way to link Python programs to data sources using an ODBC driver. The target (Average Hours Per Day) in the research was calculated by following consecutive steps of querying the data. Joining the Home Care table with the Service data table, and then using the column calculations over the joined and resulting table.

4.2 Data Wrangling, Visualization, and Exploratory Analysis

To understand the data, viewing the patterns and doing the statistical analysis are very helpful. The process of sorting the massive data set and making it easily accessible for analysis falls under the topic of data wrangling. The provided data set was very large, so data wrangling was

adopted to make the data readily available for use. This was done by converting the data into a data frame and making interpretation easily assessable through the joining of relevant tables. The data were visualized using the Python libraries Matplotlib and Seaborn. Here, the plot of the response variable clearly depicts that it was heavily right-skewed with skewness of 14.01. The box plot provided a clear picture that there were some outliers. Moreover, I also noticed that some entries were erroneous as they did not follow the standard pattern. I contemplate that these entries might have been wrongly entered into the database by human or system error. Then, the rules of exploratory data analysis (EDA) were employed to see what the data revealed statistically. Here, attention was paid to the types of the variables. Some variables are quantitative in nature, whereas some are categorical. After producing a statistical summary of the data, the measurements of the central tendencies and the outliers were noted for quantitative variables, whereas the class count was noted for categorical variables. This provides a clear view of the distribution of the data and how balanced (or imbalanced) the classes are. The EDA reconfirmed the distribution of the response variable and its skewness, which were earlier noticed through the data visualization.

4.3 Data Preparation and Cleaning

After noticing that there are outliers in the data, the vital task is to remove the outliers. Also, some of the entries in the data had null values, so the rows with null entries were removed. In the process of data wrangling and visualization, it was noticed that the data had some missing values.

These were not actually missing but simply a single space character (‘ ’) used to represent a 0. This small but prevalent issue was a challenge to identify. However, I overcame it by replacing these space entries with zeroes. For this, I confirmed with the domain expert and the patient partners that this was an appropriate solution.

4.4 Dichotomizing the Response Variable using Quantiles and Percentiles

To understand the distribution of the target, I used boxplots and histograms. Boxplots gave complete information about the median, different quantile values, and outliers. As shown in Figure 3.2, the largest outlier value is 84.92, which is erroneous.

With the help of the histogram (Figure 3.3), it became evident that the data was highly right-skewed with a skewness value of 14.01, where mode $<$ median $<$ mean. After noticing that there were five values that were outliers in the target, it became important to remove these outliers.

With expert advice, I used values up to 24 hours of the target. Figure 4.2 and 4.3 show the histogram and the boxplot of the target with no outliers.

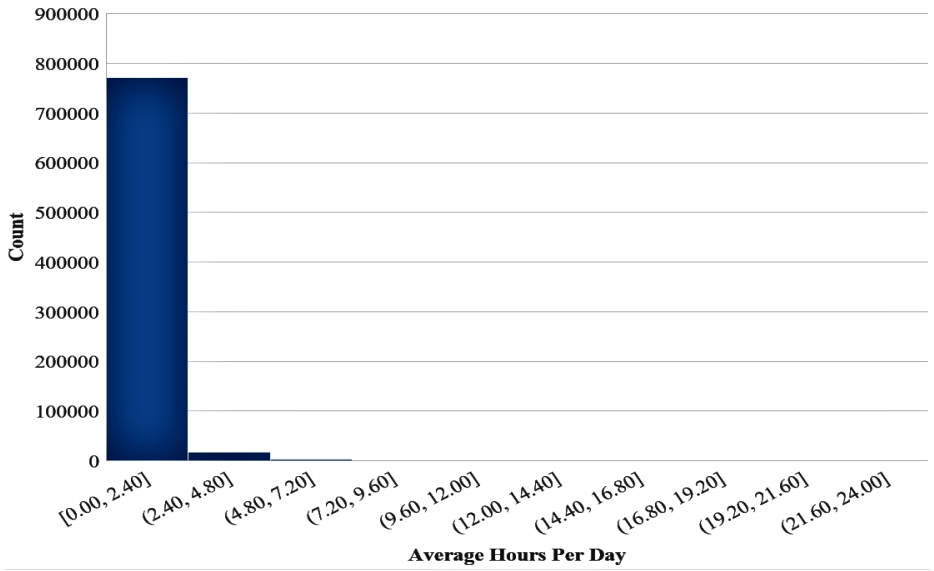

Figure 4.2: Distribution of the response variable after removing outliers.

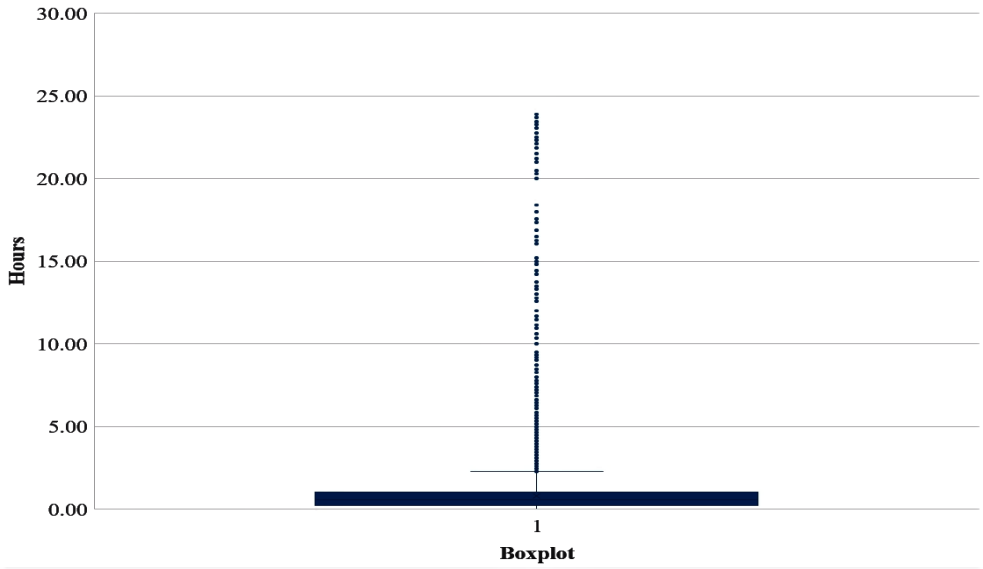

Figure 4.3: Boxplot of the response variable after removing outliers.

Table 4.1: Values after and before removing outliers from the response variable "Average Hours Per Day" when $0 \leq AverageHoursPerDay \leq 24$.

Statistical Measures	Values After Removing Outliers	Original Values
Mean	0.79	0.79
Median	0.55	0.55
75^{th} percentile	1.04	1.04
90^{th} percentile	1.59	2.05
95^{th} percentile	2.0	2.50

Some of the statistical functions were changed after removing the outliers from the target whereas, the mean, median, and 75^{th} percentile values remained unchanged. Table 4.1 shows the new values of the statistical measures after removing outliers. All the values are in units of Average Hours/ Day.

4.5 Encoding of the Data:

The majority of the variables in the data set are categorical in nature. Some categorical variables have six levels, whereas some have four. For example, the variables: AllOtherRespiratoryTreatments, MedicationsbyInjection, ExerciseTherapy, OccupationalTherapy, MedicalAlertBraceletorElectronicSecurityAlert, DayCentre, SkinTreatment, etc., are categorical in nature with four levels defined as:

0 = Not applicable

1 = Scheduled, full adherence as prescribed

2 = Scheduled, partial adherence

3 = Scheduled, not received

While using a qualitative variable to run any machine learning algorithm, there is a possibility to mislead the training of the data. For instance, if the algorithm of linear regression is applied, there might be a chance that the relation is linear for one level but quadratic or cubic for other levels [James et al., 2013]. This will result in fallacious training of the algorithm, which may affect the prediction. To overcome this issue of erroneous learning, the concept of one-hot encoding is introduced. In this, if a categorical variable has 2 levels, then a binary variable is created for each possible value. If there are 3 levels, then 3 binary variables are formed. To perform this task, the "OneHotEncoder" module of the Python sklearn library was used [Lars et al., 2011]. The process generates a dummy variable for each categorical value and stores the results in a sparse matrix. By default, the encoder automatically generates dummy variables based on the unique values of each of the categorical variables.

4.6 The Process of Feature Selection

The original data had 423 variables. It was imperative to do variable selection to find the predictors that were good in predicting the response variable. There are several methods available in machine learning which help in the process of selecting the important features. To select the features, I used the correlation technique, recursive feature elimination, and the regularisation method of lasso [Lars et al., 2011].

4.6.1 Finding Independent Variables

The work on this research began with multiple linear regression. With the provided 423 variables, it was necessary to identify which variables were linearly independent. Hence, the correlation was calculated for the whole design matrix. Only a single variable was selected from the group of highly correlated variables. This process provided a set of independent variables. The variables that possessed a high correlation with the response in this set were selected. This ultimately brought down the size of the provided features.

4.6.2 Recursive Feature Elimination with Random Forests

Recursive feature elimination minimizes the model complexity by deleting variables one by one until the number of features remaining is ideal. This technique is provided by the scikit learn library of Python [Lars et al., 2011]. I used RFE with Random Forests to select features. The reason for using Random Forests was that the R^2 was large as compared to other methods. All regression algorithms possess feature weights or coefficients which multiply with their respective feature value in order to predict a response. The same goes for Random Forests. The weights or coefficients received after training the algorithm are sorted in order. To reduce the model complexity, the values of the weights that were close to zero were eliminated by introducing a threshold with the help of the recursive feature elimination method since their extremely low value (close to zero) contributed a little to the model [Tuychiev, 2021].

4.6.3 Feature Selection with Regularization Method of Lasso

Lasso is a regularisation method. The penalty term used in the cost function is $\sum_{j=1}^{p} |\beta_j|$, where β_j, represents the coefficients of the p features [Hastie et al., 2001]. There exists a term λ (tuning parameter), which is known as the learning rate. If the value of the λ is small, the cost function behaves analogously to MSE. In this case, it reduces the effect of regularization. However, if the value of the λ is large, it imposes the regularisation effect, reducing some of the coefficients to zero. After training the data using the lasso algorithm, it assigns some coefficients (βs) to the regression equation. The values of these coefficients directly affect the prediction of the response. Hence, a threshold value for βs was chosen. Thus, only the variables with values that were larger than the threshold value were kept in order to reduce the model complexity. In this way, the Lasso method helped in selecting the variables.

Further, to cross verify the results, the domain expert and patient partners' advice was taken in deciding whether the selected variables were important or not from an application perspective. The mentioned steps help in selecting the 61 features out of the total provided features.

4.7 Machine Learning Cross-Validation Methods

Cross-validation provides several methods to improve the issue of

over-fitting when the classes are imbalanced. K-fold, Stratified K-fold, and Shuffle-Split cross-validation techniques were used for this work. While dividing the data into training and test sets, the ratio of 80:20 or 90:10 of the data was used. The k-Fold cross-validation was used with k value 10 in order to balance the bias-variance trade-off of the data [James et al., 2013].

4.8 Machine Learning for Regression Problems

To predict the target variable, multiple linear regression was first used. After that, the shrinkage methods of Ridge and Lasso were employed, followed by the Decision Tree and Ensemble methods. A decision tree uses only one variable to split the whole data at the root node. It means that it is only giving importance to the 2 or 3 variables that are at the top nodes to split the whole data. Thus, it undermines the importance of other variables when the variable size is large. Moreover, visualizing the decision tree with large depth is complex when the features are large in number. In order to improve that issue, the approach of ensemble methods was applied. To execute all the mentioned algorithms, the scikit-learn library of Python was used.

4.9 Machine Learning for Classification Problems

Machine learning methods KNN, decision trees, logistic regression, and ensemble methods were used for classification. I observed that the KNN algorithm suffered from the curse of dimensionality. It was computationally inefficient. The Euclidean distance matrix was used to run the algorithm. As the KNN algorithm suffered from the curse of dimensionality, it was dropped from the analysis.

4.10 Computational Resources

These are the specifications of the server used for running the program code:

- CPU: four Xeon E7-4850 with 40 cores running at 2.00 GHz
- RAM: 256 GB
- Operating System: Windows Server 2019

It is a powerful server system that allows running many programs in parallel efficiently. In order to reduce the processing time, the programs were broken into chunks and executed in parallel because the individual cores are only 2 GHz. However, there are 40 of them available along with a large amount of RAM. Specifically, 24 programs were executed in parallel and tested by evaluating the model performance.

Chapter 5

Results

In this chapter, I report the results of the various machine learning algorithms. Initially, the selected features are provided followed by the findings for regression and classification.

5.1 Selected Features

The given data set has 423 attributes. There are 132 description variables which are removed. On the remaining 192 variables, the algorithm of Recursive Feature Elimination (RFE) was applied with Random Forests . RFE helped in selecting 53 variables out of 192 variables. Twenty seven variables showing high correlation with the response were selected from 192 variables. The regularisation technique of LASSO also helped in selecting three important variables. Further, these 27, 53, and 3 variables were presented to the domain experts and patient partners. These selected

attributes were cross-verified and assessed based on realistic health scenarios. The whole process allowed me to select 61 features out of 423 original attributes. The list of 61 selected features is given below.

1 . Hearing
2 . MakingSelfUnderstood
3 . AbilityToUnderstandOthers
4 . SadMoodRecurrentCryingTearfulness
5 . LengthOfTimeAloneDuringDay
6 . LiveWithClientSecondary
7 . RelationshipToClientPrimary
8 . HoursOfInformalHelp5WeekDays
9 . HoursOfInformalHelp2WeekendDays
10 . MealPreparationSelfPerformance
11 . MealPreparationDifficulty
12 . OrdinaryHouseworkDifficulty
13 . ManagingFinancesSelfPerformance
14 . ManagingFinancesDifficulty
15 . ManagingMedicationsSelfPerformance
16 . ManagaingMedicationsDifficulty
17 . PhoneUseSelfPerformance
18 . PhoneUseDifficulty
19 . ShoppingSelfPerformance
20 . ShoppingDifficulty
21 . TransportationSelfPerfromance
22 . TransportationDifficulty
23 . MobilityInBed
24 . Transfer

25 . LocoMotionInHome

26 . LocoMotionOutsideHome

27 . DressingUpperBody

28 . DressingLowerBody

29 . Eating

30 . ToiletUse

31 . PersonalHygiene

32 . Bathing

33 . ModeOfLocoMotionIndoors

34 . ModeOfLocoMotionOutdoors

35 . StaminaDays

36 . BladderContinence

37 . BowelInContinence

38 . RenalFailure

39 . CoronaryHeartDisesase

40 . Hypertension

41 . IrregularlyIrregularPluse

42 . DementiaOtherThanAlzheimers

43 . Parkinsonsism

44 . Arthritis

45 . HipFracture

46 . OtherFractures

47 . Osteoporosis

48 . Glaucoma

49 . AnyPsychiatricDiagnosis

50 . PainFrequency

51 . FallsFrequency

52 . Swallowing

53 . BetterOffInOtherLivingArrangment

54 . HomeHealthAidesDays

55 . HomeHealthAidesHours

56 . HomemakingServiceHours

57 . MealsHours

58 . PhysicalTherapyHours

59 . MedicalAlertBraceletorElectronicSecurityAlert

60 . NumberofMedications

61 . Anxiolytic

5.2 Results for Regression Problem

For regression, the R^2 value and the mean squared error (MSE) are the two important measures to check the performance of the algorithms. The R^2 value is the statistical measure that computes the percentage of variance in the dependent variable that is explained by the independent variable or variables in the regression model. Mean squared error (MSE) provides a measurement of how far the predictions were from the actual values on average. Figure 5.1 and 5.2 present the R^2 and the MSE values respectively of the regression methods to predict the target.

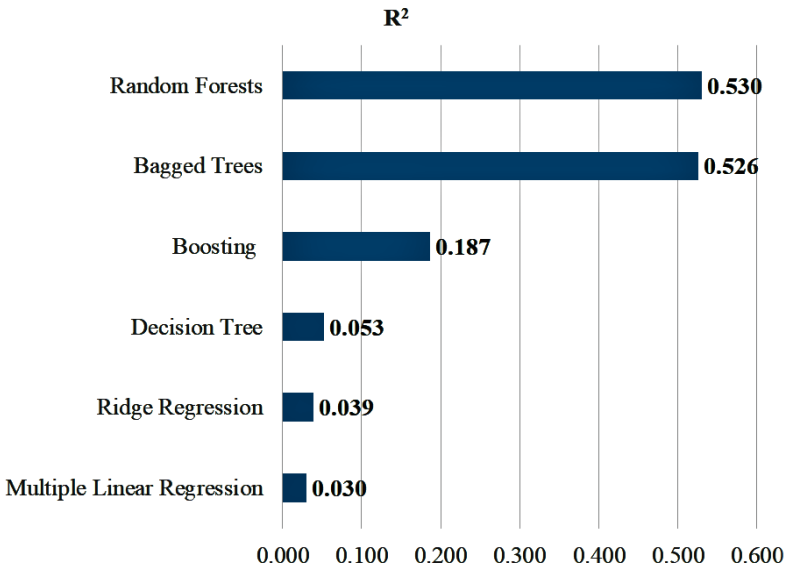

Figure 5.1: R^2 for the regression problem, which was used to predict the target (Average Hours Per Day).

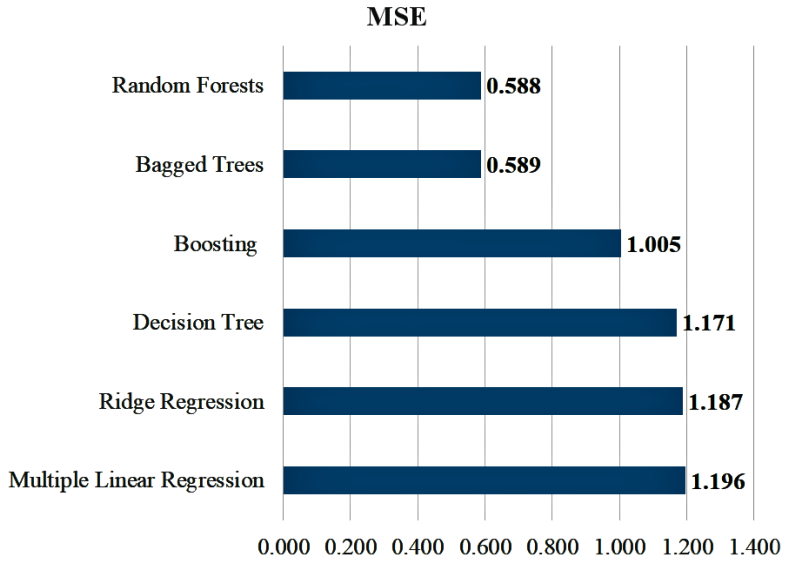

Figure 5.2: MSE for the regression problem, which was used to predict the target (Average Hours Per Day).

Table 5.1: Prediction of the first 10 values of the response variable "Average Hours Per Day" using Random Forests. The bold predicted values are close to the actual values.

Index	Actual Values	Predicted Values
0	0.245	0.198
1	0.540	0.359
2	2.258	**2.265**
3	1.496	**1.487**
4	0.975	0.501
5	0.690	**0.638**
6	1.596	**1.581**
7	0.597	0.439
8	2.258	**2.265**
9	2.258	**2.208**

Table 5.1 shows the actual and predicted values of the response variable. The Random Forests method was used to predict these values. The standard deviation and confidence interval (95%) of the response variable are 1.103 and 0.002, respectively. Taking these facts into account, the predicted values which were within one decimal place of the actual values are presented in bold (Table 5.1).

The ensemble methods Bagged Trees and Random Forests had comparatively high R^2 values, and their MSE values were also low. Hence, these were the promising performers among the regression methods. It is evident from Figure 5.1 that the Random Forest method has the largest $R2$ (53%). The ensemble methods are giving comparatively improved results in

regression since they are uniting the results of numerous basic models to enhance the prediction power.

To check the performance of the regression models, the mean squared errors (MSE) are also calculated. The values of the MSEs are shown in Figure 5.2. It is clearly evident from the plot that the MSE values of the ensemble methods are comparatively low as compared to any other algorithm for regression.

5.3 Results for Classification Problems

The response variable, Average Hours Per Day was dichotomized using the mean, median, 75^{th} percentile, 90^{th} percentile, and 95^{th} percentile of the response variable. The classification results are given in the following sections.

5.3.1 Classification Results for Mean

The mean of the response variable is 0.79 average hours per day. It was used to dichotomize the target, which resulted in an imbalanced classification. The ratio of the classes was $5 : 3$.

The Gradient Boosting, Bagged Trees, and Random Forests were the most promising predictors with the highest accuracy. If the results are compared in terms of cross-validation, the 10-fold cross-validation works well over stratified 10-fold cross-validation and shuffle split cross-validation. Bagged Trees, Gradient Boosting, and Random Forests showed an accuracy

of 0.838, 0.840, and 0.843, respectively. With stratified cross-validation, 0.796 was measured as the highest accuracy for the Gradient Boosting. Whereas with shuffle split cross-validation, Gradient Boosting came up with a 0.810 accuracy.

Figure 5.3 shows that the classifiers such as Decision Tree and Logistic were not the best fit for classification for the given data set based on accuracy. It also illustrates that the 10-fold cross-validation performs well when compared to other cross-validation techniques. The Random Forests, Gradient Boosting, and Bagged Trees have comparatively better accuracy of approximately 84% with 10-fold cross-validation.

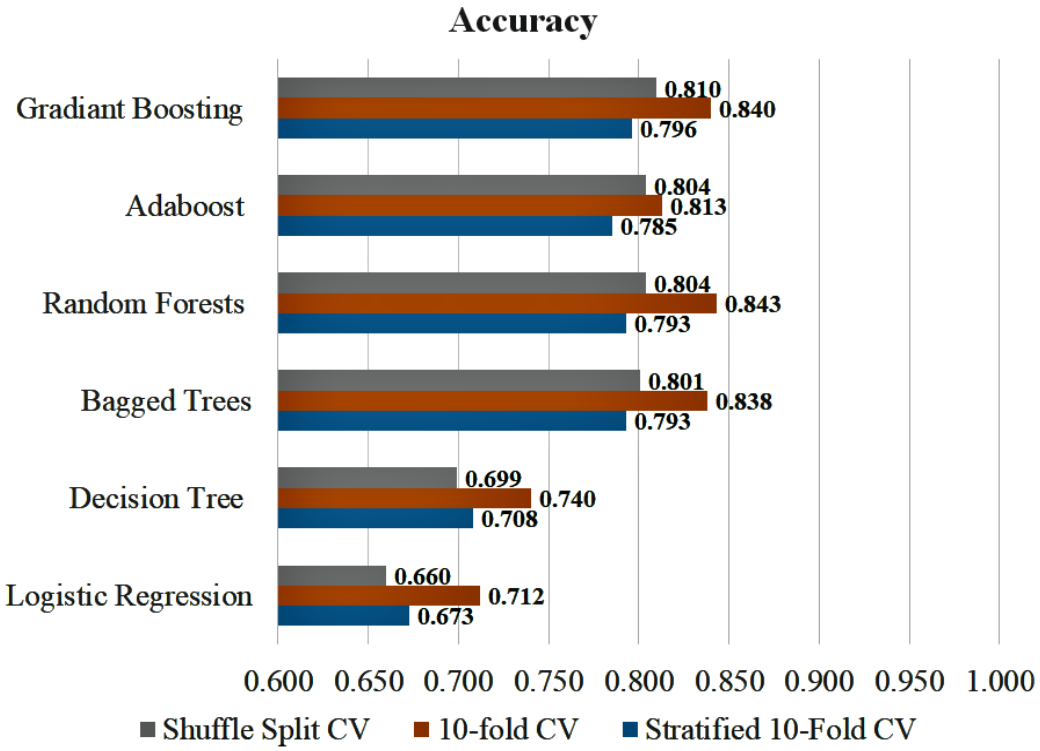

Figure 5.3: Accuracy of classification algorithms when the target is dichotomized using mean.

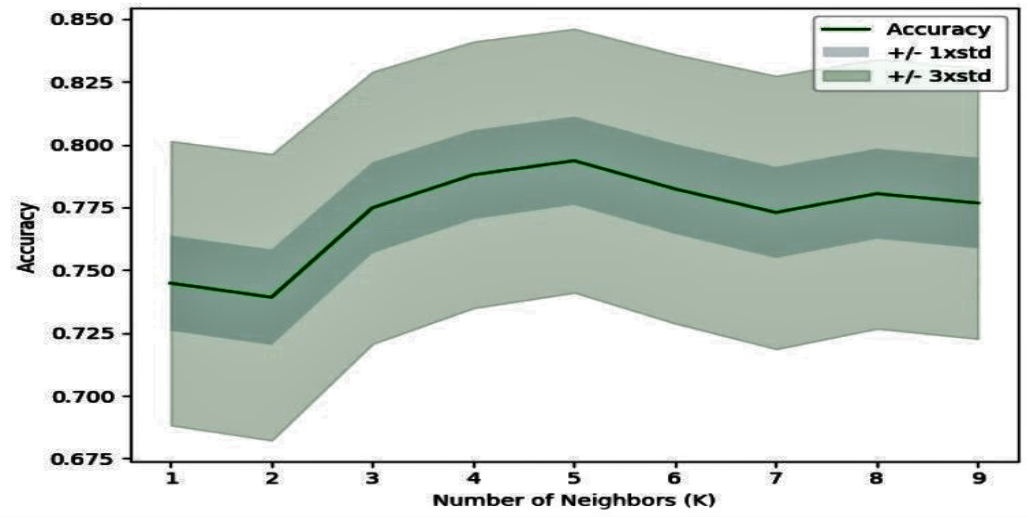

Figure 5.4: KNN with Shuffle-Split CV gives its highest accuracy of 0.787 when the K value is 5 for the given range of K.

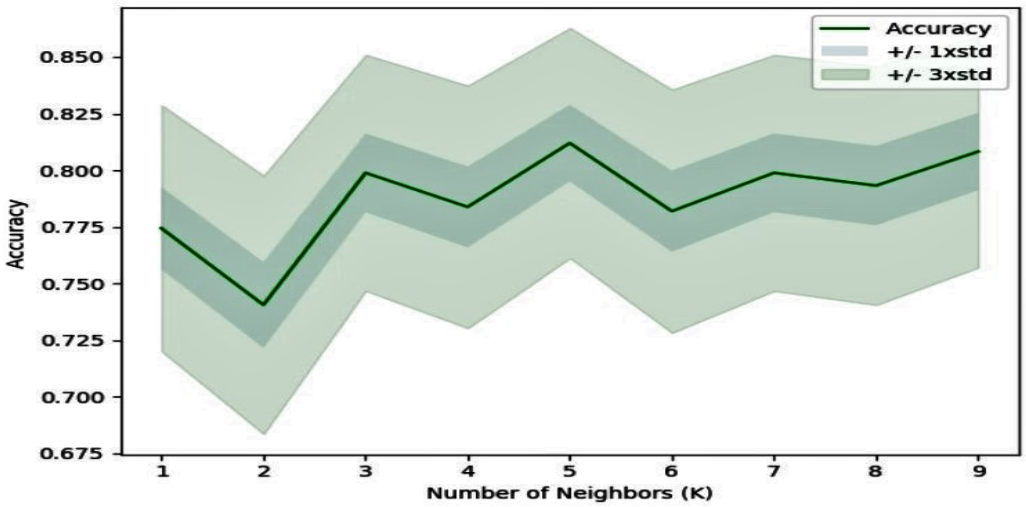

Figure 5.5: KNN with 10-fold CV gives its highest accuracy of 0.783 when the K value is 5 or 9 for the given range of K.

KNN, when run with a limited amount of data (only 1000 rows) showed the accuracy of 0.787 with Shuffle-Split cross-validation when the value of K was 5 (Figure 5.4).

It is also evident from Figure 5.5, at the K-values of 5 and 9 with the same initial 1000 rows, KNN gives almost the same accuracy of 0.783 when executed with 10-fold cross-validation. But as I increased the dimension of the data, the efficacy of the KNN algorithm decreased and its running time increased.

Cross-validation played an important role to overcome the problem of over-fitting in imbalanced class problems. 10-Fold cross-validation was the one that improved the accuracy among all the methods. Hence 10-fold cross-validation gave the highest accuracy when the target was dichotomized using the mean.

10-fold cross-validation was generating improved results with ensemble techniques as it helped in developing the less biased model in comparison to other methods. The reason behind its better performance is that it gives chance for every observation in the data set to show up in the training and test sets.

Since the classes were imbalanced, to check the performance of the model ROC AUC scores were calculated. Fig. 5.6 shows the ROC AUC scores for the different machine learning algorithms. The Random Forests and the Bagged Trees have comparatively good ROC AUC scores of 0.931 and 0.930 respectively.

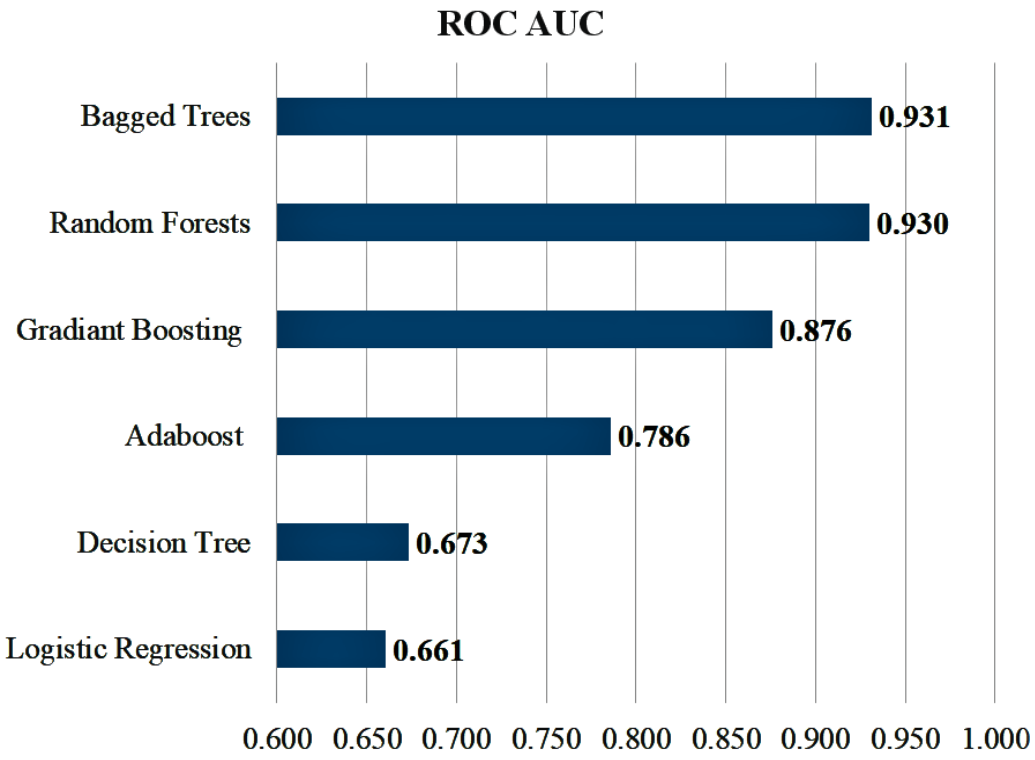

Figure 5.6: ROC AUC scores of classification when the target is dichotomized using the mean.

The top 5 important variables selected using Random Forests when the target was dichotomized using mean, are:

1. HomeHealthAidesHours
2. HoursOfInformalHelp5Weekdays
3. HoursOfInformalHelp2WeekendDays
4. NumberOfMedications
5. ModeOfLocomotionOutdoors

These variables were selected on the basis of their weightage of importance using Random Forests.

5.3.2 Classification Results for Median

The median of the response variable is 0.55 average hours per day. It was used to dichotomized the target. The accuracies for distinct machine learning methods are provided in Table 5.2.

Table 5.2: Classification evaluation using accuracy when the target is dichotomised using median. The highest accuracies are highlighted in bold.

ML Classifiers/CV	Stratified 10 Fold CV	10-Fold CV	Shuffle-Split CV
Decision Tree	0.600	0.598	0.599
Logistic Regression	0.603	0.602	0.605
Adaboost	0.620	0.619	0.623
Gradiant Boosting	0.768	0.652	0.654
Bagged Trees	**0.789**	**0.788**	**0.787**
Random Forests	**0.789**	**0.788**	**0.788**

Decision Tree with 10-fold and shuffle split cross-validation had the least accuracy of 0.59. The classifiers such as Adaboost, Logistic and Boosting had comparatively low classification accuracies. The Bagged Trees and the Random Forests had the highest accuracy of approximately 79%.

Table 5.3 shows the confusion matrix for the Bagged Trees and the Random Forest for one fold out of stratified 10 fold cross-validation. The

accuracies for the Bagged Trees and the Random Forests are;
$(31467 + 31250)/(31467 + 31250 + 8261 + 8501) = 0.789$ and
$(31463 + 31253)/(31463 + 31253 + 8265 + 8498) = 0.789$ respectively.

Table 5.3: Confusion matrix for median classification for one fold within stratified 10-fold cross-validation.

(a) Bagged Trees

		Actual	
		High	Low
Predicted	High	31467	8261
	Low	8501	31250

(b) Random Forests

		Actual	
		High	Low
Predicted	High	31463	8265
	Low	8498	31253

The top 5 important variables selected using Random Forests when the target was dichotomized using median, are:

1. HomeHealthAidesHours
2. HoursOfInformalHelp5Weekdays
3. HoursOfInformalHelp2WeekendDays
4. HomeHealthAidesDays
5. Bathing

These variables were selected on the basis of their weightage of importance using Random Forests.

ROC AUC scores were used to assess the model's performance. The ROC AUC scores for the various machine learning methods are shown in

Figure 5.7. The Random Forests and the Bagged Trees had comparatively good ROC AUC scores of 0.926 and 0.927 respectively.

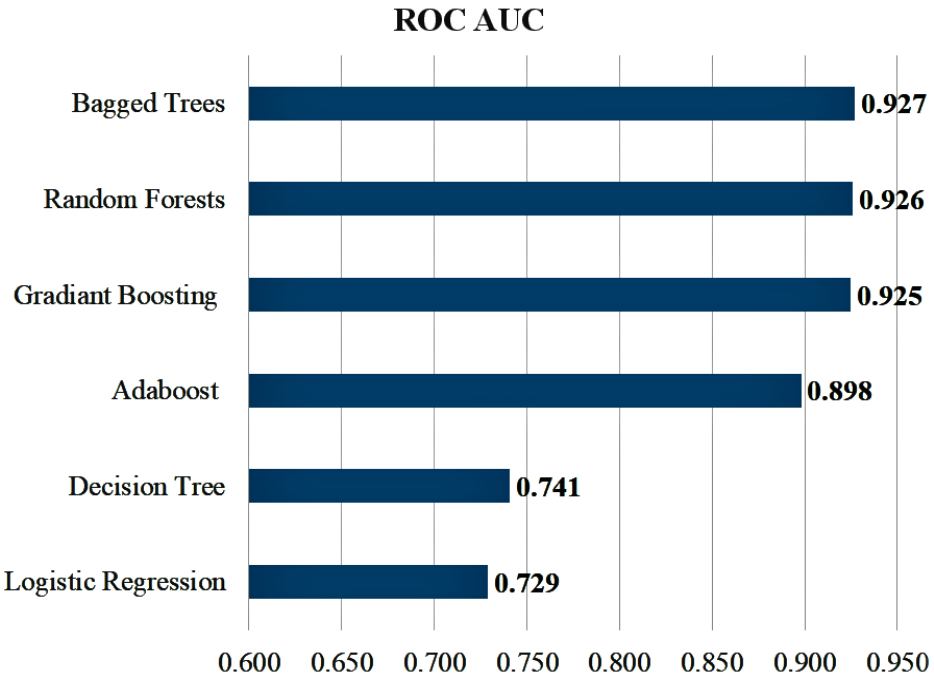

Figure 5.7: ROC AUC scores of classification when the target is dichotomized using the median.

5.3.3 Classification Results for 75^{th} Percentile

The 75^{th} percentile of the response variable is 1.04 average hours per day. When this value was used to dichotomize the target, it resulted in the following target classifications (Table 5.4).

Table 5.4: Formation of classes when the response variable "Average Hours Per Day" is dichotomized using the 75^{th} percentile.

High	198609
Low	596190

For 75^{th} percentile classification of the target, the accuaracies of all the algorithms were above 0.70 and the least accuracy recorded was 0.74. The Bagged Trees and the Random Forests had the highest accuracy of approximately 84% (Table 5.5).

Table 5.5: 75^{th} percentile classification evaluation using accuracy. The highest accuracies are highlighted in bold.

ML Classifiers/CV	Stratified 10 Fold CV	10-Fold CV	Shuffle-Split CV
Logistic Regression	0.751	0.751	0.740
Decision Tree	0.752	0.752	0.752
Adaboost	0.756	0.756	0.755
Gradiant Boosting	0.768	0.767	0.768
Bagged Trees	**0.844**	**0.841**	**0.840**
Random Forests	**0.844**	**0.843**	**0.842**

Table 5.6 shows the confusion matrix for the Bagged Trees and the

Random Forests for one fold out of stratified 10 fold cross-validation. The accuracies for the Boosting and the Random Forests are;
$(12666 + 54451)/(1873 + 74425 + 7194 + 5168) = 0.844$ and
$(2985 + 58058)/(2985 + 58058 + 16875 + 1561) = 0.768$ respectively.

Table 5.6: Confusion matrix for 75^{th} percentile classification for one fold within stratified 10-fold cross-validation.

(a) Boosting classifier

		Actual	
		High	Low
Predicted	High	2985	16875
	Low	1561	58058

(b) Random Forests

		Actual	
		High	Low
Predicted	High	12666	7194
	Low	5168	54451

As the classes for the 75^{th} percentile classification were unbalanced, ROC AUC scores were used to assess the model's performance. The ROC AUC scores for the various machine learning methods are shown in Figure 5.8. The Random Forests and the Bagged Trees had comparatively good ROC AUC scores of 0.940 and 0.939, respectively.

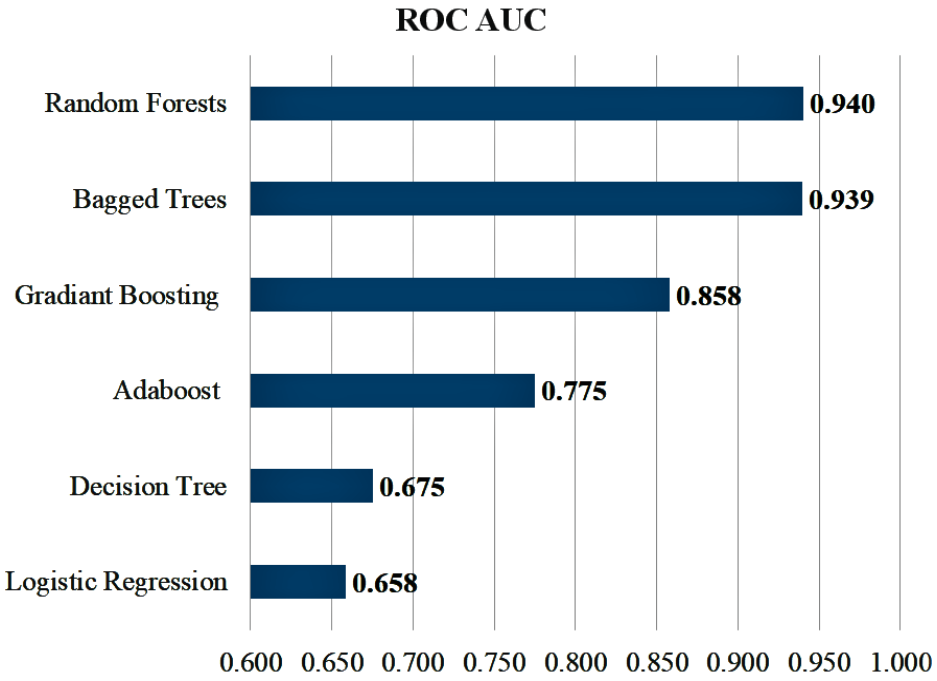

Figure 5.8: ROC AUC scores of classification when the target is dichotomized using the 75^{th} percentile.

The top 5 important variables selected using Random Forests when the target was dichotomized using the 75^{th} percentile, are:

1. HomeHealthAidesHours
2. HoursOfInformalHelp5Weekdays
3. HoursOfInformalHelp2WeekendDays
4. NumberOfMedications
5. BladderContinence

These variables were selected on the basis of their weightage of importance using Random Forests.

5.3.4 Classification Results for 90^{th} Percentile

The 90^{th} percentile of the response variable is 1.59 average hours per day. When this value was used to dichotomized the target, it resulted into following target classifications (Table 5.7).

Table 5.7: Formation of classes when the response variable "Average Hours Per Day" is dichotomized using the 90^{th} percentile.

High	78825
Low	709421

For classification on the basis of 90^{th} percentile value of the target, the Bagged Trees and the Random Forests were the most promising with the highest accuracy of approximately 92% (Table 5.8). The classifiers such as Decision Tree, Logistic and Boosting had comparatively low classification accuracies.

Table 5.9 shows the confusion matrix for the Bagged Trees and the Random Forests for one fold out of stratified 10 fold cross-validation. The accuracies for the Bagged Trees and the Random Forest are;
$(4152 + 69294)/(4152 + 69294 + 3795 + 2238) = 0.924$ and
$(4131 + 69309)/(4131 + 69309 + 3816 + 2223) = 0.924$ respectively.

Table 5.8: 90^{th} percentile classification evaluation using accuracy. The highest accuracies are highlighted in bold.

ML Algorithms/CV	Stratified 10 Fold CV	10-Fold CV	Shuffle-Split CV
Adaboost	0.896	0.903	0.904
Logistic Regression	0.901	0.891	0.902
Gradient Boosting	0.902	0.900	0.901
Decision Tree	0.913	0.915	0.914
Bagged Trees	**0.924**	**0.921**	**0.924**
Random Forests	**0.924**	**0.920**	**0.923**

Table 5.9: Confusion matrix for 90^{th} percentile classification for one fold within stratified 10-fold cross-validation.

(a) Bagged Trees

		Actual	
		High	Low
Predicted	High	4152	3795
	Low	2238	69294

(b) Random Forests

		Actual	
		High	Low
Predicted	High	4131	3816
	Low	2223	69309

Since the classes for the 90^{th} percentile classification were unbalanced, so ROC AUC scores were used to assess the model's performance. The ROC AUC scores are shown in Figure 5.9. Both Random Forests and Bagged Trees have a comparatively good ROC AUC score of 0.969. The Decision Tree has the least ROC AUC score (0.700).

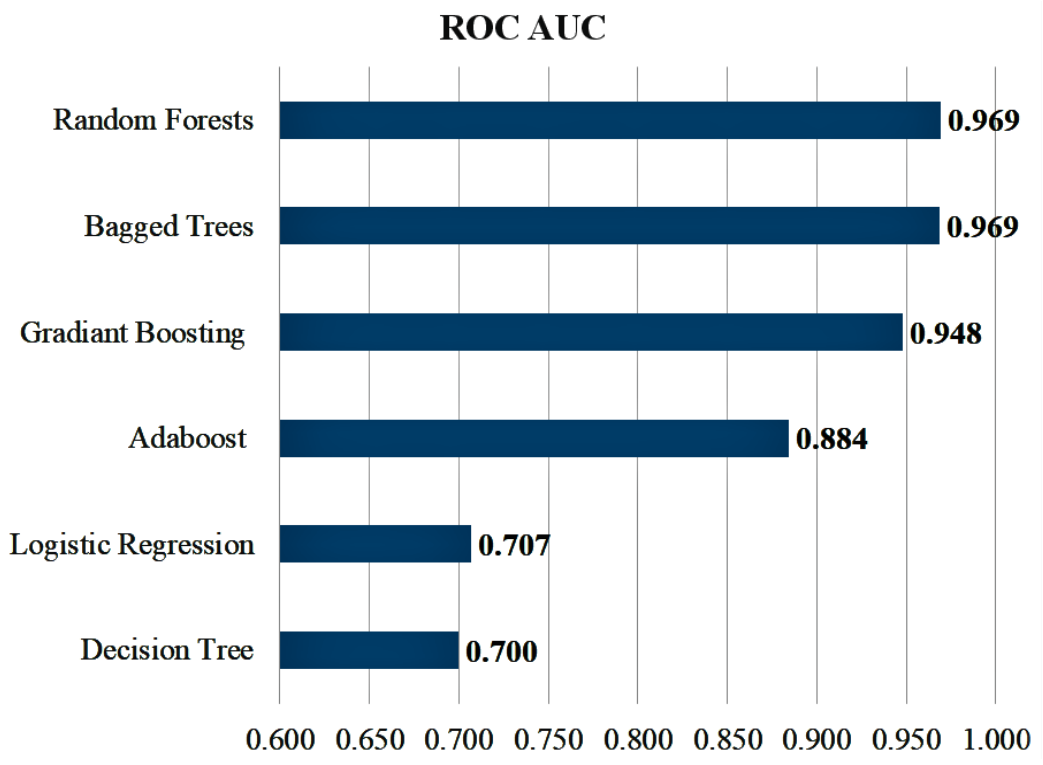

Figure 5.9: ROC AUC scores of classification when the target is dichotomized using the 90^{th} percentile.

The top 5 important variables selected using Random Forests when the target was dichotomized using the 90^{th} percentile, are:

1. HomeHealthAidesHours
2. HoursOfInformalHelp5Weekdays
3. HoursOfInformalHelp2WeekendDays
4. NumberOfMedications
5. Bathing

These variables were selected on the basis of their weightage of importance using Random Forests.

5.3.5 Classification Results for 95^{th} Percentile

The 95^{th} percentile of the response variable is 1.93 average hours per day. When this value was used to dichotomize the target, it resulted in the following target classifications (Table 5.10).

Table 5.10: Formation of classes when the response variable "Average Hours Per Day" is dichotomized using the 95^{th} percentile.

High	39356
Low	748890

Table 5.11: 95^{th} percentile classification evaluation using accuracy. The highest accuracies are highlighted in bold.

ML Algorithms/CV	Stratified 10 Fold CV	10-Fold CV	Shuffle-Split CV
Logistic Regression	0.950	0.953	0.942
Decision Tree	0.951	0.951	0.953
Adaboost	0.951	0.950	0.951
Gradient Boosting	0.953	0.952	0.951
Bagged Trees	**0.960**	**0.962**	**0.960**
Random Forests	**0.960**	**0.960**	**0.963**

For the 95^{th} percentile classification of the target, the accuracies of all the algorithms are good and the least accuracy recorded is 0.942 (Table

5.11). The Bagged Trees and the Random Forest have the highest accuracy of approximately 0.96%.

It was also noticed that when the higher percentile values (the 90^{th} and 95^{th} percentiles) were chosen to dichotomize the target cross-validation did not contribute greatly to the variation in the value of accuracy. All of the machine learning algorithms were producing analogous results for any of the chosen cross-validation techniques.

Table 5.12 shows the confusion matrix for the Bagged Trees and the Random Forest for one fold out of stratified 10 fold cross-validation. The accuracies for the Bagged Trees and the Random Forest are;
$(1873 + 74425)/(1873 + 74425 + 2094 + 1087) = 0.959$ and
$(1868 + 74421)/(1868 + 74421 + 2099 + 1091) = 0.0.959$ respectively.

Table 5.12: Confusion matrix for 95^{th} percentile classification for one fold within stratified 10-fold cross-validation.

(a) Bagged Trees

		Actual	
		High	Low
Predicted	High	1873	2094
	Low	1087	74425

(b) Random Forests

		Actual	
		High	Low
Predicted	High	1868	2099
	Low	1091	74421

The classes for the 95^{th} percentile classification were unbalanced, so ROC AUC scores were used to assess the model's performance. The ROC AUC scores for the various machine learning classification techniques are shown in Figure 5.10. Both Random Forests and the Bagged Trees have

comparatively good ROC AUC score of 0.979.

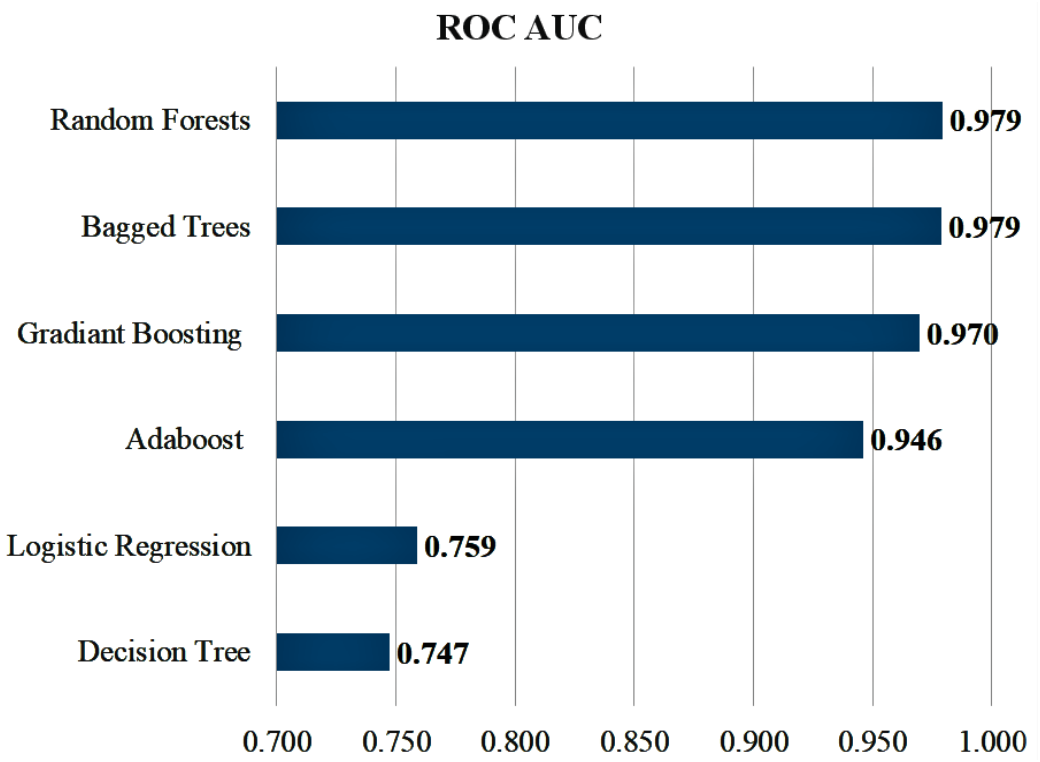

Figure 5.10: ROC AUC scores of classification when the target is dichotomized using the 95^{th} percentile.

The top 5 important variables selected using Random Forests when the target was dichotomized using the 95^{th} percentile, are:

1. HomeHealthAidesHours
2. HoursOfInformalHelp5Weekdays
3. HoursOfInformalHelp2WeekendDays
4. NumberOfMedications

5. ToiletUse

These variables were selected on the basis of their weightage of importance using Random Forests.

Chapter 6

Discussion

The original data set was large, with 837,536 records and 423 attributes. It provided me with the chance to learn how to manage and analyze large data sets. The processing time for a 1-fold cross-validation program with classification algorithms was 24 hours. The processing time for 10-fold cross-validation was 10 days. To ensure time efficiency, parallel program processing was used. The provided individual cores were only 2 GHz. However, there were 40 such kinds of cores with a maximum available RAM of 256 GB. It provided the liberty to get the results on time. The involvement of patient partners set a standard for the research. It was challenging to present machine learning results in a way that people without a statistical background could understand. Their participation increases interaction to access and address real-life home care challenges.

The original data set does not include the response variable "Average Hours Per Day". It was calculated from the "Assessment Start Date", "Care End Date", and "Hours" columns of the Service data table. The

total number of care days was calculated by subtracting the assessment start date from the care end date. The average number of hours per day for the next 21 days from the assessment start date was calculated. On health care expert recommendations, following an assessment, three weeks was identified as the crucial time period for any issues reported in the assessment to influence home care service use. The response "Average Hours Per Day" was calculated so efficiently that I was able to find the outliers.

The majority of "Average Hours Per Day" values fall within the range of the mean plus three standard deviations, i.e., $\mu + 3\sigma = 4.09$ hours. Hence, it was considered pragmatic to place the clients who were, in the majority, using small hours of Home Care services as "Low" users. However, the domain experts and the patient partners were more interested in classifying the clients who were using the Home Care services for long hours. The count of such types of clients was very small, and they were labelled as "High" users. This was the reason why only 2 classes were formed. Furthermore, this was the initial stage of the research on this Home Care data. In the future, multiple class classification of the target can be taken into consideration based on adequate reasoning.

The R^2 value of the multiple linear regression was 0.03% which was very low. This small R^2 indicated that the data were not fitting well to the regression model. It might be possible that the data points were far away from the fitted line and were not linear, which resulted in such a small R^2. The Ridge regression algorithm also had a very small $R^2(0.04\%)$. Since the Ridge is also the extension of the linear models, even after penalizing the coefficients, the data were not fitting well to the linear model. The decision

tree's variable splitting was also ineffective, yielding a small 0.05% R^2 value. However, the Random Forests and the Bagged Trees performed well, with R^2 values of 53 and 52 percent, respectively. This implies that a single decision tree was not able to explain the total variance of the regression model. Random Forests and Bagged Tress, on the other hand, rely on the performance of multiple small decision tree models and were thus provided a comparatively good $R2$ values. Therefore, more than 50% of the variance in the response variable was explained by the regression model when Random Forests and Bagged Trees were used.

Initially, the algorithm of KNN was applied for classification. But KNN suffers from the curse of the dimensionality of the data and is not computationally efficient when the size of the data is large. KNN with Euclidean distance for the range of K values from 1 to 9 and for 500 data rows was taking around 20 minutes to predict. Mathematically, KNN took 0.04 minutes for a row. If I increase the size of the data to include 837,536 rows, it will take 33,501.44 minutes, i.e., 23.25 days. These days are just to run the 1-fold of the cross-validation. For this research, 10-fold cross-validation was used. To run a 10-fold CV, KNN will take 232.64 days, i.e., 7.75 months, which is not reasonably time-efficient. The KNN algorithm has the following drawback: it doesn't scale effectively when dealing with massive datasets or the data set having high dimensions because KNN is a distance-based algorithm. When the data size is large, the effort of estimating the distance between a new point and each preexisting point is really large, which in turn lowers the algorithm's efficiency [Jain, 2021].

This research project is an example of patient-oriented research, so the

involvement of the patient partners is of paramount importance. They were vital contributors in the selection of appropriate research questions, project design, and analysis of results. Patient partners can also help uncover common threads and relevant topics by examining narratives. This research took care of the fact that the patients' desired results are encouraged and recorded. As the project continued, there were monthly meetings with researchers and patient partners where the results of the applied approaches were presented, discussed, and reviewed. The involvement of patient partners to inform the research was indispensable. Their involvement helped in correcting the applied algorithms results and declaring whether the results are promising or not. Their participation in the research gave it a decisive and experienced direction.

At this point, I would like to review and discuss the core research questions. My first research question is about finding a promising regression method that can predict the average number of hours per day of home care.

After setting the target to the average number of hours per day of service use for 21 days after a home care assessment, different regression algorithms were applied. The process of regression started with the multiple linear regression model, followed by regularisation methods of ridge and lasso. The R^2 values for the above models were very low.

A decision tree for regression was also used, but analysing the regression tree with large depth was very complex. The reason behind the decision tree's complexity was that, with 61 variables, it consisted of a large number of decision and leaf nodes. Viewing these nodes one by one was very critical. The R^2 value for the decision tree was also small. Finally, the ensemble methods of regression were employed to predict the target. The

Bagged Trees and the Random Forests gave comparatively good R^2 values. A regression model with Bagged Trees and Random Forests can be used to predict the average number of hours a patient needs in home care.

My second research question is about finding good classification methods to classify and dichotomize the number of hours per day on average a client needs in home care.

The target "Average Hours Per Day" was dichotomized into classes. The methods which were used to dichotomize the target into classes are mean, median and the higher percentile values, i.e., 75^{th}, 90^{th} and 95^{th} percentile. Hence to perform this binary classification, initially the KNN algorithm was used with a small amount of data (only 1000 rows). When the data size was increased, KNN was not computationally efficient. Hence it was dropped from the analysis. Therefore, other classification algorithms were used, namely, logistic regression and decision trees. Since the data were imbalanced, three different cross-validation techniques were used, 10-fold cross-validation, shuffle split cross-validation, and stratified 10-fold cross-validation. To visualise a decision tree having a large depth with 61 variables was very complex. Since it consisted of a large number of decision and leaf nodes. Ensemble methods were advised to use for classification and better performance. The Random Forests and Bagged Trees had good accuracy and ROC AUC scores for all classification models. Their confusion matrix results were also good having low type I and type II errors.

Therefore, it is possible to classify the average number of hours per day of home care services the patients are using on the basis of pragmatic divisions like mean, median, 75^{th}, 90^{th}, and 95^{th} percentiles. I was able to find promising classification procedures to get good accuracy and ROC

AUC scores based on these divisions.

Finding the features that are significant out of 423 given variables, is my third and last research question.

The given data set was provided with 423 variables, which means that the data had moderate dimensionality since it possessed a large number of features. It had become of utmost importance to find out the features which affect the response variable (Average Hours Per Day) greatly. Different machine learning techniques were used in order to select features. The 132 features out of 423 variables were just the descriptions of the variables, so these features were dropped, leaving 291 features to work on. The first method was to find the features that possess low correlation among them and to select a single variable from the group of highly correlated variables. This process helped in selecting independent features. These independent features, which possessed a high correlation with the response, were selected. It helped me to select 27 features. The second method for feature selection was to use the recursive feature elimination technique. This technique is employed with Random Forests. The reason for using Random Forests with the recursive feature elimination technique is that the regression and classification results were comparatively better with ensemble methods. Since RFE uses the weights (or coefficients) of the features of the trained algorithm to sort the variables. It further brought down the number of variables from 291. Through this process, only 53 variables were selected. The third technique to select the features out of 291 variables is the LASSO regularisation method, which penalizes the regression coefficients as per the regularisation of the tuning parameter. This regularisation method helped to select three variables. So, with the

help of the three techniques, the process of variable selection is done.

Further, these 27, 53, and 3 variables (the majority of the entries in selected variables through high correlation and lasso were common with the RFE selected variables) were presented in front of domain experts and patient partners. On the basis of realistic health situations, these selected features are cross verified and reviewed by the experts. The whole process allowed me to select 61 features out of 423 initial attributes.

These variables can be plugged into the trained regression and classification model to predict the number of hours on average a client requires of home care in the near future.

While working on this research, the following limitations were noticed:

- The best R^2 value for the regression model was 53%. This implies that 53% of the variance of the response variable (Average Hours Per Day) was explained by the predictor variables, which is not extremely high. This could possibly be improved by tuning the regression methodology, but it could also mean the provided variables were not sufficient to explain all of the variance of the target. Including more variables such as ethnicity, hereditary conditions, and medical history of the clients could possibly improve the R^2 value. Unfortunately, there are many possible variables like these that were not part of our data set, or, in some cases, not available at all in Canadian health data.

- Classes of the response variable were very imbalanced. The use of accuracy to evaluate the performance of the classification model was

unwise since the accuracy of the classification models became saturated irrespective of the applied cross-validation technique. The ROC AUC score was the answer to the issue, which measured the area under the curve at different probability thresholds in order to avoid over-fitting and model saturation.

Chapter 7

Conclusion

In this research, ensemble methods were found to be the most promising for use in either regression or classification of the use of these home care services. While using the mean to split the target, the Random Forests performed nicely with 10-fold cross-validation to give an accuracy of 84.30%. Furthermore, when using the 90^{th} and 95^{th} percentiles as a basis for classification, the accuracy of both the Bagged Trees and the Random Forests irrespective of the cross-validation technique reached up to 92% and 96% respectively. The ROC AUC scores of these classifiers were recorded as 0.96 and 0.97 respectively. Therefore, when the higher quantile/percentile values were used to classify the target of the interRAI home care assessment (RAI-HC) data, the performance of the machine learning classifiers increased. In regression, the use of R^2 and MSE was done to check the performance of the model. R^2 values for Random Forests and Bagged Trees were estimated as 53% and 52% respectively. Random Forests and Bagged Trees were found to be promising for both classification and regression.

The thesis consists of the three research questions, the first question is, **What is the promising method to predict the number of hours per day on average a client needs in home care?**

The Random Forests and the Bagged Trees were found to be promising machine learning methods to predict the number of hours per day on average a client needs in home care. The noted R^2 values for both methods were 0.530 and 0.526, respectively. The R^2 for boosting was 0.187 which was comparatively low.

The second research question is, **What are good classification methods to classify and dichotomize the number of hours per day on average a patient needs in home care?**

For classification, the Bagged Trees and Random Forests were found to be good choices to classify the target, Average Hours Per Day, when it was dichotomized using the mean, the median, the 75^{th} percentile, the 90^{th} percentile, and the 95^{th} percentile. The highest ROC AUC and accuracy were 0.97 and 0.96, respectively.

The third and last research question is, **What are the significant features to predict the usage of home care services for a client in the near future?**

The answer to the above question is, 61 significant features are selected through the application of 3 methods, namely, Recursive Feature Elimination with Random Forests, high correlation with the response, and the regularisation method of LASSO. The Eleven very important features are:

1. HomeHealthAidesHours
2. HoursOfInformalHelp5Weekdays
3. HoursOfInformalHelp2WeekendDays
4. NumberOfMedications
5. HomeHealthAidesDays
6. DressingLoweBody
7. PersonalHygiene
8. BladderContinence
9. Bathing
10. ModeOfLocomotionOutdoors
11. FallsFrequency

The 95^{th} percentile of the response variable is approximately two hours per day. In other words, a large majority of clients in home care receive a maximum of two hours of services on average per day. This information can be highly useful for the policymakers to allocate the available resources for the sustainable planning of home care. This information can also be beneficial for the patient partners and clients at home care to schedule their available hours effectively.

With more refinement, domain experts and health researchers in home care can use models like the ones in this research to predict and classify the usage of hours by clients. Improvement of this research may provide the

option (within some confidence interval) to identify a client who shows the traits of high fall frequency and number of medications as someone who may need more attention in home care. Similarly, if a client uses services like dressing the lower body, personal hygiene, bathing, and has a history of using an ample amount of home health aid hours and days, it could possibly be time for health care authorities to shift the client from home care to long term care.

In future work, multi-class classification can be done to classify the clients. Adding other variables such as ethnicity, hereditary conditions, and medical history of the clients could improve the predictions of the usage of home care services, and under-sampling or over-sampling techniques could be used to overcome the imbalance class issue. Along with this, the techniques of neural networks may be of good use to perform regression and classification tasks to develop improved models. For health care experts and researchers, this research is a vital step forward, and I hope it can be used as a basis to develop other models and do future research on home care or health care.

Bibliography

Sabyasachi Dash, Sushil Kumar Shakyawar, Mohit Sharma, and Sandeep Kaushik. Big data in healthcare: management, analysis and future prospects. *Journal of big data*, 6(1), 2019. ISSN 2196-1115. doi: 10.1186/s40537-019-0217-0. URL http://dx.doi.org/10.1186/s40537-019-0217-0.

Canadian Institute for Health Information. *Seniors in Transition: Exploring Pathways Across the Care Continuum*. 2017. URL https://www.cihi.ca/sites/default/files/document/seniors-in-transition-report-2017-en.pdf.

Heather Gilmour. Formal home care use in Canada. Technical Report 82-003-X, 09 2019. URL https://www150.statcan.gc.ca/n1/en/pub/82-003-x/2018009/article/00001-eng.pdf?st=D7muwQn8.

Canadian Institutes of Health Research. *Strategy for Patient-Oriented Research: Patient Engagement Framework*. 2019. URL https://cihr-irsc.gc.ca/e/48413.html.

Ministry of Health. Home Care - Province of British Columbia, 12 2017. URL https://www2.gov.bc.ca/gov/content/family-social-support

s/seniors/health-safety/health-care-programs-and-services/home-care.

Mu Zhu, Lu Cheng, Joshua J. Armstrong, Jeff W. Poss, John P. Hirdes, and Paul Stolee. *Using Machine Learning to Plan Rehabilitation for Home Care Clients: Beyond "Black-Box" Predictions*, pages 181–207. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-40017-9. doi: 10.1007/978-3-642-40017-9_9. URL https://doi.org/10.1007/978-3-642-40017-9_9.

Lu Cheng, Mu Zhu, Jeffrey W. Poss, John P. Hirdes, Christine Glenny, and Paul Stolee. Opinion versus practice regarding the use of rehabilitation services in home care: an investigation using machine learning algorithms. *BMC medical informatics and decision making*, 15(1):80, 2015. ISSN 1472-6947. doi: 10.1186/s12911-015-0203-1. URL http://dx.doi.org/10.1186/s12911-015-0203-1.

Aaron Jones, Andrew P. Costa, Angelina Pesevski, and Paul D. McNicholas. Predicting hospital and emergency department utilization among community-dwelling older adults: Statistical and machine learning approaches. *PLOS ONE*, 13(11):e0206662, 2018. doi: 10.1371/journal.pone.0206662.

Jacques-Henri Veyron, Patrick Friocourt, Olivier Jeanjean, Laurence Luquel, Nicolas Bonifas, Fabrice Denis, and Joël Belmin. Home care aides' observations and machine learning algorithms for the prediction of visits to emergency departments by older community-dwelling individuals receiving home care assistance: A proof of concept study. *PLOS ONE*, 14(8): e0220002, 2019. doi: 10.1371/journal.pone.0220002.

Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In *2020 11th International Conference on Information and Communication Systems (ICICS)*, pages 243–248, 2020. doi: 10.1109/ICICS49469.2020.239556.

Piyasak Jeatrakul and Kevin Wong. Comparing the performance of different neural networks for binary classification problems. In *2009 Eighth International Symposium on Natural Language Processing*, pages 111–115, 2009. doi: 10.1109/SNLP.2009.5340935.

Deepak Jain. KNN: Failure cases, Limitations, and Strategy to Pick the Right K, 12 2021. URL https://levelup.gitconnected.com/knn-failure-cases-limitations-and-strategy-to-pick-right-k-45de1b986428#:$%$7E:text=Doesn't$%$20work$%$20well$%$20with,the$%$20performance$%$20of$%$20the$%$20algorithm.

Joseph Rocca. Ensemble methods: bagging, boosting and stacking - Towards Data Science, 12 2021. URL https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.

Bovas Abraham and Johannes Ledolter. *Introduction to Regression Models*. 2006.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. URL https://faculty.marshall.usc.edu/gareth-james/ISL/.

Cran.R. Residual diagnostics, 2021. URL https://cran.r-project.org/web/packages/olsrr/vignettes/residual_diagnostics.html.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Abhigyan Singh. Cross-Validation Techniques - Geek Culture, 01 2022. URL https://medium.com/geekculture/cross-validation-techniques-33d389897878.

Monocasual Triangles. Percentiles and quantiles, 08 2015. URL https://www.internalpointers.com/post/percentiles-and-quantiles.

Benai Kumar. Imbalanced classification, Jul 2020. URL https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/.

Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

Buitinck Lars, Louppe Gilles, Blondel Mathieu, Pedregosa Fabian, Mueller Andreas, Grisel Olivier, Niculae Vlad, Prettenhofer Peter, Gramfort Alexandre, Grobler Jaques, Layton Robert, VanderPlas Jake, Joly Arnaud, and Holt Brian. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Sarang Narkhede. Understanding AUC - ROC Curve - Towards Data Science, 03 2022. URL https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

Bekhruz Tuychiev. Powerful feature selection with recursive feature elimination (rfe) of sklearn, May 2021. URL https://towardsdatascience.com/powerful-feature-selection-with-recursive-feature-elimination-rfe-of-sklearn-23efb2cdb54e.

Appendix A

Program Code

Importing Libraries and Metrics

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
from sklearn.model_selection import train_test_split
import sklearn.preprocessing as pp
from sklearn import linear_model
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
import sklearn.tree as tree
import sklearn.ensemble as en
from sklearn.metrics import  roc_auc_score
from sklearn.metrics import accuracy_score
```

```
from sklearn import metrics
```

Cross-validation, Machine Learning, and Scoring

```
k=10
# 10-fold cross-validation
cv1 = KFold(n_splits = k, shuffle = True)
# Shuffle Split cross-validation
cv2 = ShuffleSplit(n_splits = k, test_size=1/k)
# Stratified 10-fold cross-validation
cv3= StratifiedKFold(n_splits = k, shuffle = True)
cvs=[cv1,cv2,cv3]

for cv in cvs:
  print(cv)
  # making the list of classifiers
  clfs = [LogisticRegression(),
        DecisionTreeClassifier(max_depth = 3),
        tree.DecisionTreeClassifier(),
        en.BaggingClassifier(n_estimators=500),
        en.RandomForestClassifier(n_estimators=500),
        en.AdaBoostClassifier(n_estimators=500),
        en.GradientBoostingClassifier(n_estimators=500)]
  for clf in clfs:
    scores = np.zeros(k)
    i = 0
    roc=np.zeros(k)
    j=0
```

```
    for train_index, test_index in cv.split(X,y):
      X_train, X_test = X[train_index], X[test_index]
      y_train, y_test = y[train_index], y[test_index]
      clf.fit(X_train, y_train)
      y_pred = clf.predict(X_test)
      scores[i] = accuracy_score(y_test, y_pred)
      roc[j]=roc_auc_score(y_test,clf.predict_proba(X_test)[:,1])
      i +=1
      j += 1
    # Accuracy Score
    print(type(clf), "Scores",scores)
    # ROC AUC Score
    print(type(clf), "roc",roc)
    # Mean Accuracy Score
    print(type(clf), "Mean_scores",np.mean(scores))
    # Mean ROC AUC Score
    print(type(clf), "Mean_roc",np.mean(roc))
    print(type(clf), "St.Dev",np.std(scores))
```

Function for R^2 calculation

```
# RSquare Function
def RSquare(y_true,y_pred):
  rss=((y_true - y_pred)** 2).sum()
  tss=((y_true - y_true.mean()) ** 2).sum()
  res=1-(rss/tss)
  return res
```