Predictive Regression and Classification Modelling Using Wearable Fitness Device Data

Navin Sridhar (ns36499)

Puneeth Virushabadas (pv6678)

Urjani Chakravarti (uc845)

School of Information, The University of Texas at Austin

PA 397C Introduction to Machine Learning

Dr. Varun Rai

December 4, 2023

Abstract

This study utilizes data from widely used fitness devices such as Apple Watch and FitBit. The primary objectives include developing a predictive model for estimating an individual's calorie expenditure and constructing a classification model to classify individuals into various activity levels like Lying, Running, Self-Paced walking, Sitting, etc. The regression models faced challenges because of the dataset's nonlinear and intricate nature. Among the classification models, the random forest algorithm exhibited superior performance with an accuracy rate of 98%. The paper additionally explores the potential for further research to enhance the accuracy of the regression problem.

Introduction

In the 21st century, fitness has become integral to societal norms. Heightened awareness regarding lifestyle disorders such as diabetes, hypertension, anxiety, and obesity has led individuals to be increasingly mindful of their health decisions and physical activity levels. An increased number of individuals are opting to track their calorie intake and expenditure and monitor their heart rate and weight. Consequently, the widespread adoption of smart devices like the Apple Watch and Fitbit has become prevalent. Most technology companies have a flagship smartwatch with impressive functionality. Manufacturers have incorporated unique features in these devices to monitor calorie expenditure and classify individuals into distinct activity levels. As a result, many individuals opt to rely on smartwatches as their primary health monitoring device.

The primary motivation for this study is to harness the data derived from these smart devices and apply machine learning models for analysis. The data derived from the Apple Watch and Fitbit have been utilized for this study. The study begins by designing and developing predictive models to estimate an individual's calorie burn, considering age, weight, height, gender, number of steps taken, heart rate, distance covered, resting heart rate, Karvonen intensity, and steps times distance. In other words, these predictors have been used to predict the calorie expenditure of an individual. Additionally, the research formulates classification models to categorize individuals into various activity levels like sitting, running, walking, lying, etc. For addressing the regression problem, a range of algorithms, including Multiple Linear Regression, Ridge Regression, Lasso, Support Vector Machines, Random Forests, and Gradient Boosting Regression, are explored. In the case of the classification problem,

algorithms such as logistic regression, naive Bayes, K-nearest neighbors, support vector machines, and random forest are implemented.

This study aims to comprehensively delve into each algorithm, providing a rationale for their selection and elucidating their benefits and drawbacks. Furthermore, the objective is to conduct a comparative analysis of the various models to identify the one yielding the lowest test Mean Square Error or the highest accuracy.

Review of Literature

In the 21st century, a noticeable shift in monitoring physical health has occurred. Traditional tools such as sphygmomanometers and heart rate monitors replace wearable devices like smartwatches. A recent study by T. Poongodi et al. [1] indicates that 74% of people believe wearable sensors help them interact with physical objects. Moreover, the analysis reveals the rising popularity of wearable devices, with one in three smartphone users owning at least five wearable devices.

Wearable devices, especially smartwatches, play a significant role in monitoring physical activities, a focus area for many young adults in recent decades. A study by Hee-Jin Kim et al. [2] explores whether wearable devices and app-based interventions can effectively prevent metabolic syndrome (MetS) by increasing physical activity among middle-aged individuals in rural South Korea. The research demonstrates that these interventions are instrumental in avoiding MetS by enhancing metrics like blood pressure, waist circumference, and HbA1c. Over six months, the intervention group experienced a decrease in body weight and BMI by 0.6 (SD 1.87) and 0.21 (SD 0.76), respectively ($P<.001$).

Hence, it's reasonable to anticipate a continued surge in sales and innovation within the wearable tech industry. This notion has inspired the exploration of machine learning predictive modeling to benefit the Internet of Things sector. The question arises: can the outputs of wearable devices be predicted based on a handful of parameters? Is it feasible to forecast calorie expenditure by considering specific related factors, potentially informing predictions about weight loss or gain? Could these patterns be harnessed to develop highly accurate, intelligent, and adaptable models? Could these trends be used to create smart and adaptive models with high accuracy?

Dataset and Description

The dataset used in this project was obtained from Harvard Dataverse. It comprised participants who completed a 65-minute protocol involving 40 minutes of total treadmill time and 25 minutes of sitting or lying time. Energy expenditure was measured using indirect calorimetry. The dataset included minute-by-minute heart rate, steps, distance, and calories from Apple Watch and Fitbit. The outcome variable was categorized into activity classes: lying, sitting, walking, self-paced, Running 3 METS, 5 METS, and 7 METS. The analysis dataset encompassed 18 columns and 6264 observations of Apple Watch and Fitbit data, including parameters like age, gender, height, weight, steps, heart rate, calories, distance, resting heart rate, intensity (Karvonen), steps multiplied by distance, and activity. The data set comprised 18 columns and 6264 Apple Watch and Fitbit observations.

Exploratory Data Analysis and Principal Component Analysis

In examining the dataset sourced from Harvard Dataverse, our initial focus involved a comprehensive exploration through data analysis techniques. Utilizing graphical representations like graphs and boxplots provided a revealing insight into the dataset's characteristics. The bar graph presentation clearly explained the distribution of activity counts, elucidating lying as the most prevalent activity among the participants. Additionally, the boxplots unveiled the presence of outliers primarily in activities such as lying, running, and sitting, highlighting their scattered distribution compared to other recorded activities.

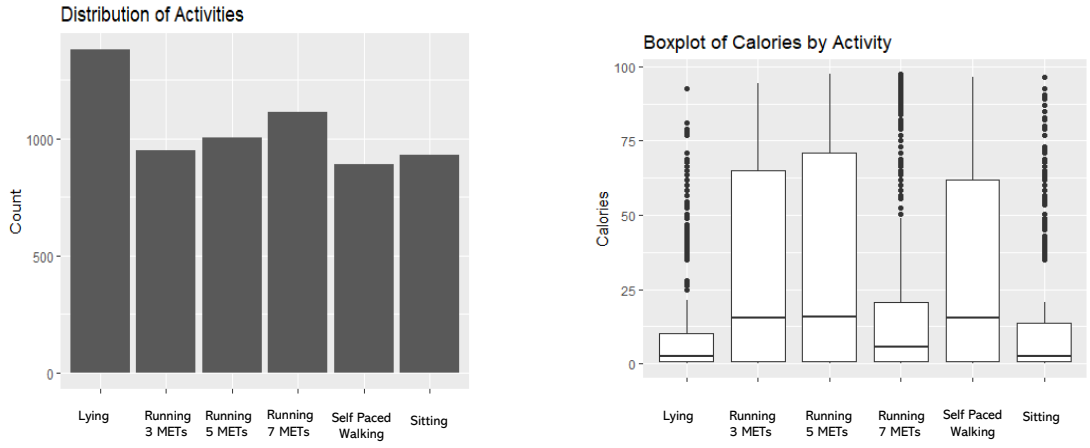


Fig 1: Data Visualization

Beyond graphical analysis, Principal Component Analysis (PCA) was employed to delve deeper into the dataset's structure and prepare it for predictive modeling using machine learning algorithms. The outcomes of PCA shed light on crucial aspects affecting the dataset's suitability for linear modeling. PCA primarily aims at maximizing variance through linear transformations along orthogonal axes. However, our analysis unearthed limitations in efficiently capturing non-linear structures inherent in the original feature space. This limitation is particularly significant when dealing with imbalanced classes or complex overlapping structures within the data, hindering clear separation in a reduced-dimensional space obtained through linear transformations like PCA.

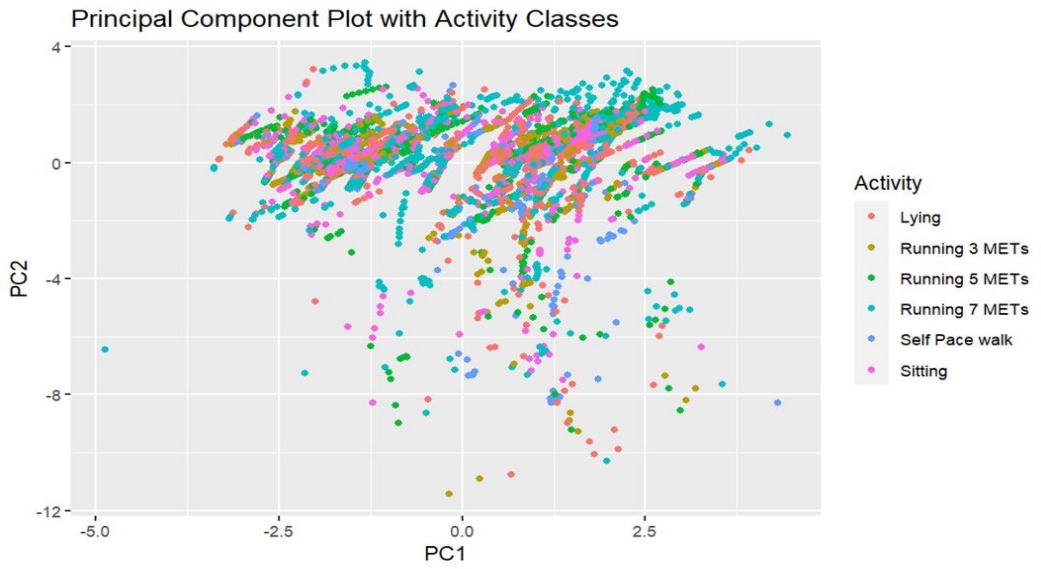


Fig 2: Principal component analysis showing a lack of cluster formation.

Furthermore, the insights drawn from PCA underscored the potential challenges in utilizing linear models for predictive tasks on this dataset. Given the dataset's non-linear and overlapping nature, the expectation is that linear models, whether applied to regression or classification problems, would likely perform sub-optimally. Understanding these nuances is pivotal for selecting appropriate machine learning models that can effectively handle the complexity and non-linearity inherent in the dataset. This preliminary analysis and preprocessing stage is crucial for the subsequent model development and validation. It provides a clearer understanding of the dataset's inherent challenges and guides the selection of suitable machine-learning methodologies to address these complexities effectively.

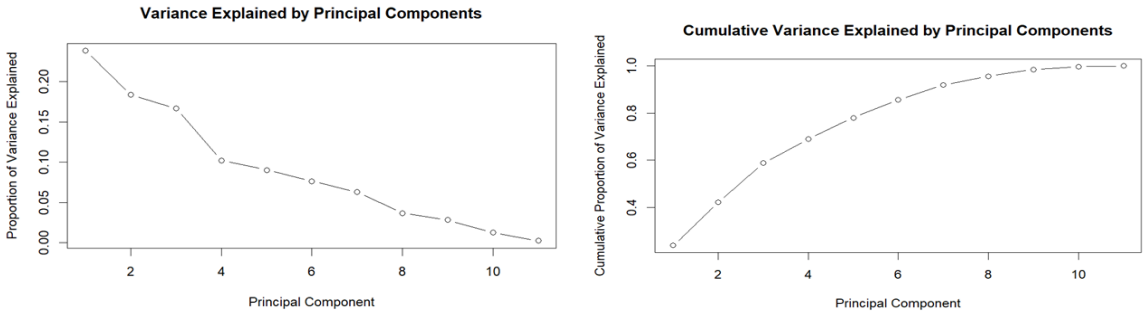


Fig 3: The number of principal components equals the number of predictors (eleven).

I. Regression Predictive Modelling

The primary goal of predictive modeling is to achieve accurate predictions of an individual's calorie expenditure by incorporating various predictors, including age, height, gender, weight, heart rate, distance, Karvonen intensity, and more. Our approach involves the implementation of diverse algorithms such as Multiple Linear Regression (including non-linear terms), shrinkage techniques like Ridge Regression and the Lasso, as well as other algorithms tailored explicitly for complex data, including Support Vector Machines (SVM), Random Forests, and Gradient Boosting. The fundamental motivation behind these efforts is to empower individuals to predict their calorie expenditure when armed with knowledge of other relevant predictors. This capability facilitates effective calorie tracking, giving individuals valuable insights into their energy expenditure. Ultimately, the aim is to enhance personal awareness and enable informed decisions regarding calorie management.

	Test MSE	Test Root MSE
Multiple Linear Regression	635.71	25.21
Ridge Regression	628.98	24.62
The Lasso	626.20	24.65
SVM - Linear	745.29	27.30
SVM - Radial	501.76	22.40
Random Forests	200.96	14.17
Gradient Boosting	233.56	15.28

Table 1: Comparison of various predictive models with Random Forests having the lowest RMSE value.

1.1 Multiple Linear Regression

For the Multiple Linear Regression model, we modeled the response variable calories against the predictors age, gender, height, weight, steps, heart rate, distance, resting heart, Karvonen intensity, and step times distance. Then, the dataset was split into train and test, and cross-validation was performed on the training data to reduce overfitting and selection bias. Finally, we tested the model using the test data by calculating the test MSE. This was used as the metric to analyze and compare the model.

As the Principal Component Analysis suggested, the data contained a high level of non-linearity. As a result, linear models are not expected to perform well. Therefore, we introduced many non-linear terms, including the square of age and weight and other transformations. However, the test MSE was equal to 635.71, and the test RMSE was equivalent to 25.21, which is unsatisfactory. Additionally, the graphs indicated a high level of heteroscedasticity, implying a non-constant variance of errors.

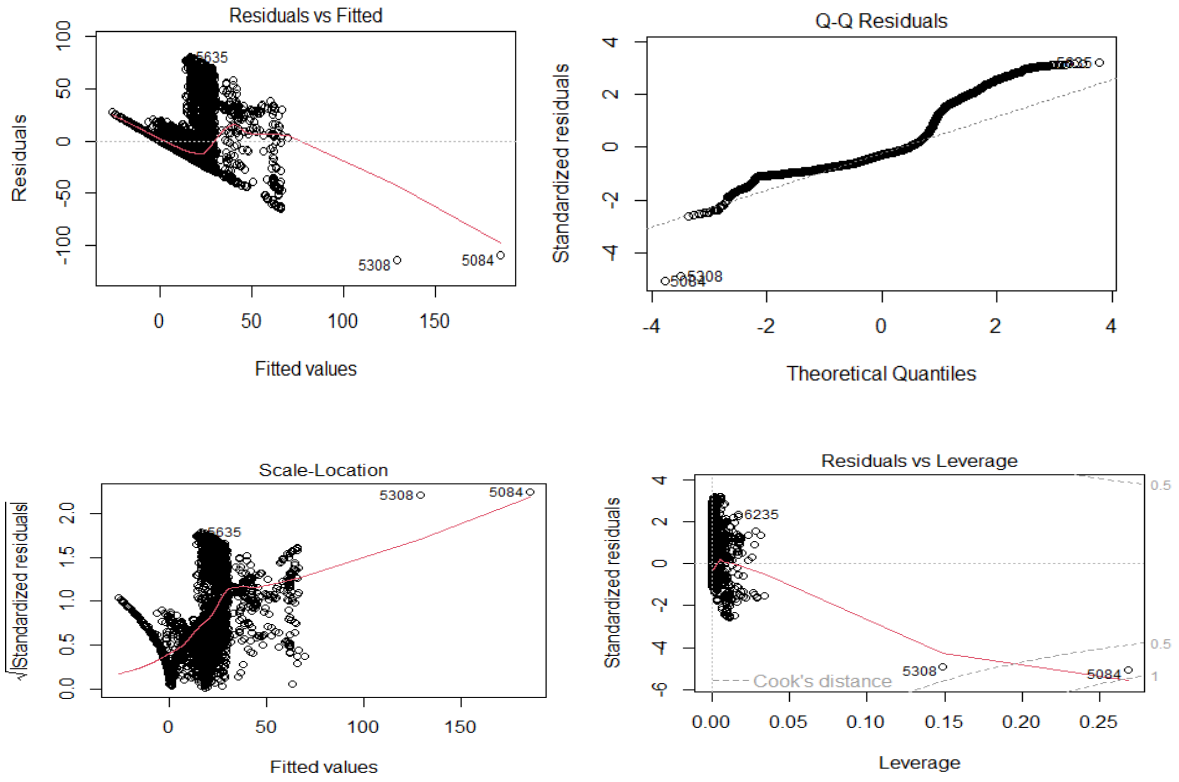


Fig 5: The model's performance does not improve upon adding non-linear terms. The MSE is 635.71, indicating a poor fit. Additionally, the graphs do not show any indication of homoscedasticity.

Even after introducing non-linear terms by squaring age and weight, as it had less significance in the previous Multiple Linear Regression Model, the resulting model didn't perform well-predicting calories, which had a test MSE of 632.15.

1.2 Ridge Regression and The Lasso

Regularizing coefficient estimates in the context of regression refers to introducing a penalty or constraint on the magnitude of the coefficients in the model. This is done to prevent the model from becoming too complex and overfitting the training data, which can lead to poor generalization performance on new, unseen data. We introduced techniques such as zero ridge regression and the lasso for shrinking the regression coefficients toward zero. The main intention was to remove unnecessary variables to improve the test MSE.

Like the MLR model, the data was split into training and testing datasets. The training dataset was used for cross-validation, and the testing dataset was used to calculate the mean square error or the root mean square error. These metrics were used to compare the model to other models and analyze the efficacy of the model in predicting calories. The test MSE from the Ridge model was 628.98, and the test MSE from the Lasso model was 626.20. Both these results only offered a slight improvement over the MLR model.

```
11 x 1 sparse Matrix of class "dgCMatrix"
                              s0
(Intercept)           19.7138057
age                   -0.3918388
gender                 2.1819478
height                 0.6747049
weight                -2.0186067
steps                 -5.4100009
heart_rate             3.1121871
distance               6.5579874
resting_heart         -1.4332256
intensity_karvonen    -6.6793332
steps_times_distance  -0.3209129
```

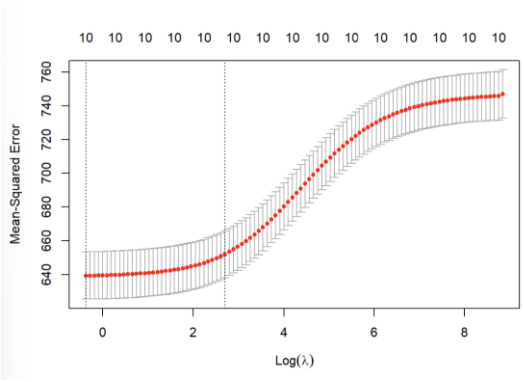


Fig 6: Ridge regression introduces a regularization term, often denoted as λ (lambda), which is multiplied by the sum of the squared values of the coefficients. This regularization term, penalizes significant coefficients and has a shrinking effect on the coefficients. As a result, it helps prevent overfitting by discouraging the model from fitting the noise in the training data. Right: Shrinkage of coefficients from Ridge Regression. Left: MSE versus Log(λ) for Ridge Regression

```
11 x 1 sparse Matrix of class "dgCMatrix"
                            s1
(Intercept)          19.713806
age                   .
gender                .
height                .
weight                .
steps                -4.134861
heart_rate            .
distance              4.524440
resting_heart         .
intensity_karvonen   -2.272192
steps_times_distance  .
```

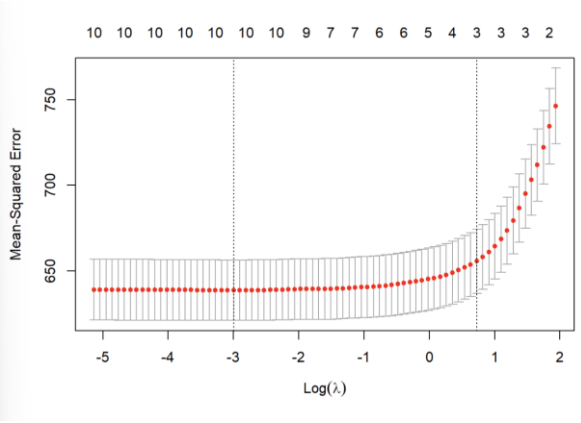


Fig 7: The Lasso induces sparsity in the model by forcing some coefficients to become precisely zero. Right: The coefficients for age, height, weight, heart rate, resting_heart, and steps_times_distance have been zero. Left: MSE versus Log(λ) for Lasso.

1.3 Support Vector Machines

Using different kernel functions, SVM can effectively model non-linear relationships between the input features and the target variable. This flexibility allows SVM to capture complex patterns in the data. Similar to the previous models, cross-validation was performed for training data. Additionally, results were compared for different tuning parameter values, C. We have utilized the linear and radial kernels to implement the SVM model for this study. The RMSE for the linear kernel was 27.30, and the RMSE for the radial kernel was 22.40 for a budget of 1. The SVM linear kernel performed worse than the MLR and shrinkage models, whereas the SVM radial kernel performed slightly better than the shrinkage models.

```
Support Vector Machines with Linear Kernel

4384 samples
  10 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 3507, 3508, 3508, 3506, 3507
Resampling results:

  RMSE      Rsquared   MAE
  27.30698  0.1087766  17.04412

Tuning parameter 'C' was held constant at a value of 1
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 4384 samples
##   10 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 3506, 3507, 3508, 3507, 3508
## Resampling results across tuning parameters:
##
##   C     RMSE      Rsquared   MAE
##   0.25  23.37905  0.3094366  14.73335
##   0.50  22.75756  0.3256673  14.33532
##   1.00  22.40037  0.3396574  14.04019
##
## Tuning parameter 'sigma' was held constant at a value of 0.1710827
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were sigma = 0.1710827 and C = 1.
```

Fig 8: Right: SVM Linear Kernel Outcome. Left: SVM Radial Kernel Outcome.

1.4 Random Forests

The Random Forests algorithm was applied to capture the data's complexity and non-linearity, which is an innate ability of the algorithm. Additionally, it decorates the trees, allowing for an improvement in test MSE values. Like the previous models, cross-validation was performed on the training set, and the test MSE or the RMSE was calculated on the test set. The RMSE was 14.65, a significant improvement over another predictive model. However, the RMSE value is still not satisfactory. This MSE varies with different iterations of training due to the innate nature of random forest to train the model by selecting random features and also due to the bootstrap aggregation technique.

```
## Random Forest
##
## 4384 samples
##   10 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 3507, 3507, 3508, 3506, 3508
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##    2    14.91251  0.7074809  8.146250
##    6    14.73442  0.7130167  7.629136
##   10    14.76261  0.7118878  7.544818
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 6.
```

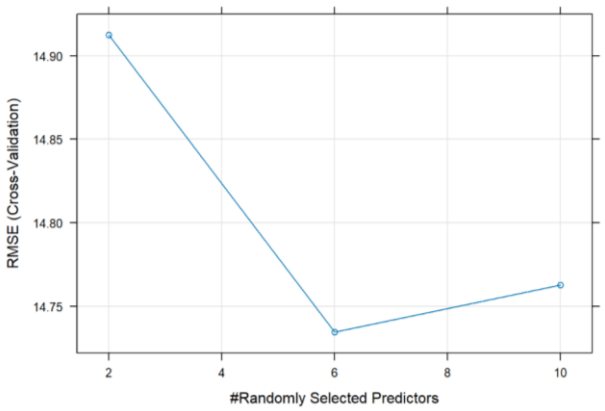


Fig: Right: In the five-folds, 3507, 3508, 3506, 3508, and 3507 samples were used, respectively. The model was tuned over different values of the number of predictor parameters (the number of variables randomly sampled as candidates at each split in a decision tree). Left: The number of predictors value equals 6, showing the lowest RMSE value.

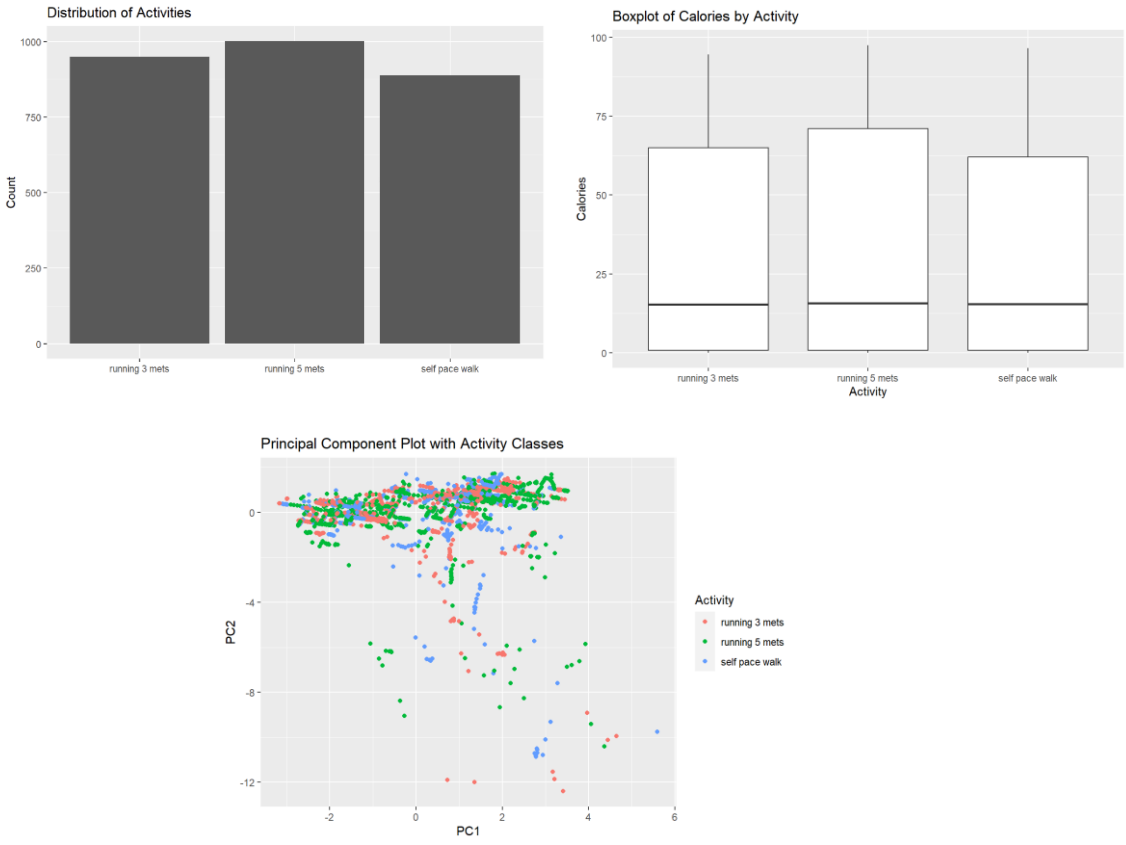
1.5 Gradient Boosting

The final algorithm applied to the predictive problem was gradient boosting to capture information available in the previous trees. A similar approach of cross-validation was followed. Tuning parameters such as the number of trees from 100, 200,…,1000 and with shrinkage parameter (learning parameter) as 0.001, 0.01, 0.1 and with various interaction depth values 2,4,6..10. The best test RMSE obtained was equal to 15.83 with several trees as 500, shrinkage parameter as 0.1 and interaction depth of 10. The value is not as good as the value obtained from Random Forests. Gradient boosting also gave different values at each iteration of training the model but provided better prediction than linear models. However, it is better than the other predictive models.

1.6 Removal of Outliers and Regression Results:

From the boxplot from data analysis from Figure 1, it was evident certain activities, such as Running 7 METs, Lying, and Siting, have many outliers. After dropping the observations with these activities, regression results didn't differ much, and random forest and gradient boosting gave the best results. Results from different models are as follows:

Fig. 9 below: PCA and Box plots after removing activities with higher outliers



	Test MSE	Test Root MSE
Multiple Linear Regression	700.35	26.46
Ridge Regression	724.15	26.91
The Lasso	700.51	24.46
SVM - Linear	817.75	28.59
SVM - Radial	293.62	17.13
Random Forests	138.95	11.78
Gradient Boosting	147.41	15.83

Table 2 Regression results after removing activities with higher outliers

1.7 Inference from Regression Problem Results

Among the results obtained from the various models, Random forest performed better, but it still didn't produce good results concerning the prediction of calories. Classification aims to predict a categorical outcome (e.g., class labels), whereas regression predicts continuous values. The relationships between significant variables for classification might translate poorly into predicting continuous outcomes. One key issue might be that the published dataset might be focused on activity classification, and the data collected might not have constant values, which is essential for regression problems. In classification, imbalanced classes might be managed using techniques like oversampling, under-sampling, or different evaluation metrics. In regression, the distribution of the target variable matters more, especially in predicting different ranges of continuous values.

II. Classification Modelling

	Accuracy Percentage
Logistic Regression	24.77%
Naïve Bayes	35.53%
KNN (k=5)	77.21%
SVM - Linear	29.25%
SVM - Radial	45.27%
Random Forest	98.87%

Table 3: Comparison of various classification models with Random Forests having the highest accuracy.

In our pursuit of classifying activity levels based on diverse predictors, we systematically complexities. Initially, the dataset underwent segregation into Train and Test sets, ensuring a clear distinction for model training and validation. Cross-validation techniques were then exclusively applied to the Train dataset to ascertain model robustness and prevent overfitting. Subsequently, the models underwent rigorous evaluation using the Test dataset, allowing us to gauge their performance on unseen data. We measured the model performance using accuracy metrics derived from confusion matrices and explored different tuning parameters across models to optimize their predictive capabilities.

2.1 Logistic Regression

The logistic regression method was chosen for its simplicity and interpretability. For this approach, the data was split into training and testing. Cross-validation was performed on the training data, and the accuracy was calculated on the test data. The accuracy obtained from this model was equal to 28.17%. The lower accuracy might be attributed to its inherent linearity, struggling to effectively capture the intricate relationships within the dataset.

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction       Lying Running 3 METs Running 5 METs Running 7 METs
##   Lying            179             97             92             73
##   Running 3 METs     4              2              1              5
##   Running 5 METs     2              4              8              2
##   Running 7 METs    25             36             60             78
##   Self Pace walk     2              0              0              3
##   Sitting            5              3              4              3
##                 Reference
## Prediction       Self Pace walk Sitting
##   Lying                     103     122
##   Running 3 METs              1       1
##   Running 5 METs              9       2
##   Running 7 METs             13      35
##   Self Pace walk              1       0
##   Sitting                     2       4
```

Fig 10: Confusion Matrix from Logistic Regression model classification

2.2 Naïve Bayes

The Naïve Bayes model was chosen for its simplicity and proficiency in handling categorical variables. While it showed a moderate accuracy of 35.55%, its assumption of independence among predictors might not align well with the dataset's complexities.

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction       Lying Running 3 METs Running 5 METs Running 7 METs
##   Lying            137             62             66             41
##   Running 3 METs    11             34             10              6
##   Running 5 METs    13             12             53             13
##   Running 7 METs    42             12             27             98
##   Self Pace walk    12             17              8              5
##   Sitting            2              5              1              1
##                 Reference
## Prediction       Self Pace walk Sitting
##   Lying                      62      88
##   Running 3 METs             20       8
##   Running 5 METs              5      14
##   Running 7 METs             10      33
##   Self Pace walk             31      10
##   Sitting                     1      11
```

Fig 11: Confusion Matrix from Naïve Bayes classification model

2.3 K Nearest Neighbours

The KNN algorithm was utilized to capture complex relationships within data. Its relatively high accuracy suggests its effectiveness in identifying patterns within neighboring data points, especially when activities cluster together. The test accuracy obtained from this algorithm was 79.31 % when K=5. K values ranging from 1 to 10 were utilized. For k values 1,2, and 3 to have higher accuracies, one could prefer to have k values as low as possible, but it could lead to overfitting, and the unseen dataset, if used, might not perform well. Here, the test split would have had a similar pattern from the train set, but if we switch devices and provide that data, it might not perform well due to overfitting. So considering other K values 4 and 5 gives a better output overall and comparatively the accuracies or same level from KNN = 4 to 7. So somewhere between these 4 and 7 would generalize the model better and perform well in this case. As we can see from PCA, the data obtained from smart wearables are non-linear and have complex structures in reduced dimensions.

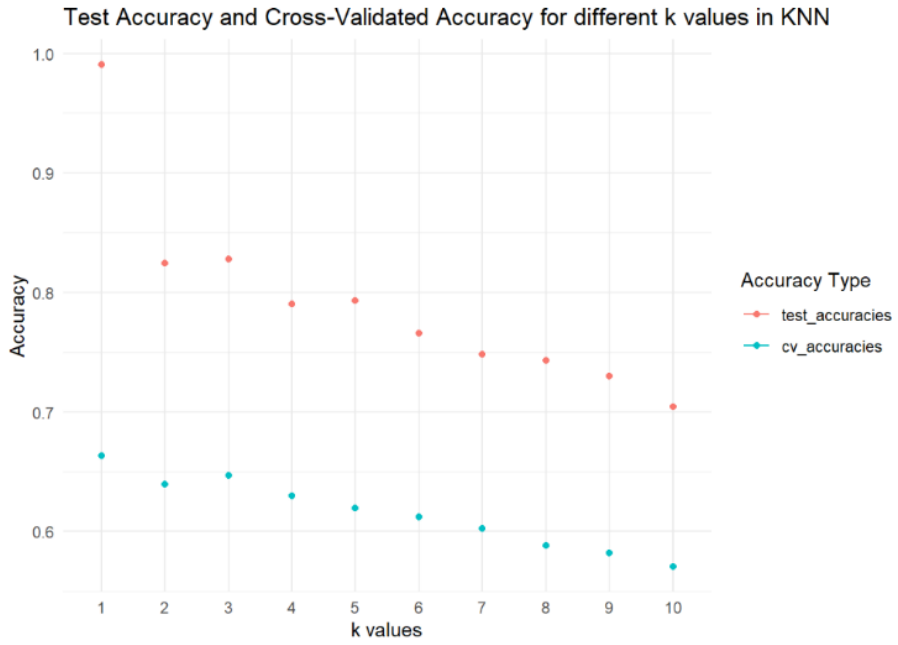


Fig 12: Cross-validated and Test Accuracy for different values of K.

```
## [1] "Confusion Matrix for k = 6"
## Confusion Matrix and Statistics
##
##                Reference
## Prediction       Lying Running 3 METs Running 5 METs Running 7 METs
##   Lying            161             7             4            13
##   Running 3 METs     7           120             5             7
##   Running 5 METs     6             4           142            17
##   Running 7 METs    14             4             7           113
##   Self Pace walk    16             3             1             4
##   Sitting           13             4             6            10
##                Reference
## Prediction       Self Pace walk Sitting
##   Lying                       9      16
##   Running 3 METs             10       8
##   Running 5 METs              0       7
##   Running 7 METs              3      11
##   Self Pace walk            102       9
##   Sitting                     5     113
```

```
## [1] "Confusion Matrix for k = 10"
## Confusion Matrix and Statistics
##
##                Reference
## Prediction       Lying Running 3 METs Running 5 METs Running 7 METs
##   Lying            136            11            13            15
##   Running 3 METs    16           113             3             5
##   Running 5 METs    13             4           128            15
##   Running 7 METs    21             6             8           116
##   Self Pace walk    17             2             4             2
##   Sitting           14             6             9            11
##                Reference
## Prediction       Self Pace walk Sitting
##   Lying                       8      21
##   Running 3 METs              9       8
##   Running 5 METs              2       9
##   Running 7 METs              2      14
##   Self Pace walk            100      14
##   Sitting                     8      98
```

Fig 13: Left: Confusion Matrix for K = 6. Right: Confusion Matrix for K = 10

K	CV_Accuracy	Test_Accuracy
1	0.6493786	0.9910448
2	0.6310327	0.8328358
3	0.6432004	0.8179104
4	0.6220525	0.7731343
5	0.6146787	0.7721393
6	0.5963330	0.7552239
7	0.5939301	0.7512438
8	0.5725794	0.7243781
9	0.5725895	0.7184080
10	0.5729919	0.7134328

Fig. 14 Cross-validated and tested accuracy for different values of k for KNN.

2.4 Support Vector Machines

SVM implementations were chosen for their capability to handle non-linear data. The linear and radial kernels were used. The accuracy of the linear kernel was 34.63%, and the accuracy of the radial kernel was equal to 47.74%. The better performance of Radial SVM indicates its adaptability to the dataset's non-linear nature, while the linear model might have struggled due to this inherent complexity.

```
## Support Vector Machines with Linear Kernel
##
## 5014 samples
##   11 predictor
##    6 classes: 'Lying', 'Running 3 METs', 'Running 5 METs', 'Running 7 METs', 'Sel
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4011, 4012, 4012, 4012, 4009
## Resampling results:
##
##   Accuracy   Kappa
##   0.3149145  0.1452199
##
## Tuning parameter 'C' was held constant at a value of 1
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 5014 samples
##   11 predictor
##    6 classes: 'Lying', 'Running 3 METs', 'Running 5 METs', 'Running 7 METs', 'Se
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4009, 4011, 4011, 4013, 4012
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.3653773  0.2115440
##   0.50  0.3797400  0.2309953
##   1.00  0.4032685  0.2627522
##
## Tuning parameter 'sigma' was held constant at a value of 0.1362409
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.1362409 and C = 1.
```

Fig. 15 SVM outcome for linear kernel and radial kernel.

2.5 Random Forests

The Random Forests method stood out for its ensemble nature, effectively handling non-linearities and intricate relationships among predictors. Its remarkably high accuracy highlights its proficiency in discerning complex patterns within the dataset, outperforming other models significantly. Random forest was executed using different numbers of trees ranging from 15 to 75, and random forest gave a better accuracy for the number of trees between 25 to 40, with overall and class-wise accuracy producing better results. With each iteration of training the model, random forests had different outcomes, and on average, the accuracy made is approximately 98%.

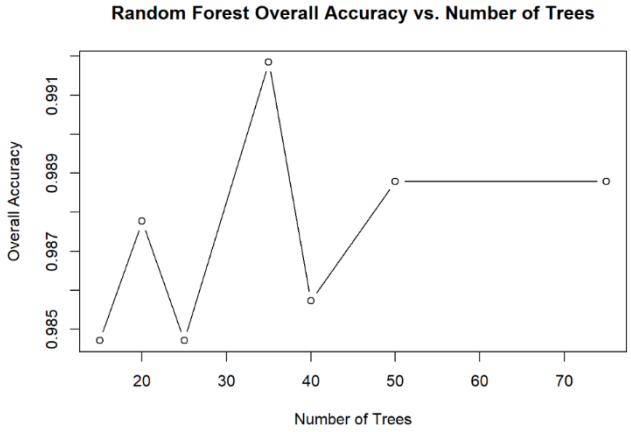


Fig. 16 Accuracy versus Number of Trees.

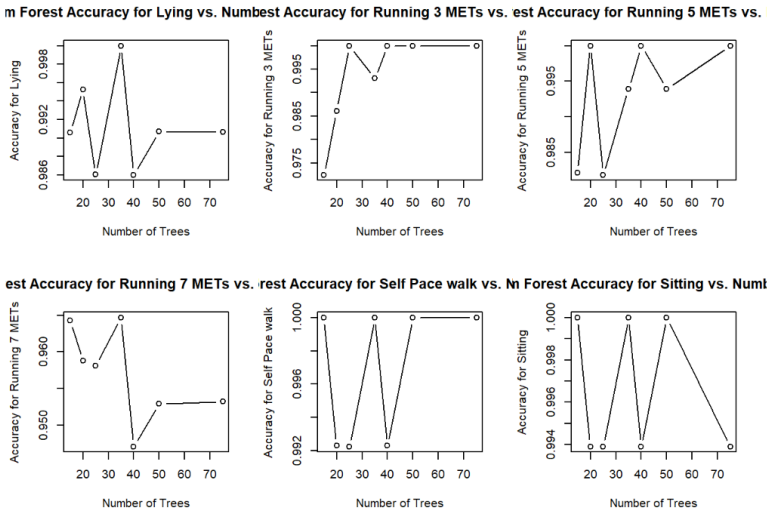


Fig. 17 Accuracy versus Number of Trees for various classes.

2.6 Comparison of Models

From the above analysis, each model showcased varying degrees of performance. Linear models like Logistic Regression and Linear SVM demonstrated lower accuracies, indicating their limitations in capturing the dataset's non-linear nature. Naive Bayes displayed moderate performance, potentially hindered by its assumption of predictor independence. KNN exhibited commendable accuracy, suggesting its ability to identify patterns based on proximal data points. However, Random Forest emerged as the standout performer, boasting notably high accuracy and showcasing robustness in capturing intricate relationships and patterns within the dataset, surpassing other models significantly. Understanding these nuances in model performances aids in selecting the most suitable model for accurately predicting activity levels in real-world scenarios, considering the trade-offs between complexity, interpretability, and accuracy.

Conclusion

The results of our comprehensive analysis and preprocessing stages, followed by rigorous regression and classification modeling, provide crucial insights into our dataset's complexities and the performance of diverse machine learning methodologies.

Regression Modelling: Estimating calorie expenditure using regression models showcased varying predictive accuracies. Linear models like Multiple Linear Regression and SVM Linear exhibited moderate performance, while Ridge and Lasso Regression improved slightly. However, ensemble methods such as Random Forests and Gradient Boosting Regression excelled, delivering significantly lower root mean squared error (RMSE) and superior predictive accuracy.

Classification Modelling: Our objective was to categorize activity levels witnessed diverse model performances. Linear models like logistic regression and naive Bayes displayed moderate accuracy, whereas K-nearest neighbors (KNN) showcased commendable performance. Support Vector Machines (SVM) varied in accuracy between linear and radial implementations. However, Random Forest emerged as the standout performer with remarkably high accuracy, surpassing other models significantly.

Comparison and Implications: The comparison across regression and classification models underscored the limitations of linear methods in capturing the complexities within our dataset. Ensemble methods, particularly Random Forests, consistently outperformed other models,

showcasing their robustness in handling non-linearities and intricate relationships among predictors.

In conclusion, the complexities inherent in our dataset demand modeling techniques capable of addressing non-linearities and complex interactions among predictors. The superiority of ensemble methods, especially Random Forests, in regression and classification tasks signifies their potential for accurate predictions. Understanding these nuances is pivotal in selecting appropriate machine learning methodologies for real-world applications, where accuracy and robustness are paramount. This comprehensive analysis is a foundational guide for informed model selection and deployment in scenarios requiring precise predictions of activity levels and calorie expenditure.

Future Work

The future directions aim to expand the scope of predictive modeling by delving deeper into advanced techniques, diverse datasets, and refined data collection methodologies. Such endeavors will enhance predictive accuracy and applicability in real-world scenarios, thereby advancing the field of predictive analytics in activity levels and calorie expenditure estimation.

Exploration of Neural Networks: Further exploration involving Neural Networks for regression analysis on the dataset presents an avenue for enhanced predictive capabilities. Experimentation with diverse architectures, activation functions, and hyperparameters will be instrumental in assessing their efficacy in accurately predicting calorie consumption. (*Machine Learning-based Approach for Predicting Health Information Using Smartwatch Data*, n.d.) (Mekruksavanich & Jitpattanakul, 2021)

SVM with Varied Kernel Settings: Extending the analysis to encompass Support Vector Machines (SVM) for regression tasks, particularly investigating the impact of different kernel functions like linear, polynomial, and radial basis functions, offers an opportunity to optimize accuracy in calorie predictions.

Assessment of Classification Accuracy: Evaluating classification performance using varied device datasets presents an intriguing prospect. Testing multiple classification algorithms such as Random Forests, SVM, and Neural Networks with data collected from different devices will shed light on model adaptability and accuracy across diverse data sources. This comparative analysis aims to understand the models' robustness when trained and tested on device-specific data.

Subject Knowledge Expert for Data Collection: Developing robust strategies and methodologies for data collection, specifically focusing on acquiring precise data for regression problems, is critical. This involves leveraging subject knowledge expertise to select appropriate data sources (e.g., surveys, sensors, experiments), defining meticulous data gathering protocols, and ensuring stringent quality control measures to maintain data integrity throughout the collection process.

References

Harvard Dataverse. (2020). Replication Data for Using machine learning methods to predict physical activity types with Apple Watch and Fitbit data using indirect calorimetry as the criterion. *Harvard Dataverse*. https://doi.org/10.7910/DVN/ZS2Z2J

Human activity recognition based on improved Bayesian Convolution Network to analyze health care data using wearable IoT devices. (n.d.). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9086799

Kim, H. J., Lee, K. H., & Lee, J. H. (2022). The effect of a mobile and wearable device intervention on increased physical activity to prevent metabolic syndrome: an observational study. *Jmir Mhealth and Uhealth*, *10*(2), e34059. https://doi.org/10.2196/34059

Machine learning-based approach for predicting health information using smartwatch data. (n.d.). IEEE Conference Publication | IEEE Xplore. https://ieeexplore-ieee-org.ezproxy.lib.utexas.edu/abstract/document/9733559

Mekruksavanich, S., & Jitpattanakul, A. (2021). Deep Convolutional Neural Network with RNNs for Complex Activity Recognition Using Wrist-worn Wearable Sensor Data. *Electronics*, *10*(14), 1685. https://doi.org/10.3390/electronics10141685

Poongodi, T., Krishnamurthi, R., Indrakumari, R., Suresh, P., & Balamurugan, B. (2019). Wearable devices and IoT. In *Intelligent systems reference library* (pp. 245–273). https://doi.org/10.1007/978-3-030-23983-1_10

Sarker, I. H. (2021). Machine learning: algorithms, Real-World applications and research directions. *SN Computer Science*, *2*(3). https://doi.org/10.1007/s42979-021-00592-x

Serpush, F., Menhaj, M. B., Masoumi, B., & Karasfi, B. (2022). Wearable Sensor-Based Human Activity Recognition in the Smart Healthcare System. *Computational Intelligence and Neuroscience*, *2022*, 1–31. https://doi.org/10.1155/2022/1391906