Michael Bai

Physical Activity Recognition through Analysis of Wearable Sensor Data

Objective

The primary objective of this project is to create an accurate, robust, and efficient predictive model that is able to identify specific physical activities being performed by a user utilizing sensor data acquired from the PAMAP2 Physical Activity Monitoring of Aging Populations dataset gathered from full-motion wearable devices.

This will enable us to explore the potential of machine learning models to grant us deeper understanding of human physical activity and healthcare monitoring, enabling us to create more effective healthcare interventions, develop personalized healthcare and fitness routines, and improve the quality of life for aging populations, contributing to the broader field of predictive health analytics.

Significance

The importance of this project lies in its potential to revolutionize how we can use wearable technology to enhance our physical well-being. Accurate physical activity predictions can aid in providing personalized healthcare advice by monitoring patients' physical status and detecting abnormal patterns that may indicate underlying health issues.

In addition, in the world of sports and fitness, the model can help tailor training regimes to an individual's needs by understanding their body's limitations and norms when exercising. For instance, by analyzing sensor data during a person's workout routine, the model can discern the intensity of different activities whose results can then be used to predict when the individual might be nearing their physiological limits. This ability to gauge workout intensity can guide the formulation of training programs that optimize performance while minimizing the risk of injury.

Moreover, our predictive model can simply provide a deeper analysis on the diverse variety of exercises in a person's regimen. For a professional athlete, this could mean a nuanced understanding of how different training elements impact their heart rate or body temperature. For others, it could help in monitoring whether exercises are performed correctly or whether the intensity is in line with their fitness level.

The implications of a highly accurate model opens the door to comparative analytics, where a user's performance and physiological markers for some identified activity may be compared to the average performance and attributes for the same identified activity.

Data Gathering

This project makes use of a slightly simplified version of the publicly available and labeled PAMAP2 Physical Activity Monitoring of Aging Populations dataset supplied by the University of California Irvine's Machine Learning Repository.

The dataset comprises data from 9 subjects wearing 3 inertial measurement units on their hands, chest, and ankles performing 13 different physical activities. The dataset has 2,864,056 records and 33 features, and is available as a well-organized CSV file. For the original, non-simplified dataset from UC Irvine, click here.

Each record, representing a snapshot of an instance of physical activity, is labeled by activityID (the exercise being performed) and PeopleId (which of the 9 persons is performing it). The three inertial measurement units placed on the hands, ankles, and chest provide measurements in the X, Y, and Z axis for acceleration (m/s^2), gyroscope (rad/s), and magnetometer (muT), as well as the temperature in degrees Celsius.

The acceleration measures the rate of change in the velocity of an object, useful in detecting sudden movements.

Gyroscope measurements measure the rotational rate around an axis (how fast something is rotating or spinning), used to gauge orientation.

Magnetometer measurements measure the strength and direction of a magnetic field used as a compass to determine the orientation of the subject relative to the Earth's north.

A separate heart rate monitor is used to measure the heart rate of the subject.

Exploratory Data Analysis

Missing Data Analysis

Missing data analysis was performed on the dataset yielding 46 values missing for "heart_rate" out of 2,864,056 records, indicating that a vast majority of the data is present. The *readme.md* file, provided by the dataset upon download, indicates that the missing data is due to minor malfunctions in the heart rate sensor caused by wireless disconnections.
However, upon analysis of the missing records, all 46 are marked under 'transient activities', indicating that the data is missing not at random. Despite the observation, this may not actually be influential since only 46 records out of over 800,000 records for 'transient activities' are missing. Considering the fact that 'transient activities' are over-represented in our dataset, these missing values are rendered less than surprising.

There are no other missing values, nor common missing-number placeholders such as ‘NaN’, ‘???’, 0, ‘unknown’, etc… for any other feature.

Outlier Detection

Outliers for each numerical feature were calculated on the dataset through the IQR method. More specifically, values that lie outside of 1.5 * IQR (Q3 - Q1) are counted as outliers. The results of this analysis are shown below:

```
Column 'heart_rate' has 33781 outliers
Column 'hand temperature (°C)' has 8347 outliers
Column 'hand acceleration X ±16g' has 20232 outliers
Column 'hand acceleration Y ±16g' has 176483 outliers
Column 'hand acceleration Z ±16g' has 51951 outliers
Column 'hand gyroscope X' has 505302 outliers
Column 'hand gyroscope Y' has 523368 outliers
Column 'hand gyroscope Z' has 677605 outliers
Column 'hand magnetometer X' has 18337 outliers
Column 'hand magnetometer Y' has 31867 outliers
Column 'hand magnetometer Z' has 33295 outliers
Column 'chest temperature (°C)' has 8896 outliers
Column 'chest acceleration X ±16g' has 166882 outliers
Column 'chest acceleration Y ±16g' has 464667 outliers
Column 'chest acceleration Z ±16g' has 216368 outliers
Column 'chest gyroscope X' has 442854 outliers
Column 'chest gyroscope Y' has 504882 outliers
Column 'chest gyroscope Z' has 450402 outliers
Column 'chest magnetometer X' has 38052 outliers
Column 'chest magnetometer Y' has 248453 outliers
Column 'chest magnetometer Z' has 25577 outliers
Column 'ankle temperature (°C)' has 75725 outliers
Column 'ankle acceleration X ±16g' has 1007136 outliers
Column 'ankle acceleration Y ±16g' has 515850 outliers
Column 'ankle acceleration Z ±16g' has 210652 outliers
Column 'ankle gyroscope X' has 882668 outliers
Column 'ankle gyroscope Y' has 802742 outliers
Column 'ankle gyroscope Z' has 929448 outliers
Column 'ankle magnetometer X' has 74319 outliers
Column 'ankle magnetometer Y' has 30954 outliers
Column 'ankle magnetometer Z' has 37252 outliers
Column 'PeopleId' has 0 outliers
```

Despite possible initial concerns, outliers, in the context of physiological data should not be removed, nor treated as poor data since there is a high natural variability in human physiology and behavior, especially depending on the activity being performed. Simply put, different activities necessitate different levels of physical strain, and the amalgamation of statistics from 13 different unique exercises will lead to the existence of outliers.

Indeed, outliers may be legitimate data and incredibly useful. For instance, during intense physical activities like running or rope jumping, it is perfectly normal for heart rate to reach levels that are very high compared to rest or light activity, and are classified as outliers despite being expected and useful.

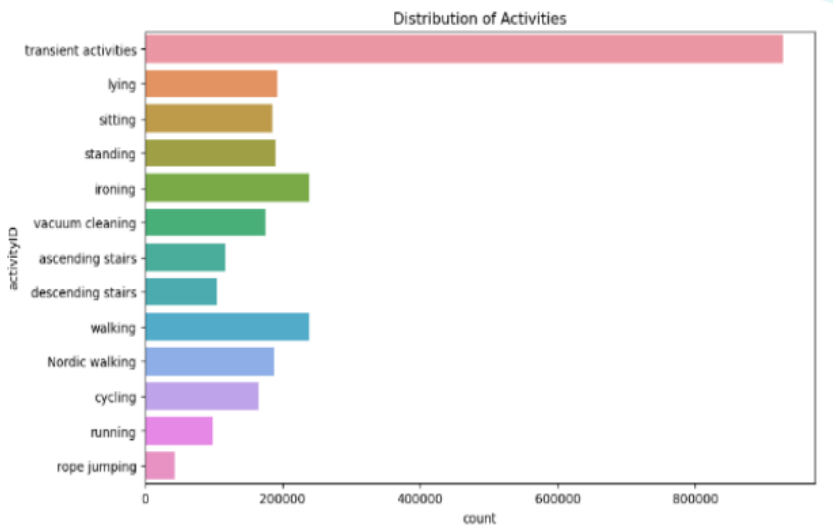
The detection of outliers simply serves to demonstrate the vast variability that is present in each numerical feature, leading to the idea that the type of physical activity being performed leads to a wide range of possible measurements, which may aid in the construction of our multi-class classification model.

Distribution of Activities

The unique activities in this dataset are ‘transient activities’, ‘lying’, ‘sitting’, ‘standing’, ‘ironing’, ‘vacuum cleaning’, ‘ascending stairs’, ‘descending stairs’, ‘walking’, ‘Nordic walking’, ‘cycling’, ‘running’, and ‘rope jumping’.

Upon analysis of the distribution of activities, we see that transient activities (no particular activity / resting) makes up nearly 33% of represented activities, rendering this dataset imbalanced. Otherwise, the dataset would be fairly balanced for the purposes of our multi-classification model. More discussion on what is to be done regarding this potential issue is discussed further in the paper.

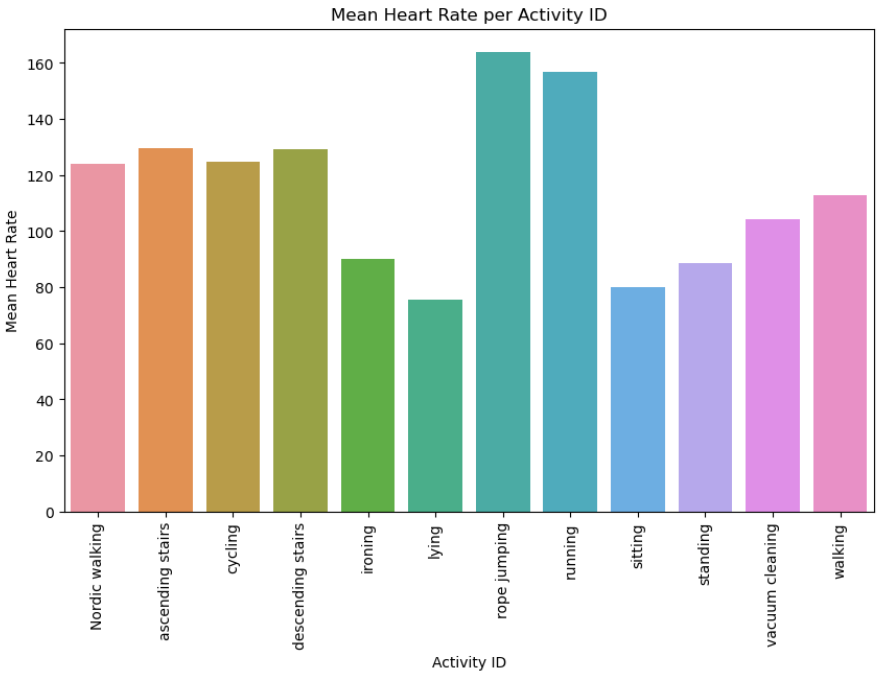
The distribution is shown below.



Case Study: Heart Rate

We also decided to take a closer look at heart rate, given that heart rate seems like a naturally good indicator for our model. Upon graphing the mean heart rate for activity ID, we noticed that rope jumping and running are the two most demanding physical activities.

The graph is as shown:

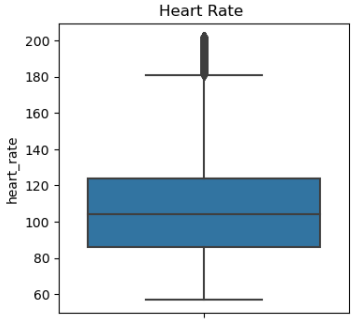


To analyze whether these results are significant, and thus, may provide a deeper understanding of whether or not heart rate is a potentially useful feature, a Student's T-test is conducted with the following hypotheses:

H0: Mean HR of rope jumping and running is not significantly different from the mean of the other activities.
HA: Mean HR of rope jumping and running is significantly different from the mean of the other activities.

The resulting p-value is 0.0 with a t-statistic of 877.53, resoundingly indicating that the mean heart rate between rope jumping and running is indeed significantly different from the mean of the other activities.

A box-plot constructed from heart rate also reveals the prevalence of outliers which may be a useful distinguishing factor that may help our model perform better.



Model the Data - Preprocessing

Our data needs to undergo a refinement process, often known as preprocessing, before it is suitable for modeling. This process encompasses various steps, each serving a unique purpose, specifically feature engineering, feature reduction, balancing the dataset, and encoding.

Feature Engineering and Reduction

Feature engineering shapes the raw data into meaningful and actionable inputs that can capture the intricacies of data and reveal patterns and trends more effectively. In our context, feature engineering will also aid in feature reduction. By combining existing features into a fewer number of representative features, we can drop those existing features and eliminate unnecessary noise and redundancy, as well as reducing the Curse of Dimensionality, which may penalize the performance of a model due to the existence of too many features.

For our project, we computed the magnitudes from the X, Y, and Z components for each of the hand, chest, and ankle accelerations, gyroscope readings, and magnetometer readings, resulting in 9 new features that represent the overall magnitude of these features respectively. This step is essential as it aggregates the multidimensional information into a single meaningful feature, improving both model performance and model interpretability, while reducing overfitting.

Specifically, the following features are created:

‘hand_acceleration_magnitude’
‘hand_gyroscope_magnitude’
‘hand_magnetometer_magnitude’
‘ankle_acceleration_magnitude’
‘ankle_gyroscope_magnitude’
‘ankle_magnetometer_magnitude’
‘chest_acceleration_magnitude’
‘chest_gyroscope_magnitude’,
‘chest_magnetometer_magnitude.

Additionally, we have decided to remove the ‘PeopleId’ feature. PeopleId refers to the person who performed the exercise and can be helpful for the model to learn more about the habits, patterns, and performance trends of an individual person, and use that information to better predict other activities performed by that person. While this feature is potentially amazing for uses in devices such as the Apple Watch, that work to only serve and analyze the user in particular, for the purposes of our multi-class classification model, the model should not draw

influence from the person performing the exercise, simply the physiological measurements of said person.

After feature engineering and reduction, feature count is reduced from 33 to 15.

Removing Transient Activities

As previously discussed, the existence of the activity type ‘transient activities’ in the ‘activityID’ feature leads to an imbalanced dataset, with around 33% of activities being represented as ‘transient activities’. In addition, the records marked ‘transient activities’ are the only ones that have missing values for heart rate. In order to eliminate imbalance in our dataset as well as the few instances of missing values, records in which ‘activityID’ == ‘transient activities’ are removed.

Goalwise, we are able to, and should, remove transient activities since the objective of our model is to classify active activities or exercises. We can ignore those cases in which the person is simply resting, due to the guidelines laid out in our objective.

Encoding Categorical Variables

Machine learning models require numerical input, and thus any categorical variables must be converted into a suitable numerical form. We used label encoding to transform these categorical variables. In label encoding, each unique category value is assigned a unique integer value that it represents.

While in many instances dealing with categorical variables, one-hot encoding is preferred, the restrictions placed upon multi-class classification problems necessitates the use of label encoding.

This is because the ‘confusion_matrix’ function from sklearn among other functions does not support multi-label classification problems. In a multi-label problem, each instance can belong to multiple classes, represented by multiple columns after one-hot encoding. A confusion matrix is generally used for single-label classification problems as for multi-label problems, there is not a single confusion matrix that can adequately capture the results.

One-hot encoding would cause the target variable ‘activityID’ to change into a multi-label format, where each instance CAN potentially belong to multiple classes. This is a completely different kind of problem, as our model assumes that each instance can only belong to one class, and thus, label-encoding, which does not modify the fundamental identity of our problem, is preferred.

Model the Data - Selection and Fitting

After preprocessing and removing ‘transient activities’ records, our dataset contains 1,936,481 records. The data is split 70:30 for training and testing of our predictive models.

Our premier model will be built off of XGBoost, short for Extreme Gradient Boosting. XGBoost is renowned for its efficient and powerful performance in a variety of machine learning tasks. The strength of XGBoost lies in its capacity to continually build new models that specifically target and rectify errors made by preceding models. Through this iterative process, each new model contributes to a more refined, final prediction. In a way, it is similar to random forest, utilizing an ensemble of models to build up to a more powerful one.

One of the unique characteristics of XGBoost is its built-in regularization which helps to prevent overfitting, a common issue where models perform exceptionally well on training data but fail to generalize to new, unseen data. This is supplemented with penalty terms and cross-validation techniques, making it a robust choice for our dataset.

However, to ensure a comprehensive understanding of our data and potential prediction performance, Decision Tree and Random Forest models will also be trained and evaluated. These models offer a different approach to understanding our data, and by comparing their results with those of the XGBoost, we can gain a broader perspective on the performance of our dataset.

The Decision Tree model will be fit without any pre-processing, which represents a baseline for which to compare the performance of our model.

We expect that XGBoost will perform better than Random Forest which will perform better than Decision Tree.

Model the Data - Results and Evaluation

XGBoost

The results for XGBoost are shown below:

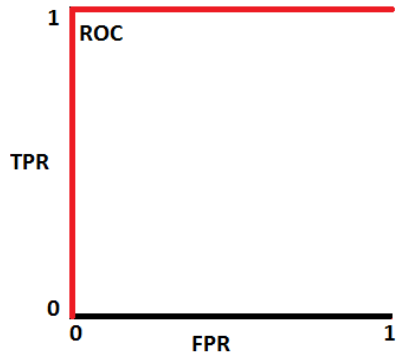
```
XGBoost Classifier Results:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56413
           1       1.00      1.00      1.00     35213
           2       1.00      1.00      1.00     49403
           3       1.00      1.00      1.00     31672
           4       1.00      1.00      1.00     71455
           5       1.00      1.00      1.00     57909
           6       1.00      1.00      1.00     12827
           7       1.00      1.00      1.00     29422
           8       1.00      1.00      1.00     55843
           9       1.00      1.00      1.00     56907
          10       1.00      1.00      1.00     52233
          11       1.00      1.00      1.00     71648

    accuracy                           1.00    580945
   macro avg       1.00      1.00      1.00    580945
weighted avg       1.00      1.00      1.00    580945

[[56413     0     0     0     0     0     0     0     0     0     0     0]
 [    0 35138     0    75     0     0     0     0     0     0     0     0]
 [    0     0 49403     0     0     0     0     0     0     0     0     0]
 [    0    56     0 31616     0     0     0     0     0     0     0     0]
 [    0     0     0     0 71401     0     0     0     0    54     0     0]
 [    0     0     0     0     0 57907     0     0     2     0     0     0]
 [    0     0     0     0     0     0 12827     0     0     0     0     0]
 [    0     0     0     0     0     0     2 29420     0     0     0     0]
 [    0     0     0     0     0     0     0     0 55831    12     0     0]
 [    0     0     0     0    39     0     0     0     3 56865     0     0]
 [    0     0     0     0     0     0     0     0     0     0 52233     0]
 [    0     0     0     0     0     0     0     0     0     0     0 71648]]
```

Key:
0: Nordic walking
1: Ascending stairs
2: Cycling
3: Descending stairs
4: Ironing
5: Lying
6: Rope jumping
7: Running
8: Sitting
9: Standing
10: Vacuum cleaning
11: Walking

Overall, the results of this model are beyond excellent and near perfect. Accuracy is the percentage of correct predictions made by the model out of all predictions. Precision is the proportion of true positive predictions out of all positive predictions made by the model. Recall is the proportion of true positive predictions out of all actual positive instances in the dataset. Finally, F1 - Score is the harmonic mean of precision and recall ranging from 0 to 1.

Across the entire board, precision, recall, f1-score, and accuracy are all rated 1.00 / 1.00, indicating that the model is highly capable of accurately predicting the activity type given these sensor data measurements. Macro and Micro F1-scores are also calculated with results 1.00 and 1.00 respectively.

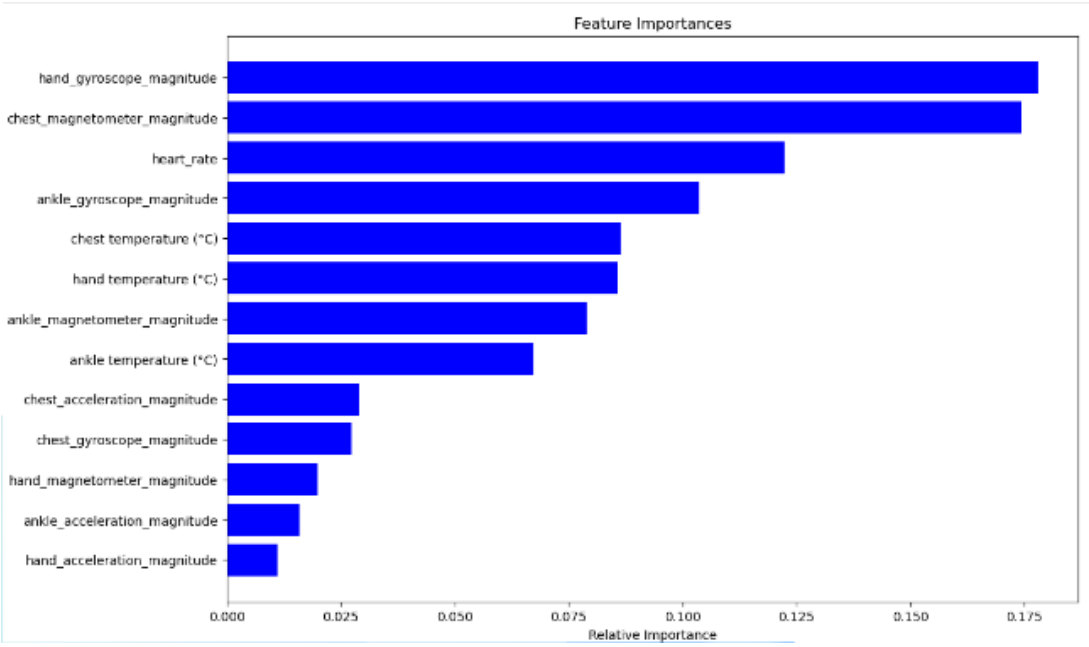
Should a ROC curve be plotted, the graph would look like so:



The red line represents the ROC curve for a perfect classifier.

Though there are minimal misclassifications, some minor confusions between ascending and descending stairs, as well as ironing and standing exist. More specifically, 75 instances of ascending stairs out of 35168 were incorrectly classified as descending stairs while 56 out of 31616 instances of descending stairs were incorrectly classified as ascending stairs. Meanwhile, 54 out of 71401 instances of ironing were misclassified as standing, while 39 out of 56865 instances of standing were misclassified as ironing.

The f-importances, or a measure of how much a feature contributed to the model's behavior relative to other features is shown below:



The model highlights the importance of gyroscope and magnetometer data, particularly from the hand and chest, suggesting that upper body movement is key for physical activity classification. Heart rate also stands out as a significant physiological marker, as expected.

Interestingly, acceleration data holds lesser importance in this model, suggesting that

rotational movements and orientation (captured by the gyroscope and magnetometer) might provide more distinct patterns for the activities in the PAMAP2 dataset.

Random Forest

The results for random forest are shown below:

```
Random Forest Classifier Results:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56413
           1       1.00      1.00      1.00     35213
           2       1.00      1.00      1.00     49403
           3       1.00      1.00      1.00     31672
           4       1.00      1.00      1.00     71455
           5       1.00      1.00      1.00     57909
           6       1.00      1.00      1.00     12827
           7       1.00      1.00      1.00     29422
           8       1.00      1.00      1.00     55843
           9       1.00      1.00      1.00     56907
          10       1.00      1.00      1.00     52233
          11       1.00      1.00      1.00     71648

    accuracy                           1.00    580945
   macro avg       1.00      1.00      1.00    580945
weighted avg       1.00      1.00      1.00    580945

[[56413     0     0     0     0     0     0     0     0     0     0     0]
 [    0 35148     0    65     0     0     0     0     0     0     0     0]
 [    0     0 49403     0     0     0     0     0     0     0     0     0]
 [    0    38     0 31634     0     0     0     0     0     0     0     0]
 [    0     0     0     0 71395     0     0     0     0    60     0     0]
 [    0     0     0     0     0 57907     0     0     2     0     0     0]
 [    0     0     0     0     0     0 12827     0     0     0     0     0]
 [    0     0     0     0     0     0     2 29420     0     0     0     0]
 [    0     0     0     0     0     0     0     0 55822    21     0     0]
 [    0     0     0     0    28     0     0     0     1 56878     0     0]
 [    0     0     0     0     0     0     0     0     0     0 52233     0]
 [    0     0     0     0     0     0     0     0     0     0     0 71648]]
```

Key:
0: Nordic walking
1: Ascending stairs
2: Cycling
3: Descending stairs
4: Ironing
5: Lying
6: Rope jumping
7: Running
8: Sitting
9: Standing
10: Vacuum cleaning
11: Walking

These results are also excellent and nearly identical to that of XGBoost. Other than the obvious observation that all metrics scored 1.00/1.00, the confusions remain similar with ascending and descending stairs, and ironing and standing. These results suggest that the data

provided by the sensors, combined with our preprocessing and feature engineering, are more than sufficient to create a highly accurate and deployable classifier. Because of the quality of our data and preprocessing techniques, XGBoost and Random Forest, which are both relatively high-performing models, have no problem with this multi-class classification task.

Decision Tree, No Pre-processing

Our decision tree model was trained and evaluated without preprocessing in order to provide a baseline for which to evaluate the improvement of our model after introducing feature engineering and dimensionality reduction. Missing values for heart_rate were median imputed.

The results are shown on the next page.

```
Decision Tree Classifier Results:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     56573
           1       0.97      0.97      0.97     34956
           2       0.99      0.99      0.99     49616
           3       0.97      0.97      0.97     31462
           4       1.00      1.00      1.00     71893
           5       1.00      1.00      1.00     57691
           6       0.98      0.98      0.98     12949
           7       0.99      0.99      0.99     29419
           8       1.00      1.00      1.00     55317
           9       1.00      1.00      1.00     56935
          10       0.99      0.99      0.99    278428
          11       0.99      0.99      0.99     52500
          12       0.99      0.99      0.99     71478

    accuracy                           0.99    859217
   macro avg       0.99      0.99      0.99    859217
weighted avg       0.99      0.99      0.99    859217

[[ 56145     19     52     17      2      0      5     14      0      3
     263      3     50]
 [     9  33983     32    330      0      0     12      7      0      1
     507     57     18]
 [    68     25  49155     17      2      0      6     30      1      0
     269     17     26]
 [    11    345     24  30502     16      1      6      9      4      3
     512     16     13]
 [     1      1      2      3  71689      1      0      1      5     59
     111     19      1]
 [     0      4      0      0      1  57596      0      1      3      1
      84      1      0]
 [     6     11     13      4      0      0  12677     88      0      0
     143      3      4]
 [     9     13     13     12      1      1     80  29050      3      0
     235      1      1]
 [     0      1      0      0      1     11      0      0  55158     21
     114     11      0]
 [     0      2      0      0     58      0      0      0     22  56718
     128      6      1]
 [   211    438    193    460    110     76    112    164     98    142
  275852    323    249]
 [     0     34     12     17     30      0      5      2     12      3
     323  52057      5]
 [    32     21     23     13      2      0      1      1      0      0
     309      7  71069]]
```

Key:
0: Nordic walking
1: Ascending stairs
2: Cycling
3: Descending stairs
4: Ironing
5: Lying
6: Rope jumping
7: Running
8: Sitting
9: Standing
10: Transient Activities
11: Vacuum cleaning
12: Walking

The results of this classifier are still good, though not perfect. There are scores that range as low as 0.97, which is still phenomenal. The re-introduction of ‘transient activities’, which has been cut from our premier model, poses confusion and noise, as predicted. The

removal of this, alongside feature engineering, was a beneficial choice in the creation of our perfect model.

Conclusions and Implications

At first glance, the results of our model may seem suspicious. After all, the XGBoost and Random Forest model were basically perfect, very rarely misclassifying activities, even those that may seem closely related. This seems too good to be true.

However, near perfect performance on the models instead provide testimony to the fact that the metrics capable of being captured by the sensors are **highly personalized, accurate, and useful**.

Indeed, the 13 unique activities in our dataset demand a varying spectrum of physical effort, each marking its own “signature” that is able to separate itself from other types of exercises. For instance, a subject who is in the act of running may have wildly higher body temperature, heart rate, and ankle and hand acceleration than an individual who is simply ironing. This extensive variation in data values across different metrics allows our models to confidently and accurately distinguish one activity from another.

Our high-performing classification models have shown that they can be very useful in several real-world applications, particularly in health monitoring, fitness tracking, and patient rehabilitation.

In health monitoring, the models' ability to recognize activities in real-time could enhance patient care. Any sudden changes in activities, indicative of potential health issues, could be flagged for immediate attention.

In fitness tracking, these models could offer valuable feedback to users. By accurately classifying and tracking different activities, users can gain detailed insights about their workout routines and adjust them to better reach their fitness goals as well as monitor signs of fatigue and overexertion by comparing said metrics to their baseline.

In patient rehabilitation, the models could be used to monitor a patient's progress. Accurate recognition of various activities can help healthcare professionals to assess the effectiveness of the rehabilitation program and make necessary adjustments.

The high performance of our models demonstrates the feasibility of using classification models in activity recognition. It highlights the precision and personalization our sensor data can provide, creating a strong foundation for further research and development in these areas.