**Fitbit Activity KPI**

Kunjan Malik
Dept. Software Engineering
San Jose State University
San Jose, CA, USA
kunjan.malik@sjsu.edu

Shivam Shrivastav
Dept. Software Engineering
San Jose State University
San Jose, CA, USA
shivam.shrivastav@sjsu.edu

Praveen Prabhakar Nayak
Dept. Software Engineering
San Jose State University
San Jose, CA, USA
praveenprabhakar.nayak@sjsu.edu

Yadnyshree Savant
Dept. Software Engineering
San Jose State University
San Jose, CA, USA
yadnyshree.savant@sjsu.edu

***Abstract*—Fitness tracker research has been heavily focused on by academia in recent years, including efforts to assess fitness, sleep, heart health, general wellbeing, recuperation from medical maladies, and more. Scholars have used techniques in statistics, machine learning, deep learning, and several other fields to analyze, classify, and predict daily user behavior patterns and outliers in those patterns. Fitbit activity data has been used to predict and visualize general fitness before and during COVID19 lockdown. This project will focus on finding behavior patterns with FitBit data, and then finding indicators that predict calories burned versus actual. Correlating these activities will be performed by using data mining techniques with Python, on a dataset of 35 users over a 60-day time period. In this paper, we are proposing new features to consider for building the classification model. The new features will help the model to be resilient. Training the model with a limited dataset makes the model overfit the data. To avoid that and to enrich the dataset, data amalgamation is applied to the primary Fitbit data with another dataset collected from the Fitbit activity tracker. By performing feature engineering on the available dataset which helped to extract the latent variables in the dataset. We have used regression and classification algorithms in a muller loop to evaluate the overall model's accuracy and metric.**

**Keywords—Data amalgamation, Feature engineering: Data cleaning and Data preprocessing, Visualization, Modelling**

## I. INTRODUCTION

This project aims to develop a fitness dashboard with my KPIs using fitbit data.The downloaded data from Fitbit gave us 380 days of data. Each feature was a separate file which we merged before uploading to the Colab. Calories, Distance and Steps were originally at a minute frequency but we converted the frequency to daily.

Fitbit activity data has been used to predict and chart general fitness. This project will focus on finding behavior patterns with FitBit data, and then finding indicators that predict calories burned actual versus predicted. Correlating these activities will be performed by using data mining techniques with Python, on a dataset of 35 users over a 60-day time period in 2020.

*1) Datasets*

Following are the dataset features recorded by the fitness tracker: Calories - Number of calories burned that day; Sedentary minutes - are basically the number of minutes we are sitting or lying down that day; Moderately active minutes & Very active minutes on the Fitbit dashboard are added

together and referred to as Active minutes. To earn active minutes we have to go through 10 minutes or more of continuous moderate-to-intense activity; Lightly active minutes are the minutes between sedentary and moderately/very active; Distance - the total distance travelled in cm; Steps - are the number of steps taken that day.

```
Data columns (total 16 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Id                         940 non-null    int64
 1   ActivityDate               940 non-null    datetime64[ns
 2   TotalSteps                 861 non-null    float64
 3   TotalDistance              861 non-null    float64
 4   TrackerDistance            940 non-null    float64
 5   LoggedActivitiesDistance   940 non-null    float64
 6   VeryActiveDistance         940 non-null    float64
 7   ModeratelyActiveDistance   940 non-null    float64
 8   LightActiveDistance        940 non-null    float64
 9   SedentaryActiveDistance    940 non-null    float64
 10  VeryActiveMinutes          861 non-null    float64
 11  FairlyActiveMinutes        861 non-null    float64
 12  LightlyActiveMinutes       861 non-null    float64
 13  SedentaryMinutes           861 non-null    float64
 14  Calories                   861 non-null    float64
 15  TotalMinutes               940 non-null    int64
dtypes: datetime64[ns](1), float64(13), int64(2)
```

Fig. 1– Columns in the datasett

## II. EDA and Visualization

EDA — Exploratory Data Analysis is a way of visualizing, summarizing and interpreting the information that is hidden in rows and column format. Once EDA is complete and insights are drawn, its feature can be used for supervised and unsupervised machine learning modelling.

The target feature here is calories burned, the plot below shows it is pretty close to a normal distribution.


Fig. 2– Calories Burnt Histogram (kcal)

Using the plot we see that the Average calories burned over the period was 2,500.

## III. Data Enrichment

*A. Data Preparation:* We checked for the missing values below in our initial dataset and found several values missing. There were missing values for some of the minute features which we filled based on adding each minute feature together and subtracted from 1,440. There are 41 days where total minutes equal 1,440 minutes. However, 16 of these are days where zero active minutes, distance traveled and steps taken were registered. So we replaced these days with NaNs. For the days where the total number of minutes total 1,440 minutes, we found the average proportion for each minute feature. From this we see that sedentary accounts for on average 91% of the 1,440 minutes each day.

*B. Data Distribution:* The target feature is calories burned, the plot below shows it is pretty close to a normal distribution.

*C. Data Amalgamation:* We merged the pre lockdown and during lockdown activity data into a single dataset. We also converted the date from string value to datetime format.

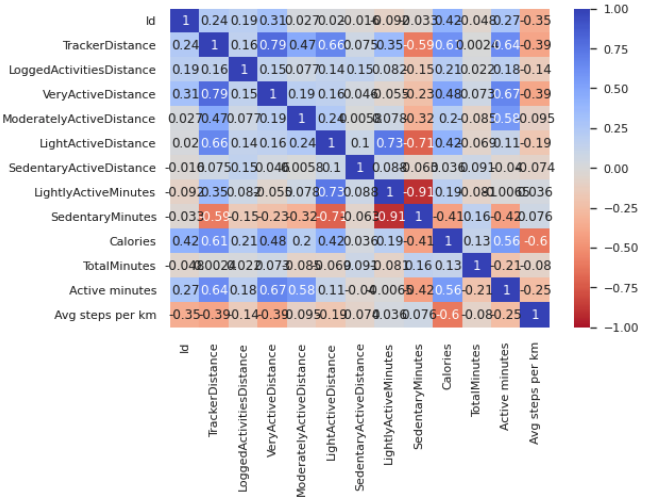



Fig. 3– Correlation Heatmap

## IV. Principal Component Analysis

PCA is used in exploratory data analysis and for making predictive models. It is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible.

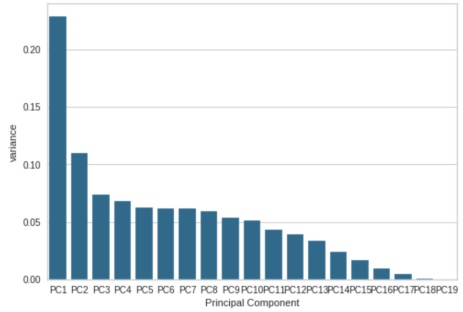



Fig. 4– Bar Plot PCA vs Variance

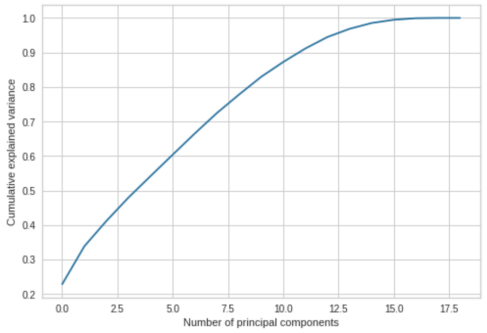



Fig. 5– Principal Component vs Explained Variance

V. Feature Importance

Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable. The role of feature importance in a predictive modeling problem.

Based on the Gini Importance of the features, we predicted that, 'In Lockdown_yes', 'Calories', 'Active Minutes' and 'Avg steps per km' are the most important factors to check the activity.

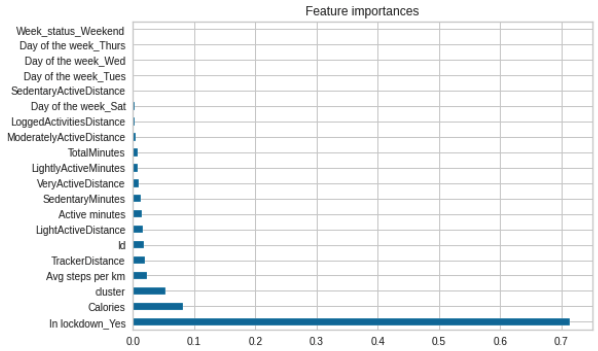


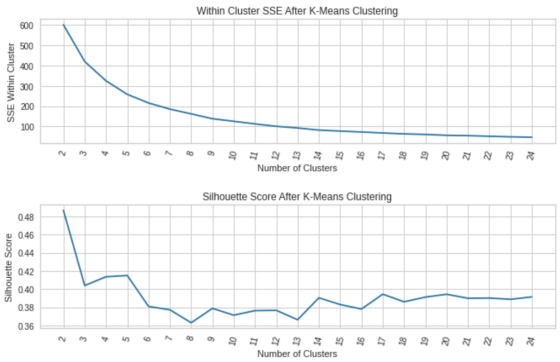

Fig 6 - Feature importance bar chart

VI Fractal Clustering on latent variables

Looking at the performance of various clusters

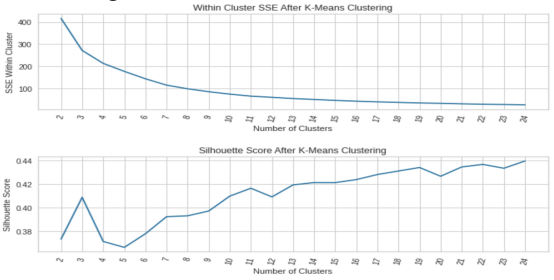
using K-Means. Performance is evaluated within cluster SSE and silhouette score.

▫ Calories vs Active Minutes




```
clustering performance
----------------------------------
silhouette score: 0.49
sse withing cluster: 601.0
```

Fig 7- SSE score with cluster 2

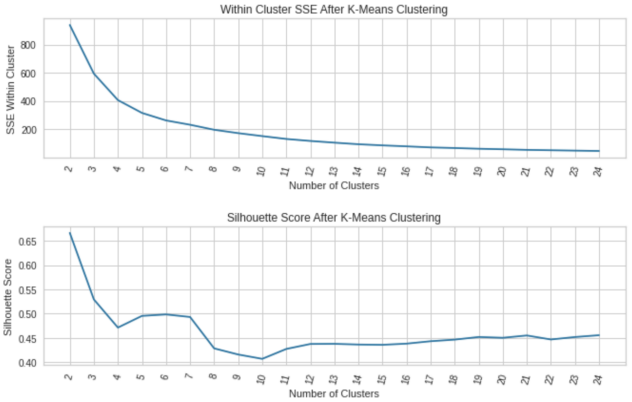



```
clustering performance
----------------------------------
silhouette score: 0.39
sse withing cluster: 98.0
```

Fig 8- SSE score with cluster 8

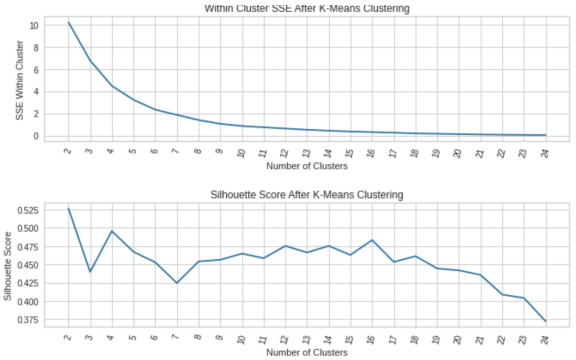
After both the trials , cluster 5 was identified as the Golden Cluster with latent variable as Active Minutes

▫Calories vs VeryActiveDistance




```
clustering performance
----------------------------------
silhouette score: 0.49
sse withing cluster: 232.0
```

Fig 9- SSE score with cluster 7




```
clustering performance
----------------------------------
silhouette score: 0.5
sse withing cluster: 4.0
```

Fig 10- SSE score with cluster 4

After both the trials , cluster 1 was identified as the Golden Cluster with latent variable as Very Active Distance

VII. Modelling

The process of modeling means training a machine learning algorithm to predict the labels from the features, tuning it for the business need, and validating it on holdout data. The output from modeling is a trained model that can be used for inference, making predictions on new data points.

We splitted our data set into 80% train data and 20% test data. With that, we used multiple regression (MLR) models and Classifiers using Muller loop to find the best fit for the given data.

We deduced that the performance of the model increases over the time. An interesting finding from root mean squared error and absolute mean error with the GBM models was that days of the week feature is important for this predicted analysis in our project of fitbit data.

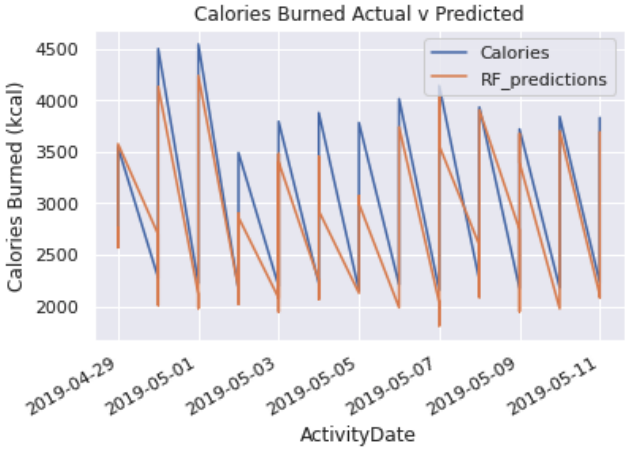



Fig 8 - Actual Calories burnt vs Predicted

We plotted the predicted calories over the actual calories in the test data below. It's close but could be better.

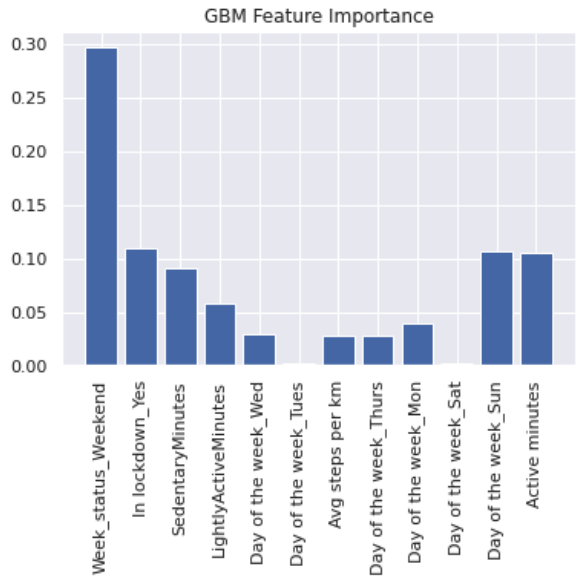



Fig 9 - GBM Feature Importance

Regression analysis is a fundamental concept in the field of machine learning. It falls under supervised learning wherein the algorithm is trained with both input features and output labels. It helps in establishing a relationship among the variables by estimating how one variable affects the other.

We applied seven regression algorithms to check the best suited algorithms for our dataset. Following are the regression algorithms applied: Gradient Boosting Regression, Random Forest Regression, Linear Regression, SVR, Decision Tree Regression, AdaBoost Regression, Gaussian Process Regression. Below is the result with best regression with Gradient Boosting Regression having maximum accuracy of 88.94%.

```
--------------------------------------------------------------------------------
Best --> regressors = GradientBoostingRegressor, Score (test, accuracy) = 81.74
R2 SCORE = 1.00,
regressors = GradientBoostingRegressor, Score (test, accuracy) = 88.94,
R2 SCORE = 0.96,
regressors = RandomForestRegressor, Score (test, accuracy) = 84.20,
R2 SCORE = 0.74,
regressors = LinearRegression, Score (test, accuracy) = 65.37,
R2 SCORE = 0.01,
regressors = SVR, Score (test, accuracy) = 0.66,
R2 SCORE = 1.00,
regressors = DecisionTreeRegressor, Score (test, accuracy) = 87.17,
R2 SCORE = 0.84,
regressors = AdaBoostRegressor, Score (test, accuracy) = 80.25,
R2 SCORE = 0.03,
regressors = GaussianProcessRegressor, Score (test, accuracy) = 2.70,
--------------------------------------------------------------------------------
Best --> regressors = GradientBoostingRegressor, Score (test, accuracy) = 88.94
```

Fig 10 - Best Regression Algorithm

Classifier is the algorithm itself – the rules used by machines to classify data. A classification model, on the other hand, is the end result of your classifier's machine learning. The model is trained using the classifier, so that the model, ultimately, classifies your data.

We applied seven regression algorithms to check the best suited algorithms for our dataset. Following are the regression algorithms applied: Nearest Neighbors, Linear SVM, RBF SVM, Decision Tree, Ada Boost, Random Forest, Neural Net, Naive Bayes, QDA. Below is the result with best regression with Random Forest having maximum accuracy of 88.83%.

```
--------------------------------------------------------------------------------
Best --> Classifier = Random Forest, Score (test, accuracy) = 88.83
--------------------------------------------------------------------------------
```

Fig 11 - Best Classifier- Random Forest

V111. Confusion Matrix for Best Classifier.

A confusion matrix is a summary of prediction results on classification problems. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix.

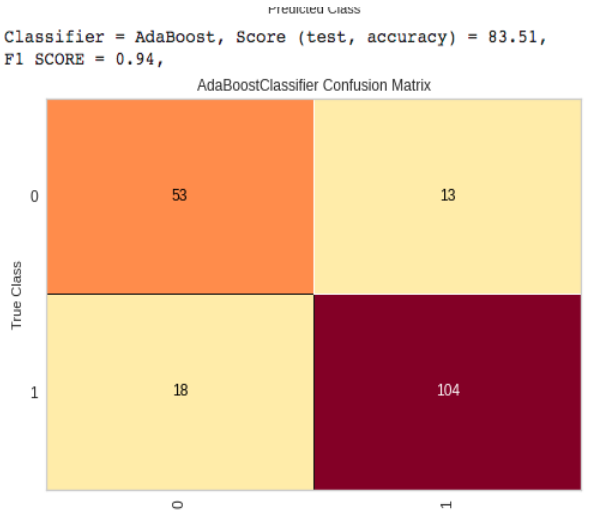



Fig 12 - Decision Tree

IX Gini Score.

It is the measurement of the impurity or randomness in the data pointA high order of disorder means a low level of impurity, let me simplify it. Entropy is calculated between 0 and 1, although depending upon the number of groups and classes present in the dataset it could be larger than 1 but it signifies the same meaning, i.e. higher level of disorder.

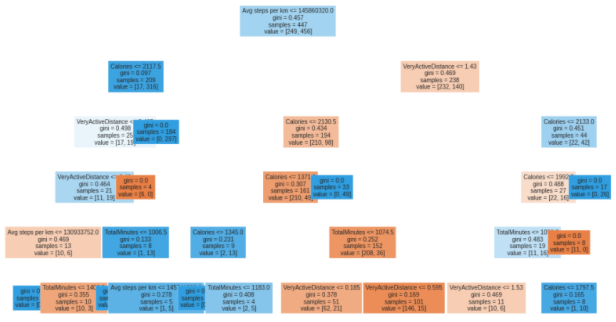


Fig 13 - Gini Tree

X. REFERENCES


[1] Neelam Tyagi, "Understanding the Gini Index and Information Gain in Decision Trees" 23-March-2020.[Online]. Available: https://medium.com/analytics-steps/understanding-the-gini-index-and-information-gain-in-decision-trees-ab4720518ba8

[2] Tobias Geisler Mesevage, "Machine Learning Classifiers - The Algorithms & How They Work" 14-Dec-2020.[Online]. Available: https://monkeylearn.com/blog/what-is-a-classifier/

[3] Vihar Kumara, "Regression in Machine Learning: What it is and Examples of Different Models" 04-Sep-2019.[Online]. Available: https://builtin.com/data-science/regression-machine-learning