Jacob Haight
Layton Washburn
github
presentation

CS 5830 Project 8

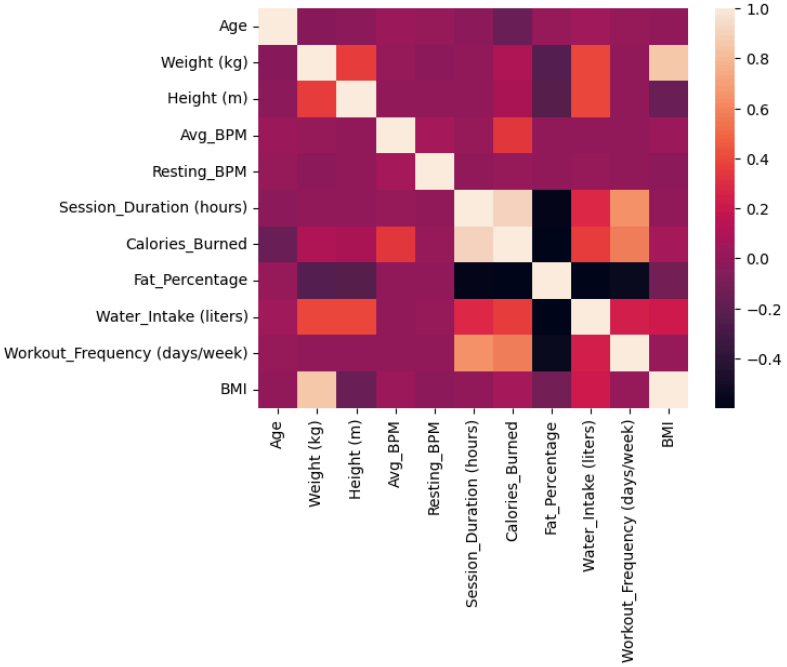
1. Introduction

Our research and project was based on an exercise data set, which can be found at, https://www.kaggle.com/datasets/valakhorasani/gym-members-exercise-dataset. Our stakeholders are people interested in opening gyms, people working out, and anyone interested in work out statistics. We utilized two different types of machine learning models, decision trees and neural networks to predict the calorie category a workout falls into. We needed to figure out which features to use and we did that by using a correlation matrix and chose the features,Session_Duration (hours), Experience_Level, Fat_Percentage, Workout_Frequency (days/week), Water_Intake (liters), Avg_BPM. For the neural networks, we created a function that allowed us to build a model and modify the hidden layers. We tested various sets of hidden networks to produce a model. Our decision tree model turned out to be a good predictor of a calorie category.

2. Dataset

This data set focuses on certain features such as gender, height, weight, BMI, and others. After discussing which variable to predict, our project focused on predicting calories burned, in one of three categories (small, medium, and large) and will inform stakeholders of which calorie category a workout falls into.

3. Analysis technique

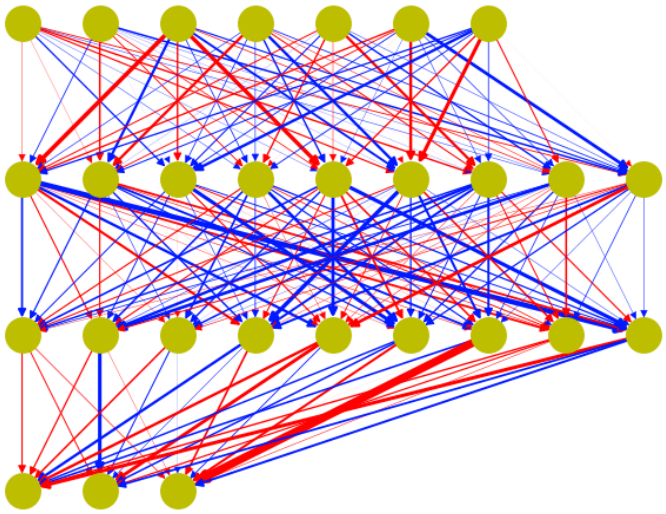


For the neural network, we decided to choose features based on this correlation matrix, which was a pair-wise correlation. Based on this heatmap, we then found the most correlated features used to test and select the optimal group of features.

Different sizes of trees and networks were tested on different feature sets to find an effective model.

4. Results
    1. Neural Network

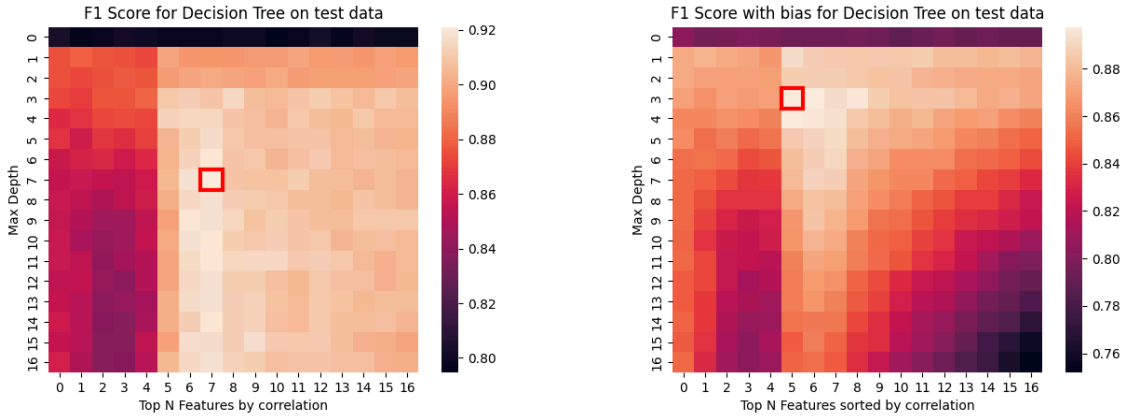
	precision	recall	f1-score	support
Large	0.69	0.67	0.68	30
Medium	0.94	0.93	0.94	198
Small	0.79	0.94	0.86	16
accuracy			0.90	244
macro avg	0.81	0.85	0.82	244
weighted avg	0.90	0.90	0.90	244



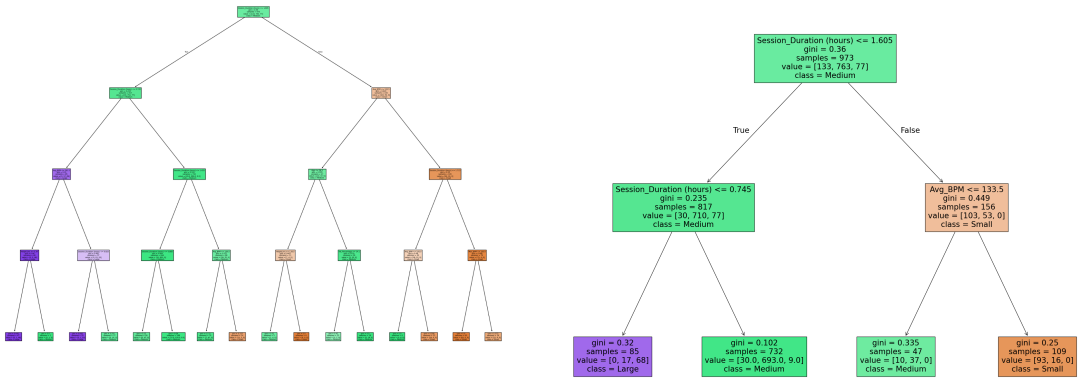
These results were that a neural network would be an acceptable model to use for calorie burnt prediction. Although some changes are needed. The Large calorie category is not performing in proportion to the Medium and Small categories. One idea is the amount of instances in the Small category was disproportionate to the other categories. The number of instances in Small was 77, Medium was 763, and Large was 133. One thing that we could try was to make the bins more equally sized. Overall the neural network performed decently might be suitable for use with some modifications. One thing that we can see immediately from the model visualization is that there are more positive weights. We really did not see any correlations between the edge weights and the decision trees.

2. Decision Tree

The first step in analyzing the effectiveness of the decision tree at many different scales was plotting many comparisons of f1 scores in a heatmap. Each heat map below shows the average f1 score of 100 tested train test splits for each combination of the top N correlated features and the depth of the tree. The red box shows the best average F1 score.



The graph on the left has three main regions, with the top and left region showing models that have too high of a bias because of the small model size. The heat map on the left shows each score biased to take into account the complexity of the model while getting a close score to the maximum. The unbiased maximum had an f1 score of .92 and the model on the right had a score of .91.



The tree on the left is the tree generated from the optimal unbiased parameters and the one on the right uses the same top 6 correlated features (Session_Duration, Experience_Level, Fat_Percentage, Workout_Frequency, Water_Intake, Avg_BPM) but

with only two layers of depth for only a reduction in the f1 score to .89. You can see that the first two levels of the trees are the exact same and mainly use the session duration because of its high correlation with the amount of calories burned. It seems that these features selected track the duration of the workout and the intensity, with some features like water consumed could be for both, as longer and more intense workouts would consume more water, while average heart rate would just code more for intensity.

5. Technical

The exercise data set was fairly clean from the start, the only modification or manipulation that we needed to make was transforming the calories burned column into a categorical representation of the quantitative variable. We binned the calories into the following ranges, small was [0, 500), medium was [500, 1200), and finally [1200, inf).

Decision trees and neural networks are good choices for this analysis because they are effective models at predicting categorical variables.Decision trees are used to iterate through the possible routes and combinations that the model can take. Neural networks are ideal as they are commonly used to predict categorical variables. Back propagation is used to help reduce the error on the model and update the weights.

1. Neural Network

We ran multiple different combinations of hidden layers such as (13, 13, 2, 14, 11)

, (13, 13,14, 11), (13, 14, 11), (13, 11), (17, 14), (9, 9), (5, 5). The best run was (9, 9), with the precision, recall, f1-score, and support results listed above in the results section. The scores went down as we increased the hidden networks. We would have to run more tests to ensure that this trend was accurate.

At the very beginning of writing code, the confusion matrix was read wrong and we were actually using the least correlated variables instead of the most. It took a second to realize this until we compared results and realized that the decision tree results were so much higher than the neural network until we changed the correlated features.

2. Decision Tree

After the features were sorted by the correlation, an optimal model was selected based on the depth and number of top features. The f1 score was calculated from an average of 100 different train/test splits making the number very accurate. The calculation to find an almost as good model for a lot less complexity was used. Specifically subtracting .15*(feature_num/max_features)/(depth/max_depth). This gives the result that is on the corner of the high biased areas of the heatmap or the “elbow” if viewing each dimension in 2d.