Predicting Sleep Quality and Mood: Comparison and Analysis of Machine Learning Methods

Thanh Pham, *Student, Rochester Institute of Technology,*

***Abstract*—Sleep is a vital indicator of overall health. The average human spends up to one-third of their lives asleep. Therefore, it is important to take sleep health seriously. Machine learning and deep learning have been increasing in popularity over the decades. Therefore, it is needed to clarify which methods perform better under which conditions and types of features are needed. Linear Regression, Support Vector Machine, and Perceptron Neural Network are a few of the machine and deep learning techniques that were taken into consideration for this paper. Knowing the advantages and disadvantages of each method would allow for better understanding and prediction of sleep health, specifically the sleep quality and mood.**

I. Introduction

Sleep is a key marker of general well-being and prosperity. We spend up to 33% of our lives sleeping and the condition of our ”sleep health” remains a fundamental question all through our lifespan. Machine Learning and Deep Learning have been increasing in popularity and adoption in recent decades. These methods have the potential to predict trends or classify various objects. In the past decade, many technological advances in hardware and software have made it possible to implement artificial intelligence that could think similarly to a human brain.

In this paper, three widely used methods were carefully chosen, analysis, and compared: Linear Regression that included Least-Square Fit and Gradient Descent with/without regularization, multi-class Support Vector Machine (mult-SVM), and a Perceptron Neural Network. The remaining of this paper is organized as follows. After the introduction, Section II details a brief background into the dataset and the feature selection used for sleep quality prediction. Section III describes each of the methods used and their implementations for the experiment. Section IV presents the results acquired from the experiment and then compared and analyzed against each implemented method. Section V summarizes the comparison and analysis from the results of the experiment and re-iterates the main take-away points from the experiment.

II. DataSet and Features

A. Data

The sleep dataset was collected from the iOS application, Sleep Cycle. The application allows it to export the Sleep Cycle database as a CSV-file, for analysis in programs such as MATLAB. The dataset is a collection of personal sleep features and metadata that dates back to 2015. The sleep features include attributes of the sleep and user’s health, such as start time, duration of sleep, and heart rate.The metadata uses more abstract features, such as mood and sleep quality, generated from user input and a proprietary hidden formula, respectively. This experiment uses a subset The dataset contains 697 samples. Due to unimplemented features of the application for the oldest data points, 208 of 697 samples were missing either the mood or the heart rate. Ideally, these data points would be excluded, but due to it being a small dataset this was corrected by synthesizing random values of mood and between the minimum and maximum heart rate. The dataset was divided with 90% of the samples for training and the 10% for testing. Below is the set of fields for a sleep log in the dataset.

Feature	Type	Description
Start Time	int	Begin time of sleep
End Time	int	End time of sleep
Sleep Quality	float	Overall sleep quality
Duration	int	Total time spent sleeping
Mood	int	Mood at end of sleep
Heart Rate	int	Heart rate at end of sleep
Activity(Steps)	int	Steps taken in a day

B. Feature Extraction

1) *Baseline Features:* The dataset contains a plethora of features types, such as the date time for most of any timing-related features and class-type for mood. Start time, end time, and duration features were processed and converted to a float-type for total minutes ranging from 0 to 1440 to represent 12 AM to 12AM in a day. In order to find accurate relationships between each feature, all features were normalized ranging from 0 to 1.

III. Methods

A. Feature Extraction

1) *Forward Feature Selection:* In total, the final dataset consisted of the top 3 features starting with the most prominent to least, some of which were not relevant to predicting sleep quality or mood.

   a) *Mood Prediction Features:*

Feature	Type	Description
End Time	int	End time of sleep
Heart Rate	int	Heart rate at end of sleep
Activity(Steps)	int	Steps taken in a day

   b) *Sleep Quality Prediction Features:*

Feature	Type	Description
Duration	int	Total time spent sleeping
Start Time	int	Begin time of sleep
End Time	int	End time of sleep

B. Regression

1) *Least-Squares Fit (LSF):* Least-Squares Fit is the most common type of linear regression that can fit both lines and polynomials. Before a relationship between pairs of quantities can be generated, it's important to perform a correlation analysis between two features to determine whether a linear relationship exists. If the relationship is non-linear, then a correlation analysis cannot detect it.
2) *Gradient Descent (GD):* Gradient Descent is a type of linear regression with multiple variables that attempts to minimize the cost function based on the model's variables. It uses an appropriate learning rate that takes steps in the direction of the slope. This method will always converge to a local minimum that best accurately fits high dimensional problem.
3) *Regularization:* Regularization is a tuning parameter that tweaks a model's complexity for over-fitting or under-fitting predictions. There are various types of regularizers, such as Best Subset, Lasso, Ridge, Elastic Net, and infinity. For this experiment, Lasso regularization taken in consideration to minimize the sum of absolute value of coefficients.

C. Classification

1) *Multi-Support Vector Machine (multi-SVM):* Support Vector Machines are one of the most common and widely used in early methods of learning. The concept of a SVM is that a separating hyperplane is constructed that can separate two classes by maximizing the margin which is the distance between the support vectors which are the boundaries of two classes, regardless of the number of dimensions the hyperplane is on. Traditionally, a SVM can only split two classes making it a binary classifier which would not work in this experiment. However, a SVM can be converted to a multi-class SVM by enabling specific coding methods for it to use. There are many coding methods, but the three most common ones are one-versus-all(OVA), one-versus-one(OVO), and ordinal. OVA assigns one class as positive while the rest are negative. This method exhausts all combinations of class assignments. OVO assigns one class as a positive, another as a negative, and the rest are ignored. This method exhausts all combinations of class pair assignments. Ordinal assigns the first class as negative and the rest are positive. The next iteration, a second class is assigned negative. This repeats until there are no more positive assignments.
2) *Perceptron Neural Network:* A Perceptron NN is one of the simplest NNs which consists of a single layer that can train its weights and biases to output a correct target vector based on the input vector. The single layer takes in an input of n features with n number of weights and outputs

IV. Results

The mentioned methods were used to predict and classify sleep quality and mood, respectively, in the previously described dataset. The dataset consists of the following 3 classes: sad, neutral, and happy; While, the sleep quality was a percent represented as a decimal value between 0 to 1. Each method was trained and tested using the 90/10 split.

A. Regression

1) *Least-Squares Fit:*

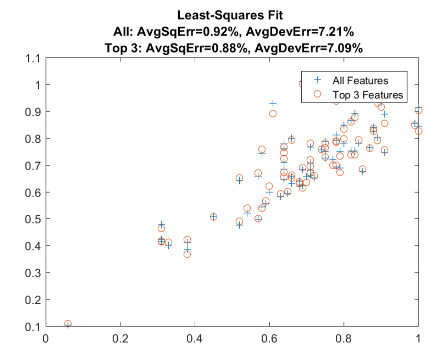

Fig. 1. Least-Squares Fit

The LSF method of linear regression is shown in Fig 1. The top 3 features were clustered more closely than when all features are considered. As demonstrated, The average errors for top 3 features were slightly lower and showed that LSF performed better when only the most important features were accounted for. However, in retrospect, the accuracy would be equivalent to 99.08% and 99.12%. This seems to be very high and the LSF has a high probability of it over-fitting the data.

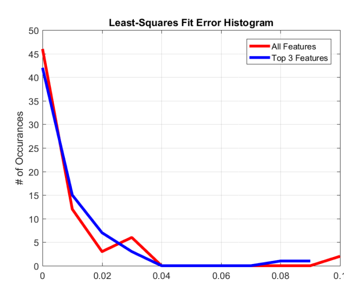

Fig. 2. Least-Squares Fit Histogram

In Fig. 2, the LSF error histogram of the top 3 features was slightly lower than the LSF of all features. However, there is not much difference between the two because the LSF is over-fitting the data.

2) *Gradient Descent:*

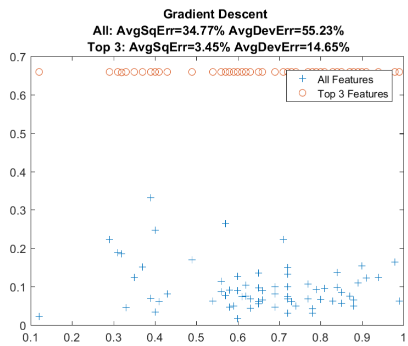

Fig. 3. Gradient Descent

The GD method of linear regression is shown in Fig 3. The top 3 features were clustered more linearly while the all-features plot was very broad and scattered. The top 3 features improved the average square error by over 90% which correlates with how the data points are plotted. The accuracy is 65.23% for all features while 96.56% for the top 3 features. Overall, having less features seemed to improve the model's prediction accuracy.

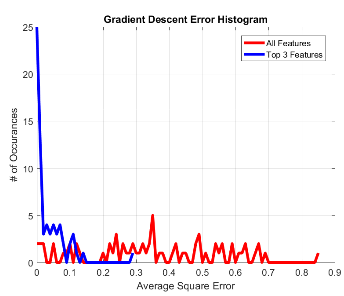

Fig. 4. Gradient Descent Histogram

In Fig. 4, the GD error histogram of the top 3 features was overall much lower compared to the histogram of all features. There were more occurrences of lower average square error between the ground truth and the predicted values of the top 3 features. When compared to all features, not only is there more variance in its error, but the accumulation of error is much greater.

3) *Regularization:*

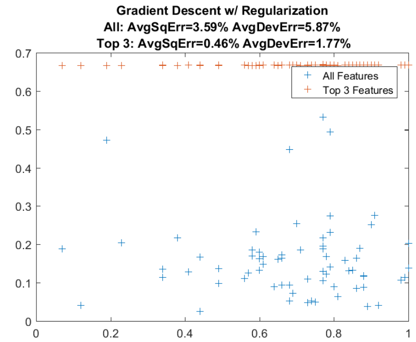

Fig. 5. Gradient Descent Error

In Fig. 5, again, the top 3 features when plotted had a consistent linear relationship. Lasso regularization was added and it was found that the most optimal $\lambda = .001$. The smaller the λ, the less regularization was needed in order to avoid over-fitting the data.

B. Classification

1) *Multi-Support Vector Machine (multi-SVM):*

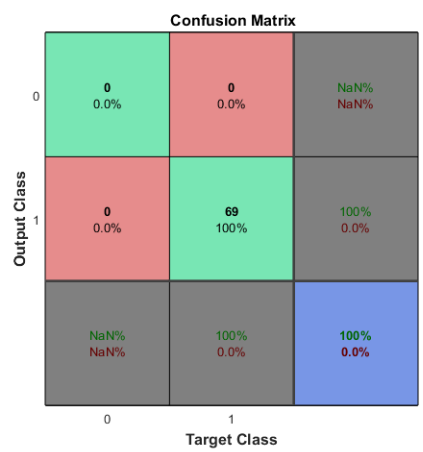

Fig. 6. Gradient Descent Error

In Fig. 6, due to improper implementation, the multiSVM confusion matrix reported 69 of 69 moods as accurate and showed that the accuracy was 100%. The implementation classified two classes rather than three which was incorrect. However, in earlier implementations, the accuracy of the multiSVM was between 45-70%.

2) *Perceptron Neural Network:*

Fig. 7. Gradient Descent Error

In Fig. 7, due to improper implementation, the Perceptron neural network confusion matrix included all 697 samples as moods. It shows a 5.9% accuracy which was very low for a deep learning method.

V. Conclusion

In conclusion, the comparison and analysis provided an overview of various machine learning and deep learning techniques and their application to predicting sleep quality and classifying moods. The main take-aways from the analysis can be summarized in the following points:

- Gradient Descent as the method for linear regression performed better than the Least-Square Fit method. However, both methods performed shockingly well, but was most likely due to over-fitting.
- Regularization of the generated model should be taken into consideration to reduce over-fitting of data points.
- Deep learning methods should perform better than traditional machine learning when implemented properly.

It was discovered that out of all of the 6 features, the top 3 features had an important effect on the sleep quality and mood. For a better sleep quality, the duration of the sleep mattered the most, then the starting time of sleep and lastly, the ending time of sleep. Intuitively, this makes sense because usually, the longer you sleep, the better sleep quality you should get especially when you go to bed. However when predicting the mood, the top 3 features change to the end time of sleep, heart rate, and the number of steps taken in a day. These features also intuitively make sense when looking at the end time of sleep. For this dataset, the end time of sleep is usually in the morning, but if the end time was during the night, then prediction of mood could drastically change. Overall, the comparison and analysis of various techniques gave us new insight on predicting overall sleep health. In the future, more data should be considered especially from other testers. This dataset was tracked specifically for one person.