Age and Screen Time: Identifying Predictive Factors Through Statistical Modeling

Cody Coleman

May 20th 2025

1 Introduction

This project's main goal is to predict data related towards either a person's screen time or what a person's age is because of certain factors, which will be mentioned in the data selection below. The reason why these data were selected was because of one major reason. This project was chosen because of how much I and the rest of the world use screens everyday. I personally have a minimum of 12 hours a day when I am looking at a screen due to school or entertainment. No matter what you are doing in this day and age, you will always be looking at a screen, which is why I wanted to predict what factors contribute to the use of screens. The project outlined the need for three separate models. The three models chosen were one grid search model using random forest regression, one support vector machine, and the other model is a regression-based neural network because the data is binary.

2 Data

The data set that was selected was taken from a free online resource called Kaggle. The data was then cleaned by first dropping a data set that was the gender of the person, as well as their id number. The gender was not super important to the study, so the data was dropped. From there all strings were transformed into integers by using a function. All columns used during the study go as follows: `age`, `gender`, `daily_screen_time_hours`, `phone_usage_hours`, `laptop_usage_hours`, `tablet_usage_hours`, `tv_usage_hours`, `social_media_hours`, `work_related_hours`, `entertainment_hours`, `gaming_hours`, `sleep_duration_hours`, `sleep_quality`, `mood_rating`, `stress_level`, `physical_activity_hours_per_week`, `location_type`, `mental_health_score`, `uses_wellness_apps`, `eats_healthy`, `caffeine_intake_mg_per_day`, `weekly_anxiety_score`, `weekly_depression_score`, `mindfulness_minutes_per_day`. The two target variables were `daily_screen_time_hours` as well as `age`.

2.1 Total Screen-time

This section goes into the target variable daily_screen_time_hours. There were three models that were used in this section, which were mentioned above. Two grid searches, as well as a neural network. When testing these models, the model was told to reserve 20% of it's data for testing. As well as having a seed of 1218727. For the decision tree regressor grid search model, the params were as follows: squared_error, friedman_mse, absolute_error, poisson. The max depth was a list of: [2, 4, 5, 6]. When predicting using this model, the root mean square error was 1.9665946461633514. Which when compared to a real value such as 6.8, it would output a predicted value of 5.972869147659058. The next model was a neural network. The neural network's input shape remained constant, staying at 23, due to the x.shape staying the same. For the metrics section, the model used "mae". The neural network has 1000 epochs, but there is a callback function to stop early, with a patience of 2. There was a total batch size of 10. The actual model went as follows, Input, Dense Hidden Layer, Dropout, another Dense layer, another dropout function, then one last dense hidden layer, with the output layer being a dense layer. The first dense layer had 256 total neurons, with an activation of "relu". The first dropout function dropped 30% of the active neurons, so that the data would not overfit. The second dense layer contained 128 neurons, with another activation of "relu". The next dropout function dropped 20% of the neurons. The last dense hidden layer had 64 neurons with relu as the activation. The output layer had one neuron due to the data being binary, with an activation of "linear". When compiling the model, the optimizer was "adam" and the loss function also used "mae". The summary concluded that there were 47,361 total parameters. When fitting the data, the root mean square error was 2.3109510247381, and the real value of 3.7 was close to the predicted value of 4.557. But when another example was checked, the real value was 8.7, while the predicted value turned out to be 5.243303, which was a good amount off from the original number. The last model using the daily_screen_time_hours target class was a Support Vector Regressor, which consisted of one "kernel", which was "linear". The next parameter "gamma" used both "scale" and "auto". Lastly, the model had three numbers in the "C" parameter which were: [0.1, 1, 10]. The Support Vector Regressor used a grid search as well for fitting. When predicting the model, the root mean squared error turned out to be 1.9744543754612025. When predicting compared to the real value, the predicted value turned out to be 5.7818690568469835 while the real value was 6.8. The model did however stay pretty consistent. The model predicted a value of 5.633341838906339 when the real value was 7.7, and also it predicted a value of 6.226171207241344 when the value was 7.4, so the model was not terrible compared to some of t he others.

2.2 Age

This other section goes into the other target variable mentioned age. The models used above were also used down here as well, but the parameters of each were switched up a bit. Compared to the first the criterion stayed the same, staying as: `squared_error`, `friedman_mse`, absolute_error, `poisson`. The max depth has changed to: [1, 3, 4, 5]. When training the decision tree regressor, the root mean square error was 14.84519351313976. When comparing one of the predicted points it was pretty accurate, but when checked against other different real points, the prediction stayed between 30 and 40 no matter the data-point. This could be from overfitting, which in future iterations could be changed. The second neural network's metrics changed to "mse" compared to the first, but the loss stayed the same at "mae". The amount of epochs was dropped to 100, while the batch size was increased to 100. The input shape remained the same, but the first hidden layer was changed. The activation remained the same throughout all of them, but the neuron count for the first hidden layer changed to 128. The second hidden layer's neuron count changed to 10, same with the third. The output layer remained the same as well. The model still used the optimizer "adam" though. When fitting, the model had a value loss of 13.3463 and a root mean square error of 15.69661808013916. Compared to a real value of 37, the model predicted it would be 39.57062. But when tested against another value of 63, it predicted 43.42427. The model was semi-accurate but would just stay between 36 and 45. The last model was another grid search using a support vector machine. The kernel took in "linear" just like the other support vector machine. The model also took in "scale" and "auto" for "gamma" as well. The "C" did change however to: [0.5, 1.5, 10.5]. The model's root mean square error is 1.974567057663765. When checking the model's real values compared to it's predicted values, it was accurate. The first real value tested was 5.2, the model predicted 6.235715684326707. The second real value was 5.9 and it predicted 5.778250744781318. The last real value was 7.6 and it's last prediction was 6.253989822460782.

3 Conclusion

The goal behind this project was to predict two different target class variables. Those two target class variables being: `daily_screen_time_hours` and `age`. I used six total models, four of them being grid searches, while the other two were neural networks. Two of the four grid searches were support vector regressors, while the other two were random forest regressors. The best model within the `daily_screen_time_hours` target class was the random forest regressor with a root mean square error of 1.9665946461633514. The best model within the `age` target class was the support vector machine with a root mean square error of 1.974567057663765. Both models predicted way better than the others in comparison. Although, most of the better models were within the `daily_screen_time_hours`, compared to `age` target class.