University of Hawaii at Manoa

Predicting Daily Sleep Duration Using Activity and Demographic Features from the 2023 American Time Use Survey

Contents

1. Introduction 1
  1.1. Background and Motivation 2
  1.2. Problem Statement 2
  1.3. Objectives of the Study 2
  1.4. Significance of the Study 3
2. Literature Review 4
3. Data Description 5
  3.1. Data Source 5
  3.2. Outcome Variable 5
  3.3. Predictor Variables 6
  3.4. Feature Engineering 6
4. Methodology and Model Evaluation 10
  4.1. Data Splitting and Preprocessing 10
  4.2. Linear Model 10
    4.2.1. Model Selection Approach 13
  .4.3 Nonlinear Model 21
    4.3.1 Neural Networks 21
    4.3.2 Neural Network Architecture 21
    4.3.3 Neural Network Parameters 21
    4.3.4 Training Results and Discussion 22
    4.3.5 Discussion and Future Works 23
5. Conclusion 25

Figure 1: Model 1 regularization parameter selection.. 14
Figure 2: Model 2 regularization parameter selection.. 15
Figure 3: Model 3 regularization parameter selection.. 15
Figure 4: Model 3 result on test data.. 16
Figure 5: Lasso ranking for Base Model (Model 3).. 17
Figure 6: Model 3_9 result on test data... 18
Figure 7: Change in correlation value form model M3_1 to M3_9................................ 19
Figure 8: Improvement in metric from Model M3_1 to M3_9.. 19
Figure 9: Improvement in prediction on test data from model M3_1 to M3_9............... 20

Table 1: Variables' categories and classification 7
Table 2: Comparison of Models M1 to M3 14
Table 3: Comparison of Models M3_1 to M3_9 17
Table 4: Neural network parameters 22
Table 5: Neural Network Results 23

1. Introduction

In the first chapter we discuss background and motivation, reviewing some of the most important previous works in this area, as well as introducing completely our problem and its objectives.

For this study, we collected and processed all available 2023 microdata files from the **American Time Use Survey (ATUS)**, including the **atusact**, **atussum**, **atusresp**, and **atuscps** datasets. Each of these files offers a different layer of information about respondents' daily activities, personal demographics, and household characteristics. Before conducting any analysis, we thoroughly reviewed the official codebooks and technical documentation to ensure a clear understanding of the variable definitions, activity coding structures, and the relationships between different datasets.

After importing the data into R, we used the tidyverse package for data cleaning and transformation. Our initial focus was on the atusact file, which records episode-level activity data for each respondent over a 24-hour period. Each activity is coded using a hierarchical scheme and is accompanied by a duration variable (TUACTDUR24) and a social context variable (TUWHO_CODE), which indicates whether the activity was performed alone or with others. We filtered this file to identify and extract durations for several key categories of activities, including exercise (code 1301), travel (codes beginning with 18), sleep (code 010101), work (codes beginning with 0501), internet use (codes 160101, 150101, 120307, 120308), and leisure (codes 120307 and 120308). Travel activities were further classified by purpose, distinguishing between travel for work (1505), socializing (181201), entertainment (181205), and exercise (181301).

For each of these activity types, we calculated the total time each respondent spent per day and also disaggregated this time based on the social context. Specifically, we separated each duration into "alone" time (when TUWHO_CODE was 0) and group time (when `TUWHO_CODE` was greater than 0). This allowed us to assess both the quantity and the social nature of time use across all major activities.

Once we had aggregated daily activity durations for each respondent, we enriched the dataset by merging in demographic and socioeconomic characteristics from the other ATUS files using the common identifier TUCASEID. From the atussum file, we extracted age, gender, and education level. The education variable, originally reported in 16 detailed categories, was recoded into three broad levels: low, medium, and high. From **atusresp**, we added employment status, whether the diary day was a weekday or weekend, household size, the number of adults in the household, the number of children under 18, and household type (e.g., single-family home, apartment, dormitory). The **atuscps** file provided information on household income, which was available in grouped brackets rather than exact dollar amounts.

The final product of this process was a comprehensive, respondent-level dataset named D_PER_DAY. Each row in this dataset represents one respondent and includes total, alone, and group durations for each key activity type, along with demographic and household-level variables. We ensured that categorical variables were recoded into readable formats, addressed missing values where necessary, and verified the consistency and integrity of merged data. This structured dataset forms the foundation for analyzing patterns of time use behavior across different segments of the population, enabling detailed exploration of how daily routines vary by age, gender, education, employment status, income level, and household composition.

1.1. Background and Motivation

The widespread adoption of digital technologies has significantly transformed how individuals allocate their time across daily activities. With online entertainment, remote work, and virtual communication becoming common, traditional behaviors such as commuting, physical socialization, and even sleep are now subject to substitution by digital alternatives. This shift presents a challenge for transportation and urban planners: as travel becomes less necessary for certain functions, how does time get reallocated? Sleep, in particular, may serve as a critical indicator of lifestyle balance or time stress.

1.2. Problem Statement

Most transportation models emphasize physical trips while overlooking non travel behaviors that fill people's schedules. Digital engagement and lifestyle trade offs such as spending time online instead of sleeping are not typically captured in travel demand forecasting. This study addresses that gap by asking: Can we predict how individuals allocate time to sleep (and secondarily, internet use and exercise) using behavioral and demographic features from the 2023 ATUS dataset?

1.3. Objectives of the Study

- Predict total daily sleep time ($SLEEP\_DUR$) using features from ATUS 2023.
- Incrementally test which features (time-use and contextual) best improve model accuracy.
- Compare linear (Lasso) and nonlinear (Neural Network) models.
- Avoid overfitting and data leakage by carefully selecting duration features Interpret the behavioral significance of selected variables.

1.4. Significance of the Study

- Demonstrates that time-use trade-offs (e.g., internet vs. sleep) can be modeled and quantified.
- Provides insights for travel demand modeling in a post digital substitution world.
- Supports responsible machine learning by avoiding tautological patterns in time budget data.

2. Literature Review

Digital Leisure Time and Other Activities: Dong et al. (2018) analyzed ATUS 2013 data to understand how computer based leisure (like social media use) fits into overall leisure time. They defined six types of leisure (home vs. outside, computer vs. not, etc.) and used a joint discrete continuous model to capture both the choice of leisure activity type and the amount of time spent. A key finding was that increased digital leisure (time on computers) did not significantly substitute for other leisure activities. Instead, time spent online was influenced by factors such as income, number of children, employment status, and differed between weekdays and weekends. This suggests technology is adding new dimensions to leisure time without simply replacing traditional activities. Joint Activities and Companionship: Srinivasan & Bhat (2008) conducted an exploratory analysis of joint activity participation using ATUS data. They examined how often and how long people engage in activities with household vs. non household companions, across different activity purposes and days of week. Their study found that activities done with others tend to last longer and often occur at someone else's home, highlighting the importance of social context in time use patterns. Free Time and Physical Activity: Houghton et al. (2019) studied the link between discretionary free time and physical activity using ATUS data. They focused on Americans aged 15 or older and found that the amount of free time is positively associated with the likelihood of engaging in exercise. This research provides insight into how time constraints (or the lack thereof) can influence healthy behaviors. These studies underscore the richness of ATUS data for understanding time allocation and inform our approach. In particular, they motivate our inclusion of variables like employment, family structure, and weekday/weekend indicators, as well as separate accounting of activity context (alone vs. with others) in our predictive modeling.

3. Data Description

In this section the data completely is introduced, and difference between various type of variables and our approach toward handling them are discussed.

3.1. Data Source

This study uses the American Time Use Survey (ATUS) 2023 data, which includes:

1. **atusact_2023:** Contains one row per activity recorded by respondents. Includes start time, stop time, activity codes (e.g., sleeping, working), and location. Also links to who was present during the activity and whether the respondent was also working.
2. **atuswho_2023:** Lists who was present during each activity (alone, with household children, spouse, non-HH members, etc.). Linked via activity line number. One row per person present per activity.
3. **atuscps_2023:** Derived from the Current Population Survey conducted 2–5 months before the ATUS interview. Contains demographic and labor force characteristics, including: Education level ($PEEDUCA$), Family income ($HUFAMINC$), Occupation & industry, Household composition, Marital status, Employment status (full/part-time), Race, Hispanic origin, citizenship, Veteran status, disability indicators, Geographic info (region, metro status)
4. **atussum_2023:** Aggregated totals per respondent, such as: Total daily minutes spent in each activity category (e.g., work, leisure, travel, childcare), Summary by tier (ATUS activity code system) Useful for quick modeling without full activity detail.
5. **atusresp_2023:** Contains ATUS interview results per respondent, such as: Diary day of the week, Respondent weight, Who completed the diary, Employment status ($TELFS$), Age, gender ($TESEX$), Household size, presence of children Often used in combination with CPS file for analysis

These datasets were merged via the unique respondent ID ($TUCASEID$) to create a comprehensive file (D_PER_DAY) with behavioral and demographic variables.

3.2. Outcome Variable

SLEEP_DUR (minutes/day): The main continuous target variable for prediction. While other variables like INT_DUR (internet use) and EXE_DUR (exercise) were engineered, they are only used as predictors, not as additional outcomes (for first step).

3.3. Predictor Variables

Continuous variables, such as age, number of adults, number of children, and time-related variables, can take on any value within a given range and are measured on a continuous scale. In contrast, categorical variables, such as level of education, income brackets, and type of housing, represent qualitative attributes. These variables take on a limited number of discrete values that serve as proxies for different categories. For example, an income below $5,000 might be represented by a value of 1, while an income between $15,000 and $20,000 could be represented by a value of 6. These numeric codes do not carry inherent quantitative meaning and are simply labels for different groups. To appropriately handle categorical variables in modeling, we apply one-hot encoding, which transforms each category into a separate binary feature, allowing machine learning algorithms to process them without assuming any ordinal relationship between the encoded values.

Furthermore, it is important to distinguish between time-related variables and demographic variables for future analysis. **Table 1** contains the list of continuous and categorical variables, along with their corresponding classifications.

3.4. Feature Engineering

Our input dataset as it is, cannot be used for two main reasons. First of all, the categorical variables should be encoded in our algorithm with the approach of one-hot-encoding. Below we explain how each of these variables are treated to make a proper input dataset.

1. $\boldsymbol{DIARYDAY}$**:** This variable has 7 levels and by one-hot-encoding we replace it by a variable named $\boldsymbol{WEEKD}$ which is 0 if the $\boldsymbol{DIARYDAY}$ is Saturday ($DIARYDAY = 1$) or Sunday ($DIARYDAY = 7$), and otherwise it is equal to 1.

2. $\boldsymbol{HOUSE}$**:** This variable has 12 levels and by one-hot-encoding we replace it by a variable named $\boldsymbol{PERSHOU}$ which is 1 if the housing type is code 1 (permanent house), and 0 otherwise.

3. $\boldsymbol{INC}$**:** This variable has 16 levels and by one-hot-encoding we replace it by two variables named $\boldsymbol{MIDINC}$ which is 1 if family income is in the mid-range (codes 9

to 13), else 0, and $\boldsymbol{HIGHINC}$ which is 1 if income is in the high range (codes 14 to 16), else 0, and base as $\boldsymbol{LOWINC}$**.**

4. $\boldsymbol{EMP\_STAT}$**:** This variable has 5 levels and by one-hot-encoding we replace it by a variable named $\boldsymbol{EMPLOYED}$ which is 0 if the respondent is already employed (code 1), else 0.
5. $\boldsymbol{GENDER}$**:** This variable has 2 levels and by one-hot-encoding we replace it by a variable named $\boldsymbol{MALE}$ which is 1 if the respondent is male, else 0.

Table 1: Variables' categories and classification

Categories	**Classification**	**Variable**
Continuous	**Time-Related**	$TRAVEL\_DUR$
		$TRAVEL\_WORK$
		$TRAVEL\_SOCO\_DUR$
		$TRAVEL\_ENT\_DUR$
		$TRAVEL\_EXE\_DUR$
		$OTHER\_TRAVEL\_DUR$
		EXE_DUR
		$ALONE\_EXE\_DUR$
		$GROUP\_EXE\_DUR$
		INT_DUR
		$ALONE\_INT\_DUR$
		$GROUP\_INT\_DUR$
		$WORK\_DUR$
		$ALONE\_WORK\_DUR$
		$GROUP\_WORK\_DUR$
		LEI_DUR
		$ALONE\_LEI\_DUR$
		$GROUP\_LEI\_DUR$
		$OTHER\_DUR$
	Demographic	AGE
		$ADULT$
		$CHILDREN$
Categorical	**Demographic**	SEX
		EDU
		EMP_STAT
		$HOUSE$
		INC
		$DIARYDAY$

When our dataset includes categorical features, there are two main strategies we can adopt to incorporate them into our models. Assume we have one continuous feature ($TRAVEL\_DUR$) and one categorical feature (SEX ,where if $SEX = 1$ if the respondent is Male, $SEX = 2$ if Female).

1. **Exogenous Approach:** In this method, we split the dataset based on the values of a categorical variable and build separate models for each subset.
   - MALE Dataset: $SLEEP_{DUR} = \beta_{0_{MALE}} + \beta_{TT_{MALE}}(TT)$
   - FEMALE Dataset: $SLEEP_{DUR} = \beta_{0_{FEMALE}} + \beta_{TT_{FEMALE}}(TT)$
2. **Endogenous Approach:** In this more integrated method, we Encode the categorical variable using dummy coding (e.g., create a $MALE$ feature equal to 1 if $SEX = 1$, 0 otherwise). Then drop the original SEX column, and fit a single regression model on the entire dataset, including interaction terms if necessary.

   $$SLEEP_{DUR} = \beta_0 + \beta_{TT}(TT) + \beta_{MALE}(MALE) + \beta_{TT*MALE}(MALE * TT)$$

   Now, if the respondent is male ($MALE = 1$):

   $$SLEEP_{DUR} = (\beta_0 + \beta_{MALE}) + (\beta_{TT} + \beta_{TT*MALE})(TT)$$

   and if the she is female ($MALE = 0$):

   $$SLEEP_{DUR} = \beta_0 + \beta_{TT}(TT)$$

Comparing these approaches, it turns out that:

$$\beta_{0_{MALE}} = (\beta_0 + \beta_{MALE})$$

$$\beta_{TT_{MALE}} = (\beta_{TT} + \beta_{TT*MALE})$$

$$\beta_{0_{FEMALE}} = \beta_0$$

$$\beta_{TT_{FEMALE}} = \beta_{TT}$$

Both exogenous and endogenous approaches lead to the same parameterization. However, the endogenous approach is more robust, especially when handling multiple categorical variables, and allows us to use interaction terms flexibly. Therefore, for the remainder of this project, we adopt the endogenous approach.

Secondly, there is a linear dependency between some of the columns in our input dataset. In the future we make decision on which columns should be used when there is such dependency between them.

$EXE\_DUR = ALONE\_EXE\_DUR + GROUP\_EXE\_DUR$

$WORK\_DUR = ALONE\_WORK\_DUR + GROUP\_WORK\_DUR$

$INT\_DUR = ALONE\_INT\_DUR + GROUP\_INT\_DUR$

$$LEI\_DUR \; = \; ALONE\_LEI\_DUR \; + \; GROUP\_LEI\_DUR$$

$$TRAVEL_{DUR} = \; TRAVEL_{WORK_{DUR}} + TRAVEL_{SOCO_{DUR}} + TRAVEL_{ENT_{DUR}} + TRAVEL_{EXE_{DUR}} \\ + OTHER\_TRAVEL\_DUR$$

In the next section we introduce our methodology, different models (linear and nonlinear), and finally discuss how we have come up with our final model.

4. Methodology and Model Evaluation

We address this problem using two primary modeling approaches: a linear model (Linear Regression with Lasso regularization) and a nonlinear model (Neural Network).

Linear models are limited in their ability to capture interactions between continuous and categorical variables. To address this limitation, we explicitly include relevant interaction terms in the input dataset for the linear model, for example, the interaction between weekday and travel duration (e.g., $WEEKD * TRAVEL\_DUR$). Additionally, the linear model cannot rely solely on demographic variables; it requires the inclusion of certain time-related features to achieve adequate performance.

The primary objective of employing the nonlinear model is to assess whether these manually constructed interaction terms and time-related features can be omitted without sacrificing predictive accuracy, as neural networks are capable of capturing complex, nonlinear relationships implicitly.

4.1. Data Splitting and Preprocessing

Since the magnitudes of different feature values vary significantly, we normalize the data using the standard scalar method, which normalize the feature values based on their mean and variance in a way that they all have mean of zero and standard deviation of 1.We should keep in mind that categorical variables should not be normalized and test set should remain intact during the normalization phase as well as training phase.

We distribute our data across three sets: training, validation, and test with a common split of 70% training set, and 15% each of the validation and test set.

4.2. Linear Model

Linear regression is a fundamental supervised learning algorithm used to model the relationship between a dependent variable y and an independent variable x. To improve the model's generalization and prevent overfitting, we use **regularization**, which adds a penalty term to the cost function.

A linear regression follows the equation:

$$y = wx + b$$

where y is the predicted output, x is the input feature, w is the weight, and b is the bias.

To measure the error between predicted values ($\hat{y}$) and actual values (y), we use the Mean Squared Error (MSE):

$$without\ regularization: \qquad J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y)^2$$

where m is the number of training examples.

We add a penalty term to the cost function, L2 (Ridge) regularization helps control large weight values. L1 (Lasso) regularization encourages sparsity (some weights become exactly zero).

$$Ridge\ Regression: \qquad J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y)^2 + \frac{\alpha}{2m}\sum_{j=1}^{n} w_j^2$$

where n is the number of input features, α is the regularization parameter controlling the penalty strength. This method keeps all weights but shrinks them toward zero.

$$Lasso\ Regression: \qquad J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y)^2 + \frac{\alpha}{m}\sum_{j=1}^{n}\left|w_j\right|$$

In this method, the absolute value penalty forces some weights to become exactly zero, performing automatic feature selection. It is useful when there are many irrelevant or redundant features.

For gradient computation, L1 gradients are not differentiable at zero, so sub-gradient methods are typically used in practice for optimization. On the other hand, for L2 we have:

$$\frac{\partial J}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)x_i + \frac{\alpha}{m}w$$

$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)$$

Using a learning rate γ, we update the parameters:

$$w \coloneqq w - \gamma\frac{\partial J}{\partial w}$$

$$b \coloneqq b - \gamma\frac{\partial J}{\partial b}$$

This process is repeated until convergence. Once training is complete, we can use the learned parameters (w) and (b) to predict new values:

$$\hat{y} = wx + b$$

Now, the question here is which regularization we should use for our problem? We chose Lasso since:

- We have many potentially redundant or collinear activity features.
- We've also created interaction terms, which can explode feature dimensionality.
- Some features are likely irrelevant for predicting $SLEEP\_DUR$.

Lasso is the **Better Choice** since it performs automatic feature selection by shrinking some coefficients to exactly zero.

Now, a brief introduction of evaluation metrics seems necessary. The following, are the evaluation metrics we used for Linear Models:

1. Mean Squared Error (MSE): It measures the average squared difference between predicted and actual values. It penalizes large errors more strongly than small ones, and used as the loss function in linear regression models.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

2. Mean Absolute Error (MAE): It measures the average of the absolute errors. It is more robust to outliers than MSE, and represents the average distance between predicted and actual values, in the same unit as the target.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|\hat{y}_i - y_i|$$

3. Coefficient of Determination (R^2): It represents how well the model explains the variance of the target. If equal to 1 indicates perfect prediction, if equal to 0 means the model is no better than predicting the mean, and if less than 0 implies the model is worse than predicting the mean.

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{m}(\bar{y}_i - y_i)^2}$$

where, $\bar{y}$ is the mean of the true target value.

4.2.1. Model Selection Approach

In this project model selection was done in two levels. During each selection level, we used the validation set to make decision on the regularization parameter, each model was trained and then tested on validation set for different values of regularization ($\alpha \in \{0.001, 0.01, 0.1, 1, 10, 100\}$). After that, the model was used to make prediction on test data, and based on the metrics value, the final model was proposed.

4.2.1.1 First-Level Selection

At first level (choosing between Model1, Model2, and Model3) we need to make decision about the issue that among independent features which one of them should be used. We began by evaluating three initial models that only used aggregated duration features. In the dataset, several features are sums of more granular ones, prompting us to investigate whether it's better to use the aggregated or split versions. There some features that are in common between all these three models:

$WEEKD, MIDEDU, HIGHEDU, PERSHOU, MIDINC, HIGHINC, EMPLOYED, MALE, ADULT, CHILDREN$

The key difference between the models is stem from:

$EXE\_DUR = ALONE\_EXE\_DUR + GROUP\_EXE\_DUR$

$WORK\_DUR = ALONE\_WORK\_DUR + GROUP\_WORK\_DUR$

$INT\_DUR = ALONE\_INT\_DUR + GROUP\_INT\_DUR$

$LEI\_DUR = ALONE\_LEI\_DUR + GROUP\_LEI\_DUR$

$$TRAVEL_{DUR} = TRAVEL_{WORK_{DUR}} + TRAVEL_{SOCO_{DUR}} + TRAVEL_{ENT_{DUR}} + TRAVEL_{EXE_{DUR}} + OTHER\_TRAVEL\_DUR$$

At first level, we define our models:

Model 1:

$$(In - common\ features) + EXE\_DUR + WORK\_DUR + INT\_DUR + LEI\_DUR + TRAVEL\_DUR$$

Model 2:

$$(In - common\ features) + ALONE\_*/GROUP\_*\ versions\ of\ EXE/WORK/INT/LEI + TRAVEL\_DUR$$

Model 3:

$$(In - common\ features) + ALONE\_*/GROUP\_*\ versions + all\ travel\ subcomponents$$

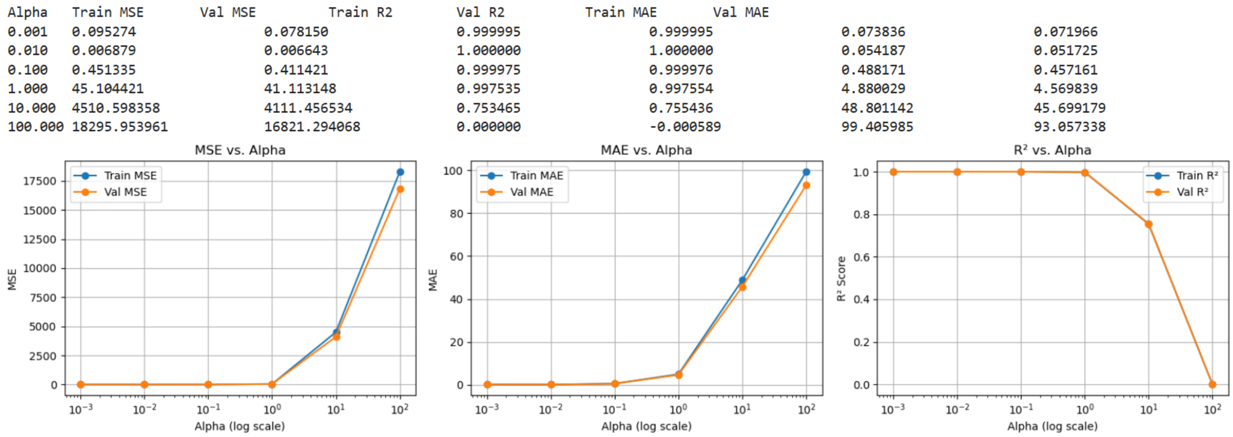
For all these three models, the regularization parameter is set to 0.1. As it is shown in **Figure 1** to **Figure 3**, regularization values smaller than 0.1 has the same performance as 0.1, but after that the performance of the model decreases conspicuously. The performance of these three models on test data is displayed in **Table 2**.

Table 2: Comparison of Models M1 to M3

Model	MSE	MAE	R^2	Correlation
Model M1	0.459788	0.487079	0.999975	1
Model M2	0.868664	0.670249	0.999952	1
Model M3	1.013205	0.714316	0.999944	1

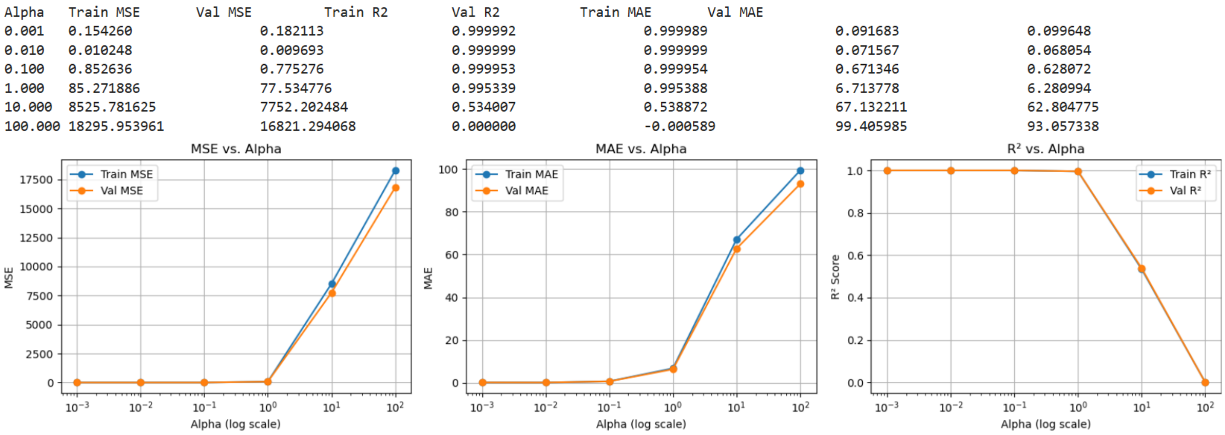
All these models are affected by data leakage because the total time (including $SLEEP\_DUR$ and all_DUR features) sums to 1440 minutes. This leads to perfect or near-perfect correlation with $SLEEP\_DUR$ and unrealistically high model metrics. Although splitting aggregated features slightly degrades performance (making Lasso's job harder), the models are still artificially inflated. We chose to proceed with Model 3 as the base for further development. In **Figure 4**, the result of Model 3 on test data is shown. As it we expected, the model works suspiciously perfect.

In the next section we go through the second loop of model selection.



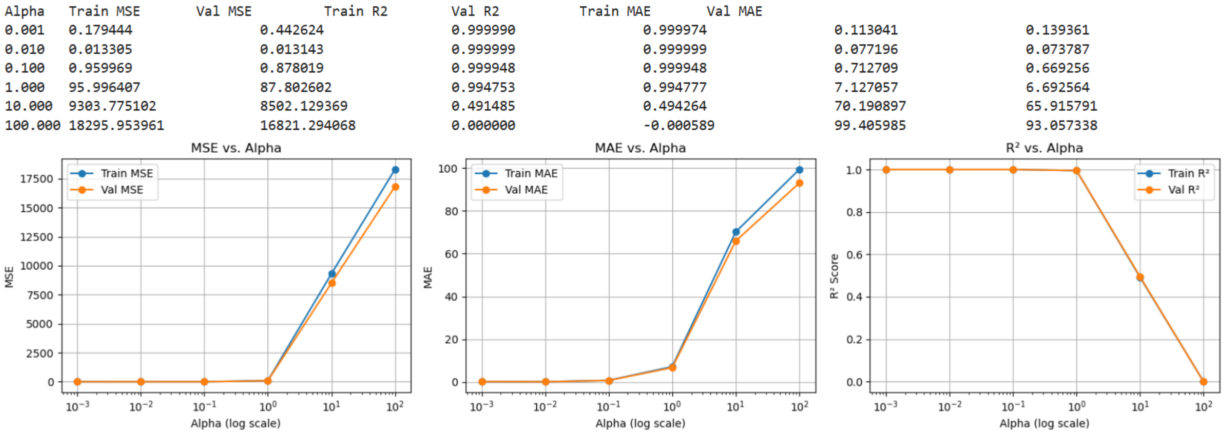
```
Alpha    Train MSE       Val MSE         Train R2        Val R2          Train MAE       Val MAE
0.001    0.095274               0.078150               0.999995               0.999995               0.073836               0.071966
0.010    0.006879               0.006643               1.000000               1.000000               0.054187               0.051725
0.100    0.451335               0.411421               0.999975               0.999976               0.488171               0.457161
1.000    45.104421              41.113148              0.997535               0.997554               4.880029               4.569839
10.000   4510.598358            4111.456534            0.753465               0.755436               48.801142              45.699179
100.000  18295.953961           16821.294068           0.000000               -0.000589              99.405985              93.057338
```

Figure 1: Model 1 regularization parameter selection



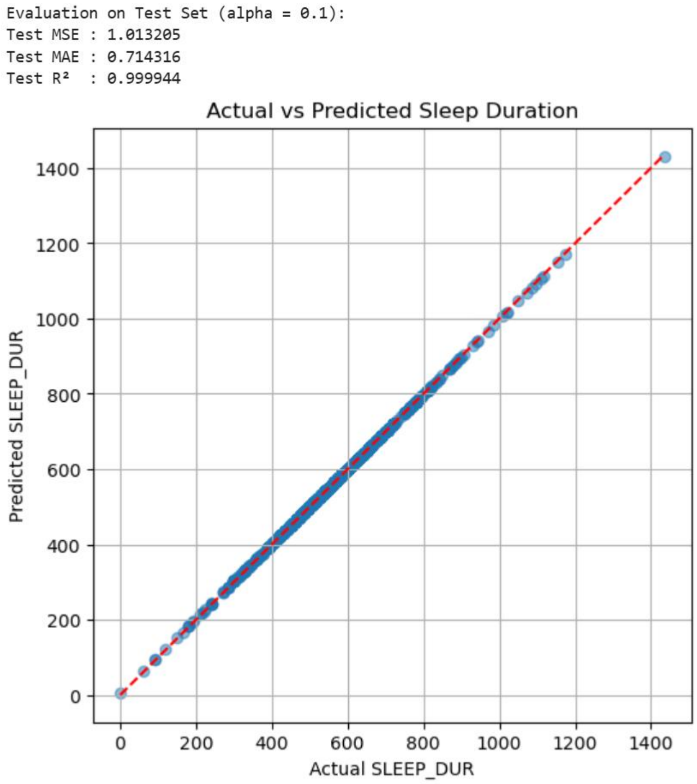
```
Alpha   Train MSE        Val MSE          Train R2         Val R2           Train MAE        Val MAE
0.001   0.154260                 0.182113                 0.999992                 0.999989                 0.091683                 0.099648
0.010   0.010248                 0.009693                 0.999999                 0.999999                 0.071567                 0.068054
0.100   0.852636                 0.775276                 0.999953                 0.999954                 0.671346                 0.628072
1.000   85.271886                77.534776                0.995339                 0.995388                 6.713778                 6.280994
10.000  8525.781625              7752.202484              0.534007                 0.538872                 67.132211                62.804775
100.000 18295.953961             16821.294068             0.000000                 -0.000589                99.405985                93.057338
```

Figure 2: Model 2 regularization parameter selection



```
Alpha   Train MSE        Val MSE          Train R2         Val R2           Train MAE        Val MAE
0.001   0.179444                 0.442624                 0.999990                 0.999974                 0.113041                 0.139361
0.010   0.013305                 0.013143                 0.999999                 0.999999                 0.077196                 0.073787
0.100   0.959969                 0.878019                 0.999948                 0.999948                 0.712709                 0.669256
1.000   95.996407                87.802602                0.994753                 0.994777                 7.127057                 6.692564
10.000  9303.775102              8502.129369              0.491485                 0.494264                 70.190897                65.915791
100.000 18295.953961             16821.294068             0.000000                 -0.000589                99.405985                93.057338
```

Figure 3: Model 3 regularization parameter selection

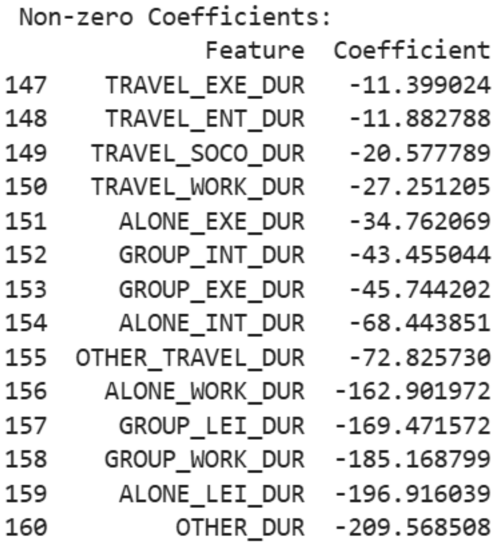


```
Evaluation on Test Set (alpha = 0.1):
Test MSE : 1.013205
Test MAE : 0.714316
Test R²  : 0.999944
```

Figure 4: Model 3 result on test data

4.2.2.2.Second-Level Selection

Starting from Model 3, we removed $all\ \_DUR$ features and created a baseline model with only demographic features. We then used Lasso importance rankings, a One-Factor-At-a-Time (OFAT) approach to incrementally reintroduce the most informative $\_DUR$ features. We stopped once the model's correlation approached 0.90 and R² neared 0.80, to avoid entering the data leakage zone. In figure 5, the lasso ranking of all $\_DUR$ features for the base model (Model 3) is depicted.



```
Non-zero Coefficients:
               Feature  Coefficient
147     TRAVEL_EXE_DUR   -11.399024
148     TRAVEL_ENT_DUR   -11.882788
149    TRAVEL_SOCO_DUR   -20.577789
150    TRAVEL_WORK_DUR   -27.251205
151      ALONE_EXE_DUR   -34.762069
152      GROUP_INT_DUR   -43.455044
153      GROUP_EXE_DUR   -45.744202
154      ALONE_INT_DUR   -68.443851
155  OTHER_TRAVEL_DUR   -72.825730
156     ALONE_WORK_DUR  -162.901972
157      GROUP_LEI_DUR  -169.471572
158     GROUP_WORK_DUR  -185.168799
159      ALONE_LEI_DUR  -196.916039
160          OTHER_DUR  -209.568508
```

Figure 5: Lasso ranking for Base Model (Model 3)

We tested for 9 different model, and for all these models the regularization parameter was set to 1 (for the same reason we explained in **4.2.1.1**). 9 different models (M3_1 to M3_9) was tested and compared. Finally, the model M3_9 was chosen as our final model. **Table 3** includes the comparison between these models and improvement in R^2 value based on using these models to predict on test data.

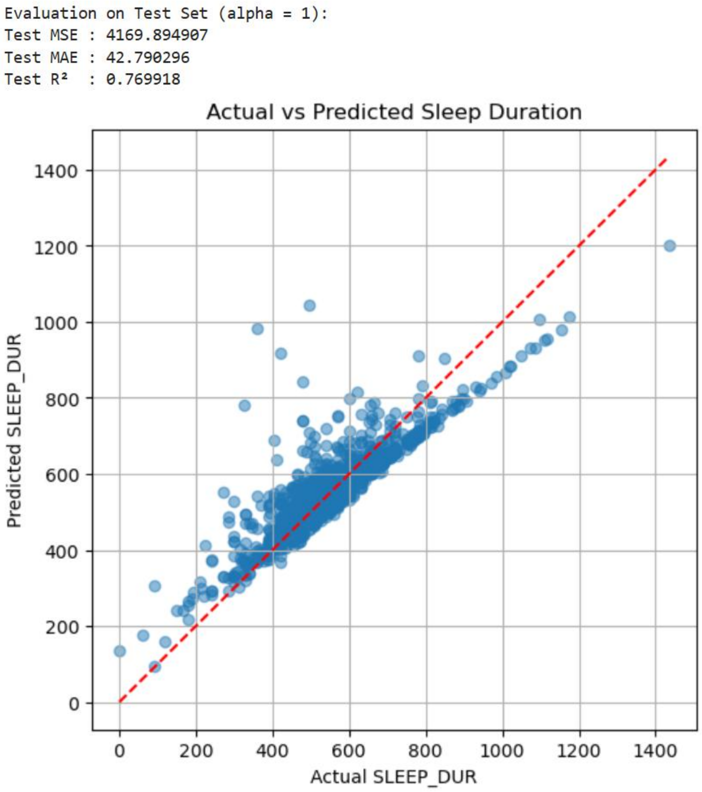
Table 3: Comparison of Models M3_1 to M3_9

Model	New Feature Added	MSE	MAE	R^2	Improvement In R^2 (%)	Correlation
Model M3_1	**Baseline only**	17000	94	0.05	−	0.06
Model M3_2	**M3_1 + OTHER_DUR**	14230	86	0.16	+220	0.32
Model M3_3	**M3_2 + ALONE_LEI_DUR**	13040	80	0.26	+63	0.42
Model M3_4	**M3_3 + GROUP_WORK_DUR**	10540	70	0.43	+65	0.58
Model M3_5	**M3_4 + GROUP_LEI_DUR**	8600	62	0.54	+26	0.66
Model M3_6	**M3_5 + ALONE_WORK_DUR**	7200	55	0.65	+20	0.73
Model M3_7	**M3_6 + OTHER_TRAVEL_DUR**	6400	51	0.70	+8	0.76
Model M3_8	**M3_7 + ALONE_INT_DUR**	4700	44	0.76	+9	0.84
Model M3_9	**M3_8 + GROUP_EXE_DUR**	3980	41	0.78	+3	0.88

Our final model is model M3_9 which explains around 78% of the variance in $SLEEP\_DUR$, with low error and a strong, safe correlation. Based on this model, the following time-related features were excluded to prevent leakage and overfitting:

$GROUP\_INT\_DUR, ALONE\_EXE\_DUR, TRAVEL\_WORK\_DUR, TRAVEL\_SOCO\_DUR, TRAVEL\_ENT\_DUR, TRAVEL\_EXE\_DUR$

The performance of this model on test data is shown in **Figure 6**. The change in the value of correlation, and the improvement of the models form model M3_1 to M3_9 can be seen through **Figure 7** and **Figure 8**, respectively. Finally, **Figure 9** shows how our model prediction on test data improved from model M3_1 to M3_9, and become closer and closer to the prefect prediction.



```
Evaluation on Test Set (alpha = 1):
Test MSE : 4169.894907
Test MAE : 42.790296
Test R²  : 0.769918
```

Figure 6: Model 3_9 result on test data

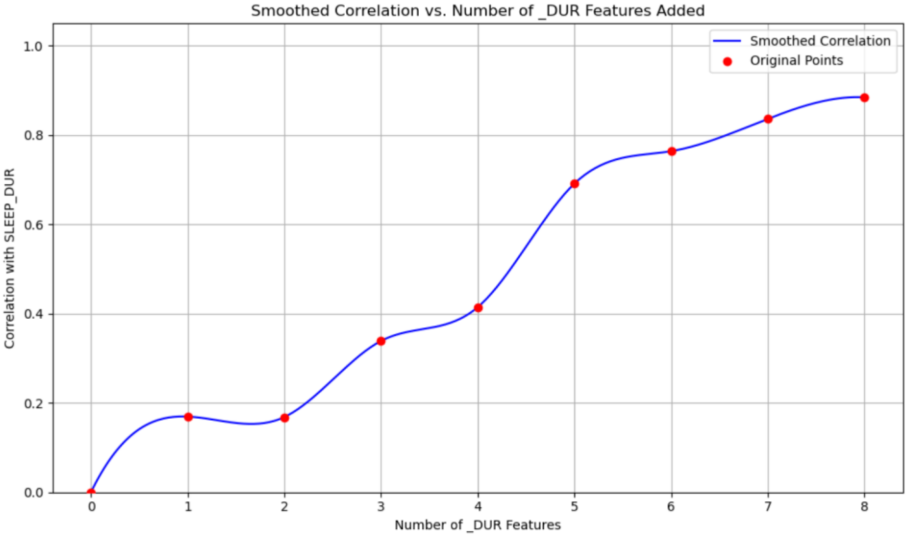


Figure 7: Change in correlation value form model M3_1 to M3_9

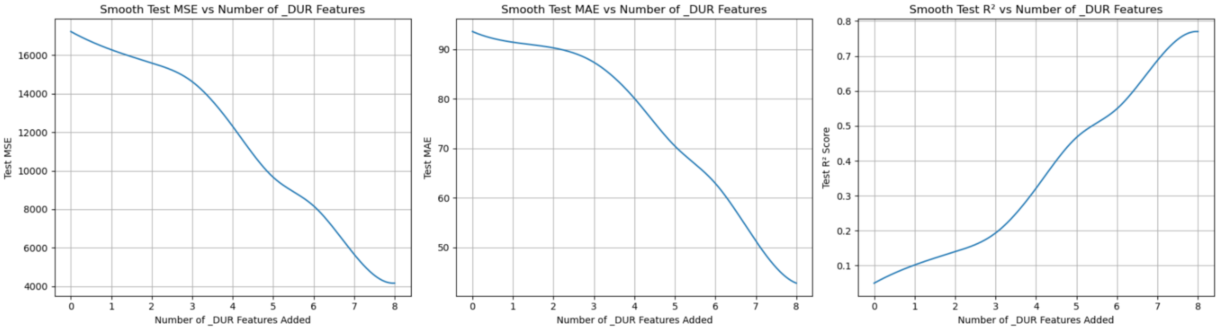
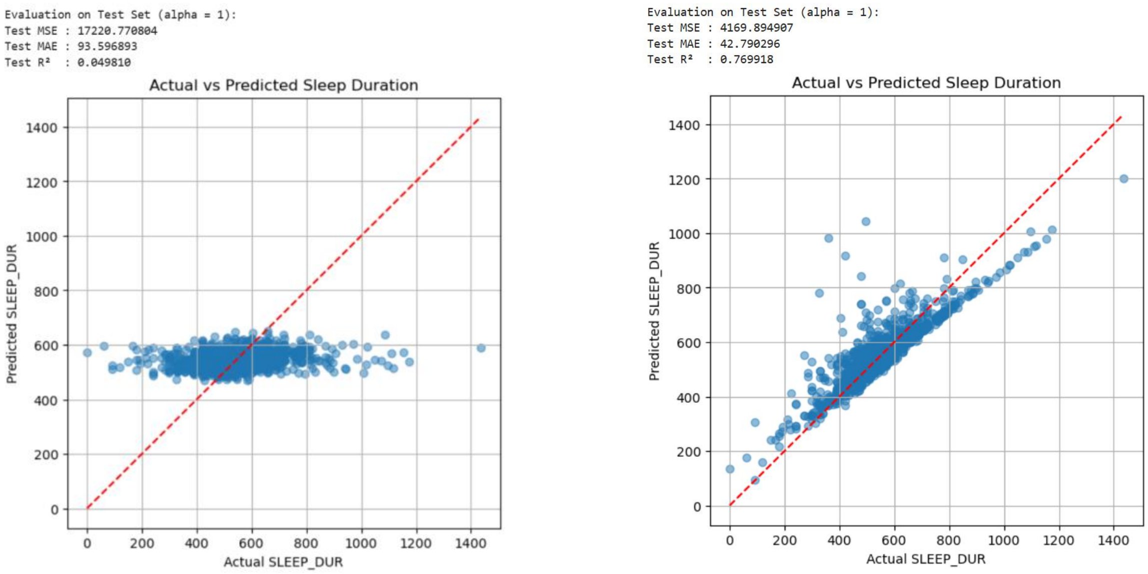


Figure 8: Improvement in metric from Model M3_1 to M3_9



```
Evaluation on Test Set (alpha = 1):
Test MSE : 17220.770804
Test MAE : 93.596893
Test R²  : 0.049810
```

```
Evaluation on Test Set (alpha = 1):
Test MSE : 4169.894907
Test MAE : 42.790296
Test R²  : 0.769918
```

Figure 9: Improvement in prediction on test data from model M3_1 to M3_9

4.2.1.1 Combinations of Every Two Time-Relevant Features

In this section, we aim to justify and evaluate our One-Factor-At-a-Time (OFAT) approach, where the order of feature inclusion is guided by the significance levels obtained from the Lasso analysis of the base model (model M3). Starting from this base model, which excludes all time-related features, we systematically tested every possible combination of two time-related features (resulting in model M3_1 variants). For each combination, we observed the change in R^2 to assess how much predictive power the additional features contributed. Finally, we compared the best-performing combinations against the hierarchy established by our OFAT selection strategy to validate its effectiveness.

Examining the results, we observe a small difference between our OFAT method and the exhaustive combination approach. The best-performing combination involves adding $OTHER\_DUR$ and $GROUP\_WORK\_DUR$, achieving an R^2 of 0.148784, which corresponds to an improvement of 271.96%. In contrast, our method selected the second-best combination with an R^2 of 0.139975 (a 249.94% improvement). These values are very close, indicating that our method came quite close to the optimal result. While both approaches are valid for adding additional features, the exhaustive method is significantly more computationally intensive. Given the marginal difference in performance, the OFAT method appears to be a more efficient and reasonable choice for feature selection in this context (model M3_4 onward).

4.3. Nonlinear Model

To evaluate the potential of nonlinear relationships in predicting daily sleep duration, we developed a Neural Network(NN) model. While linear models are interpretable and efficient, they often struggle to capture complex interactions between the features - especially between continuous and categorical variables.

4.3.1 Neural Networks

A **neural network** is a computational model composed of layers of interconnected nodes (neurons). Each layer transforms its input data using a learned function and passes the result to the next layer. The most common type of network for many tasks is the **feedforward neural network**, where information moves only in one direction—from input to output.

A simple feedforward network with one hidden layers may look like:

$$x \rightarrow Linear \rightarrow ReLU \rightarrow Linear \rightarrow ReLU \rightarrow Linear \ \rightarrow \ Output$$

4.3.2 Neural Network Architecture

Let’s introduce the neural network architecture we are going to use. The architecture consists of one hidden layer, with ReLU non-linearity.

1. Input Layer, $x \ \varepsilon \ R^n$
2. First Layer, Affine Transformation z_1 = W_1x + b_1
3. ReLu Activation, a_1 = $max\ (0, z_1)$
4. Second Linear Layer, z_2 = W_2a_1 + b_2
5. ReLu Activation, a_2 = $max\ (0, z_2)$
6. Output Layer , y = W_3a_2 + b_3

4.3.3 Neural Network Parameters

With the neural network architecture as mentioned above, we are going to train a neural network with L2 loss (mean squared error) and L1 regularization, where we're combining two key components:

1. L_2 Loss (MSE): Penalizes the squared difference between predictions and targets.
2. L_1 Regularization: Penalizes the absolute values of the model weights to promote sparsity.

Table 4: Neural network parameters

Parameter	Value	Description
Hidden Layers	[64,32]	Sizes of hidden Layers
Activation Function	ReLU	Activation used in hidden layers
Loss Function	L2 loss	Loss function used during training
Optimizer	Adam	Optimization Algorithm
Learning Rate	0.005	step size for weight updates
Batch Size	16	Number of samples per training batch
Epochs	200	Total passes through the training dataset
L1 Regularization	0.005	Prevents overfitting in hidden layers

4.3.4 Training Results and Discussion

First, we are going to train our model without any duration related features. Next, we add one feature at a time with and without interaction features. The results we obtained are tabulated as below.

Table 5: Neural Network Results

Model	Training Error	Validation Error	Test Error
Non-Duration Model	15715	16291	17535
Non-Duration Model(with interaction terms)	15468	16788	18283
Non-Duration Model + OTHER_DUR	13801	14727	17029
Non-Duration Model + OTHER_DUR (with interaction terms)	15615	16432	18027
Non-Duration Model + OTHER_DUR+ ALONE_LEI_DUR	12619	13227	15947
Non-Duration Model +...+ GROUP_EXE_DUR	1349	4438	5538

4.3.5 Discussion and Future Works

As of now, neural networks are not performing significantly better than linear regression. Possible reasons could be:

1. Exploding or vanishing gradients: Our model shows very high training error in each case, meaning it has not been trained properly. This could be because it is suffering from exploding or vanishing gradient problems because of which backpropagation is not happening properly.
2. Hyperparameter Tuning: Model needs to be experimented with batch size, learning rate and optimizer tuning.
3. Not enough non-linearity in data: The data we are using may not have enough non-linearity for the model to learn from.

In order to improve model performance, we want to do the following in future.

1. Generate X outputs for each target and train by cross entropy loss.
2. Try out combinations of input features.

5. Conclusion

In this project, we set out to predict how much people sleep each day using data from the 2023 American Time Use Survey (ATUS). Our goal was to understand how daily activities and personal characteristics relate to sleep time something essential but often overlooked. At first, when we included all activity durations in the model, we got nearly perfect results (R^2 around 1), which seemed too good to be true. And it was: the model was unintentionally "cheating" by using the fact that everyone has only 1,440 minutes in a day. This is known as target leakage.

To fix this, we carefully redesigned our approach using linear regression with Lasso regularization, which helps reduce overfitting by shrinking less important features. We started with a basic model that only used demographic info, then slowly added selected activity features based on how important they were, according to the Lasso results. After testing different alpha values (which control the regularization), we found that $\alpha = 1$ gave the best balance strong enough to simplify the model but still accurate.

Our final model (model M3_9) used eight activity related features like time spent on other activities, alone leisure, and working with others and achieved a solid performance: $R^2 = 0.78$, $MAE = 41$ minutes, and correlation = 0.88 on the test data. These numbers show the model predicts well without just memorizing the data. We also tested combinations of irrelevant features to confirm our model was robust and not just lucky.

In the end, this project showed us that with smart feature selection and a careful modeling process, it's possible to build a strong and generalizable predictor of sleep duration. It also taught us an important lesson: high accuracy isn’t always a good sign sometimes, it just means the model found a shortcut. By taking things step by step and thinking critically about the data, we built a model that not only works well but actually makes sense.