Student(s): Long Le 2203558

Title of the work: Practical Work 2 – Mobile Device Usage and User Behaviour Dataset

Prediction of Daily Screen On Time Using Neural Networks

Used algorithm(s):Artificial Neural Network (ANN)

Work description and analysis:

Description of the work:

- The dataset, user_behavior_dataset.csv, includes user behaviour and device data.
- The goal is to predict daily Screen On Time (hours/day)
- The dataset contains features such as app usage time, screen-on time, number of apps installed, data usage, and demographic data. The goal is to develop a predictive model to estimate battery drain and identify the most influential factors contributing to battery consumption.

Data preparation for the training:

The dataset includes 700 rows and 11 columns, focusing on Battery Drain (mAh/day). It includes numerical data like App Usage Time, Screen On Time, App Installed Apps, Data Usage, Age, Device Model, Operating System, and Gender. Preprocessing steps include dropping irrelevant columns, normalizing numerical features, and encoding categorical variables.

- Dataset Features:
- Key features used include:
    - App Usage Time (min/day)
    - Battery Drain (mAh/day)
    - Number of Apps Installed
    - Data Usage (MB/day)
    - User Behaviour Class
- Target variable: Screen On Time (hours/day)
- Encoding and Scaling:
    - Categorical features (Gender, Device Model, Operating System) were one-hot encoded.
    - Input features for predictions were standardized using StandardScaler.
- Correlation Analysis:
    - The correlation matrix highlighted App Usage Time and Battery Drain as highly correlated with the target variable.

df - DataFrame

Index	User ID	Device Model	erating Syste	age Time (m	On Time (hou	y Drain (mAh	er of Apps In	Usage (MB/	Age	Gender	r Behavior Cl
0	1	Google Pixel 5	Android	393	6.4	1872	67	1122	40	Male	4
1	2	OnePlus 9	Android	268	4.7	1331	42	944	47	Female	3
2	3	Xiaomi Mi 11	Android	154	4	761	32	322	42	Male	2
3	4	Google Pixel 5	Android	239	4.8	1676	56	871	20	Male	3
4	5	iPhone 12	iOS	187	4.3	1367	58	988	31	Female	3
5	6	Google Pixel 5	Android	99	2	940	35	564	31	Male	2
6	7	Samsung Galaxy S21	Android	350	7.3	1802	66	1054	21	Female	4
7	8	OnePlus 9	Android	543	11.4	2956	82	1702	31	Male	5
8	9	Samsung Galaxy S21	Android	340	7.7	2138	75	1053	42	Female	4
9	10	iPhone 12	iOS	424	6.6	1957	75	1301	42	Male	4
10	11	Google Pixel 5	Android	53	1.4	435	17	162	34	Female	1
11	12	OnePlus 9	Android	215	5.5	1690	47	641	24	Male	3
12	13	OnePlus 9	Android	462	6.2	2303	65	1099	57	Female	4
13	14	Xiaomi Mi 11	Android	215	4.9	1662	43	857	43	Male	3
14	15	iPhone 12	iOS	189	5.4	1754	53	779	49	Female	3

Figure 1. First 14 rows with all columns of training data

Neural Network Training

Model Design and Training

- Experiment 1: Single Feature (App Usage Time)
  - ANN with a single dense layer (1 feature, 1 output unit) was trained for 100 epochs.
- Experiment 2: Multi-Feature Input
  - ANN with a single dense layer for all input features was trained for 400 epochs.
- Optimizer: Adam (Learning Rate = 0.001)
- Loss Function: Mean Squared Error (MSE)

Training Curves

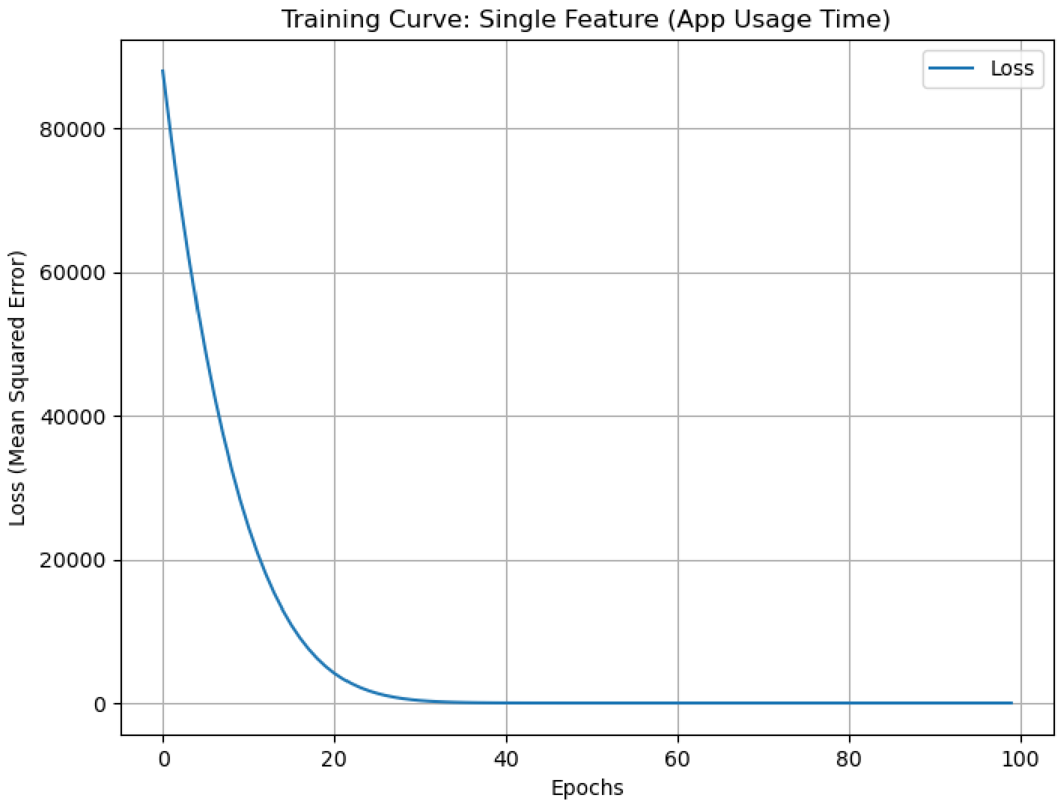

Figure 3. Training Curve of the first 100 Epoch

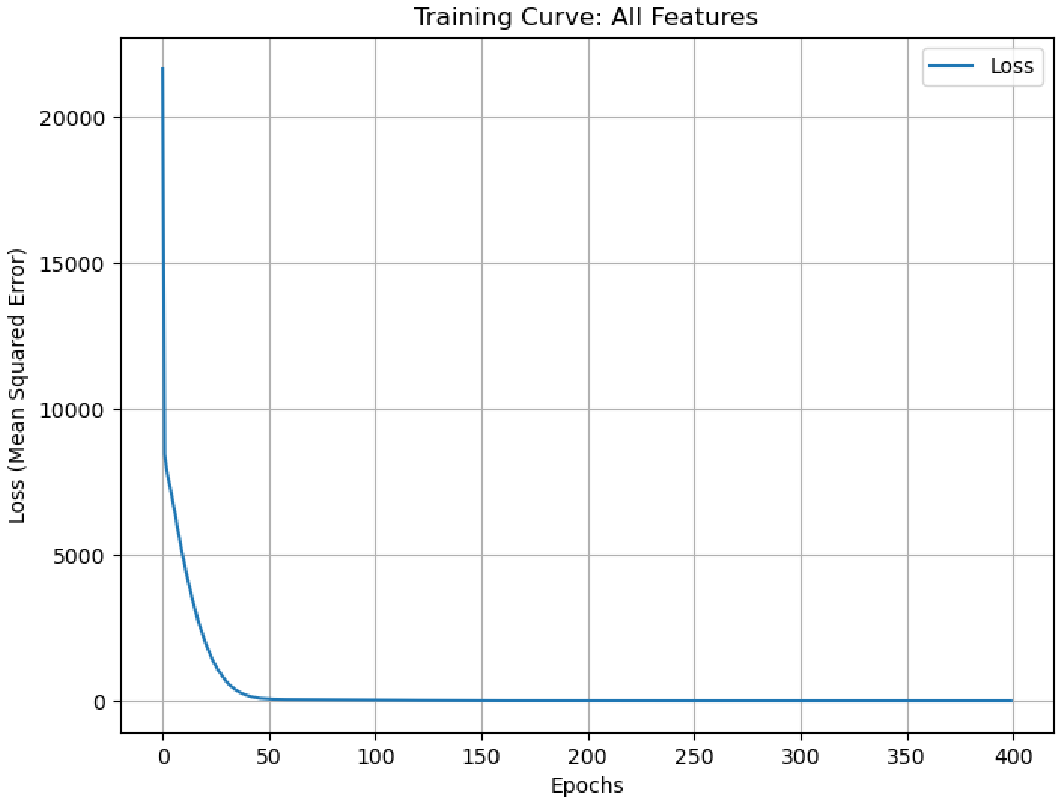

Figure 4. Training Curve of the 400 Epoch

Figure 5. Training Epoch

Relevant Metrics

Evaluation Metrics

- R^2 Score: 0.81
- Root Mean Squared Error (RMSE): 1.32

- Mean Absolute Error (MAE): 0.99

```
Epoch 400/400
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 577us/step - loss: 0.9024 - mae: 0.7040 - mse: 0.9024
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
r2 score:  0.8067618524281378
The rmse is:  1.317115442119806
The mae is:  0.9961376958574569
WARNING:tensorflow:5 out of the last 13 calls to <function
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x000001A01EEBDE40> triggered tf.function retracing. Tracing is expensive and the
excessive number of tracings could be due to (1) creating @tf.function repeatedly in a
loop, (2) passing tensors with different shapes, (3) passing Python objects instead of
tensors. For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For
(3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Predicted Screen On Time for new user: [0.41] hours/day
```

Figure 6. Result of MSE and MAE results over training epochs

Conclusions of the Results

- Validation and Analysis:
  The ANN achieved an R^2 score of 0.81, demonstrating strong predictive capability. MAE of 0.99 indicates that the average prediction error is less than 1 hour/day, which is acceptable for this application.

- Model Effectiveness:
  The multi-feature model outperformed the single-feature experiment in terms of accuracy, highlighting the importance of using all relevant features for predicting Screen On Time.

- Usability:
  The predictions can be useful in applications like mobile usage tracking or energy optimization for device manufacturers.

- Improvements:
  - Increasing data size or adding more diverse features could improve model generalization.
  - Implementing deeper neural networks or hyperparameter tuning might further enhance results.