# Project 7

## Real-Time Screen-Time Monitoring with Machine Learning

CptS 466

Kevin Evans

## System Design

In this project, a screen-time monitoring solution was created on the TM4C board. The board was connected to both a light and temperature sensor, which was read by the device. It was connected to a Bluetooth module, which transmitted the data to a nearby device. Initially, the data was collected, and Weka was used to create a model. The generated classifier was then pushed onto the device and was able to classify whether the device was nearby a screen that’s turned on.

### Experiment

The temperature and light sensors were connected to the ADC ports of the TM4C board. When the board was near a screen that was on, SW1 was pressed and recorded with the corresponding raw temperature and light sensor inputs. The data was collected over five minutes at a frequency of 1 data point per second. This data was collected over Bluetooth and Weka was used to analyze the data and create a model. The dataflow diagram is shown below in Figure 1.

Diagram

Description automatically generated

Figure 1: The dataflow diagram of the experiment.

Four scenarios were tested: using a computer (screen on), using a phone (screen on), cooking (screen off), and reading (screen off). It should be noted that the temperature varied across all tests, as both cooking and sitting nearby the temperature sensor raised the temperature noticeably. The ambient light during testing was also higher than most ambient lighting.

In Weka, two classifiers were tested: a J48 decision tree and a decision stump. The results were ten-fold cross-validated and an accuracy was recorded for each. Next, the model was then implemented on the device, so the screen on/off can be classified in real-time.

### Results

Both classifiers performed well. The decision tree has an accuracy of 97.3% and the decision stump has an accuracy of 94.0%. The results from Weka are recorded in the appendix. The performance metrics are shown below in Table 1.

Table 1: Performance metrics of each classifier.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | *f*-measure |
| J48 Decision Tree | 0.973 | 0.973 | 0.973 | 0.973 |
| Decision Stump | 0.940 | 0.944 | 0.940 | 0.940 |

The generated J48 decision tree is shown by Figure 2 below. With only four levels, it will be easy to implement with simple if-statements on the TM4C. Although temperature is on the top-most node and likely contributed the most to the classifier, this may be biased due to the lack of data points. This will be later discussed later.

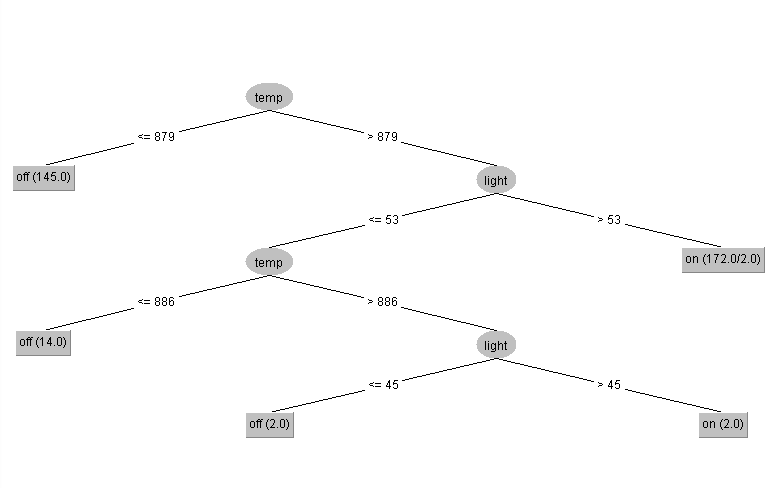


Figure 2: The J48 decision tree generated by Weka.

## Discussion

Both the temperature and light were used as features for the classifier, although the temperature should had no effect on the screen-on detection. Although the decision tree classifier has a high accuracy for the given data, it may not be sufficient for detecting screen times outside of the testing area. This is due to the ambient light and temperature conditions during testing significantly skewing the classifier. It would be wise to include different sensors instead, including a motion/IMU sensor, a microphone, etc.

When testing the device outside of the target area, i.e. testing in the classroom, the classifier performed poorly and could not detect the screen nearby. This is due to the ambient temperature in the classroom being colder than the area used in testing. It was concluded that the classifier did not capture realistic scenarios of the screen on/off times.

## Appendix

#### Listing 1: Weka Report of the Decision Tree

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: raw\_data

Instances: 335

Attributes: 3

temp

light

screen\_on

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

------------------

temp <= 879: off (145.0)

temp > 879

| light <= 53

| | temp <= 886: off (14.0)

| | temp > 886

| | | light <= 45: off (2.0)

| | | light > 45: on (2.0)

| light > 53: on (172.0/2.0)

Number of Leaves : 5

Size of the tree : 9

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 326 97.3134 %

Incorrectly Classified Instances 9 2.6866 %

Kappa statistic 0.9462

Mean absolute error 0.033

Root mean squared error 0.1547

Relative absolute error 6.6047 %

Root relative squared error 30.945 %

Total Number of Instances 335

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.975 0.029 0.970 0.975 0.972 0.946 0.981 0.988 off

0.971 0.025 0.977 0.971 0.974 0.946 0.981 0.965 on

Weighted Avg. 0.973 0.027 0.973 0.973 0.973 0.946 0.981 0.976

=== Confusion Matrix ===

a b <-- classified as

159 4 | a = off

5 167 | b = on

#### Listing 2: Weka Report of the Decision Stump

=== Run information ===

Scheme: weka.classifiers.trees.DecisionStump

Relation: raw\_data

Instances: 335

Attributes: 3

temp

light

screen\_on

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Decision Stump

Classifications

temp <= 880.0 : off

temp > 880.0 : on

temp is missing : on

Class distributions

temp <= 880.0

off on

1.0 0.0

temp > 880.0

off on

0.09473684210526316 0.9052631578947369

temp is missing

off on

0.48656716417910445 0.5134328358208955

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 315 94.0299 %

Incorrectly Classified Instances 20 5.9701 %

Kappa statistic 0.8802

Mean absolute error 0.102

Root mean squared error 0.234

Relative absolute error 20.4226 %

Root relative squared error 46.8138 %

Total Number of Instances 335

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.890 0.012 0.986 0.890 0.935 0.884 0.933 0.951 off

0.988 0.110 0.904 0.988 0.944 0.884 0.933 0.887 on

Weighted Avg. 0.940 0.062 0.944 0.940 0.940 0.884 0.933 0.918

=== Confusion Matrix ===

a b <-- classified as

145 18 | a = off

2 170 | b = on