FIT5149 S1 2021 Assessment 2: Residential Energy Appliance Classification

April 2021

Marks	35% of all marks for the unit
Due Date	~~17:00 Friday 28 May~~ **23:55 Sunday 30 May** 2021
Extension	An extension could be granted for circumstances. Please refer to the university webpage on special consideration. A special consideration application form must be submitted. Please note that ALL special consideration, including within the semester, is now to be submitted centrally. All students MUST submit an online special consideration form via Monash Connect.
Lateness	For all assessment items handed in after the official due date. Without an agreed extension, a 10% penalty applies to the student's mark for each day after the due date (including weekends and public holidays) for up to 5 days. Assessment items handed in after 5 days will not be considered/marked.
Group Member	This assignment is a **group assignment**, and you are highly encouraged to form **a group with three members, including yourself**. A group with more than three members (say >= 4) is not allowed. You may complete this assessment by yourself or form a group with two members, if you have difficulty forming a group.
Authorship	The final submission must be **identifiable your group's own work in the group**. Breaches of this requirement will result in an assignment not being accepted for assessment and may result in disciplinary actions.
Submission	Each group is required to submit two files, one PDF file contains the report, and another is a ZIP file containing the implementation and the other required files. The two files must be submitted via Moodle. All the group members are required to log in to Moodle to accept the terms and conditions on the Moodle submission page. A draft submission won't be marked.
Programming language	Either R or Python

Note: Please read the description from the start to the end carefully before you start your work! Given that it is a group assessment, **each group should evenly distribute the work among all the group members**.

1 Introduction

Energy production/consumption is the largest source of greenhouse gas emissions. Energy efficiency plays a crucial role in the transformation of future energy systems and cutting the rapid growth of global energy demand to able early decommissioning of fossil-fuel power plants and combat climate change. Electricity consumption in residential sectors accounts for more than 20% of total consumption, and thus energy-saving technology for residential buildings is of vital importance. Choosing the right time to consume the right amount of electricity will increase energy efficiency and reduce emissions.

Load monitoring (also known as load detection and load disaggregation) is a promising technique to provide detailed electricity consumption information and usage of individual appliances in residential buildings. An illustrative example of load monitoring[1] for appliances, such as refrigerator, oven, and stove, is shown in Figure 1. Take the oven as an example. When the oven is turned on, it is used for a period of time until being turned off. A more recent review of load monitoring can be found in the paper entitled "Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools-A review"[2].

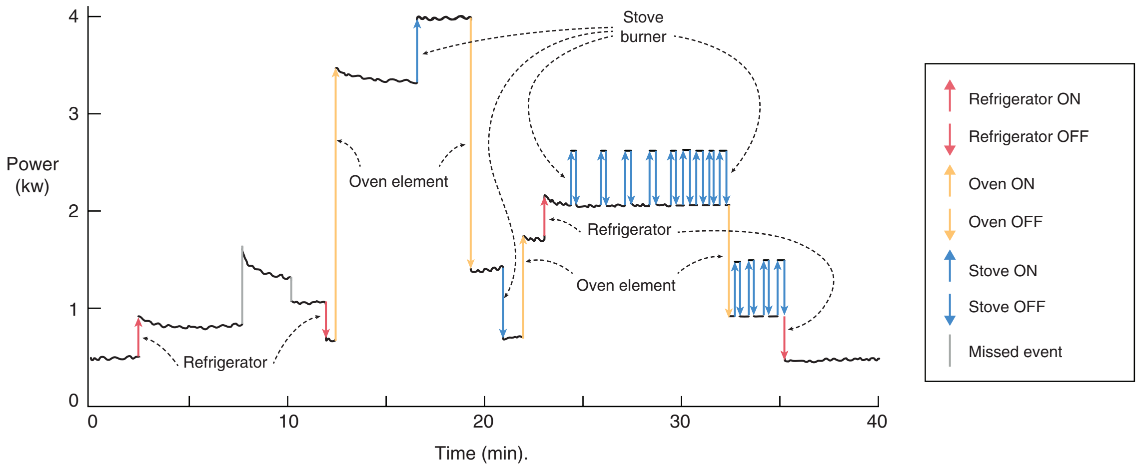

Figure 1: Example of load monitoring (Hart 1992, Copyright IEEE).

In this assessment, you are given residential aggregate load data (i.e., smart meter reading) with appliance usage labels (i.e., whether the target appliances are being used or not). The target appliances include air conditioner, electric vehicle charger, oven, cloth washer, and dryer. This is a typical classification problem, and there are many machine learning methods (such as logistic regression, SVM, and tree-based methods) that can be used in this classification task. The inputs are labels and extracted features, and the response variables are target appliance status (i.e., being used or not). Figure 2 shows a typical framework used in the supervised classification.[3] As shown in the figure, there are three major steps: generating features, developing a proper classifier, and applying the classifier to the unseen data. The feature extractor is shared by both training and prediction, which tells us that data used in training and prediction should share the same feature space.

The aim of this challenge is to develop multiple classifiers for each individual target appliance that can detect whether target appliances are being used in each time interval (e.g., 1-minute interval) as correctly as possible. In other words, you will develop five classifiers for five appliances (and each classifier solves one corresponding appliance classification problem.). The evaluation emphasises on the performance and interpretability of the classifiers. Specifically, you are expected to

[1]G.W. Hart, 1992. Nonintrusive appliance load monitoring. Proceedings of the IEEE, 80(12), pp. 1870-1891, 1992.

[2]The link to the paper is https://onlinelibrary.wiley.com/doi/epdf/10.1002/widm.1265

[3]The figure is download from https://www.nltk.org/book/ch06.html

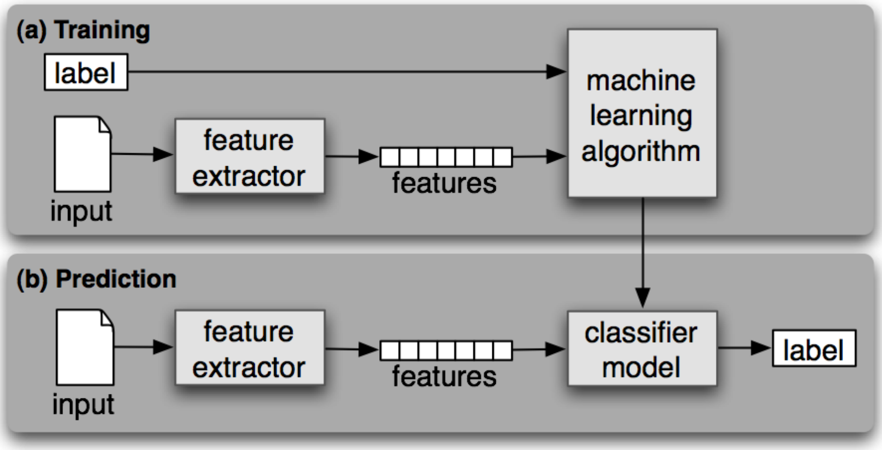

Figure 2: A general framework for the supervised classification.

- **develop five classifiers** to detect five appliances being used as accurate as possible; Note that the five target appliances can be classified by different classifiers/models using different features. You can treat five appliances in five independent classification tasks.
- and **explain the outcomes** of your classifiers. For example, what features are strongly associated with the response variables and effective in the classification task.

2 Dataset

Data Source	Type	# Appliances	# training examples	# test examples
Dataport	Energy load	5	417,720	105,540

Table 1: One-minute interval residential load data sets with five appliance labels.

We provide the following data sets (Table 1).

- train_data_withlabels.csv contains 417,720 instances, each of which has 15 columns, including load, five appliance labels (1 indicating using appliances and 0 indicating not using appliances), two time stamps (such as hour of the day and day of the week), and 7 attributes/features. The five appliances are air conditioner (ac), electric vehicle charger (ev), oven, cloth washer (wash), and dryer. Note that all the load data are minute-by-minute meter readings in a sequential manner.
- test_data_nolabels.csv: It contains 105,540 instances, each of which has 10 columns, including load, two timestamps, and 7 attributes but no appliance labels.

Note: The load, timestamps, and appliance labels are the original data, which should be used in the classification task. Other attributes/features are all extracted from the load data using various measures[4]. These sample attributes are optional for potentially improving the performance. You are highly encouraged to process the sequential load data and extract more/different features and evaluate their effectiveness in the classification task. Any information about the test data cannot be used in training the classifiers.

[4] A brief description and some hints will be provided in Section 3.

3 Feature Extraction

3.1 Introduction to Features Provided

In the provided data sets, a few features have been provided. The following is a brief introduction to the features in train_data_withlabels.csv and test_data_nolabels.csv.

- *dif* – Difference between two sequential load data points.
- *absdif* – Absolute value of dif.
- *var* – Variance of load over a neighborhood time window of 30 minutes around each load data point.
- *entropy* – The Shannon entropy[5] that measures the "forecastability" of a time series data.
- *nonlinear* – The nonlinearity coefficient[6] is used in Terasvirta's nonlinearity test.
- *hurst* – The hurst[7] is used as a measure of the long-term memory of a time series.

3.2 Hints on Feature Extraction (Optional)

Selecting relevant features and deciding how to encode them for a classification algorithm is crucial for learning a good model. The above features serve as a starting point, and you are **highly encouraged** to consider different and/or more effective features.

- Since appliance usage may last for a while (e.g., a few or several minutes), you may consider a neighborhood time window around each load data point. You can treat the window size as a hyper-parameter to tune and find the optimal size to extract features.
- There are many useful online tutorials on sequential data in either R or Python, for example,
  - Introduction to the tsfeatures R package for sequential/time-series characteristics and installation tutorial [8]
  - Introduction to the tsfresh python package for sequential/time-series characteristics and and installation tutorial[9]
  - Feature extraction in Scikit-learn[10]

4 Task Description

In this assessment, you will focus on the following two tasks:

- **Prediction Task** – You are expected to develop classifier(s) that can give you the most accurate prediction in the appliance classification task. The algorithms that you can use are not limited to the algorithms covered in the lectures/tutorials. The goal is to find the most accurate classifier.
- **Inference Task** – The purpose of the inference task is to identify the key features that have strong effects on the prediction results, indicating strong drivers of the appliance usage.

[5] https://cran.r-project.org/web/packages/tsfeatures/vignettes/tsfeatures.html#entropy
[6] https://cran.r-project.org/web/packages/tsfeatures/vignettes/tsfeatures.html#nonlinearity
[7] https://cran.r-project.org/web/packages/tsfeatures/vignettes/tsfeatures.html#hurst
[8] https://cran.r-project.org/web/packages/tsfeatures/vignettes/tsfeatures.html
[9] https://tsfresh.readthedocs.io/en/latest/
[10] https://scikit-learn.org/stable/modules/feature_extraction.html

4.1 Prediction Task

In order to find the most accurate classifiers, each group should empirically compare at least 2 different types of classification methods for each appliance classification task, and then submit the one performing the best. Please note an algorithm with different input features will only count as one type of classifier. For example, logistic regression will be count as one type of classifier, no matter what features you use.

Note that the performance of the classification task not only depends on the classifier but also on the features. As introduced in Section 3, a set of features have been provided to you as a starting point. You are also highly encouraged to try different/more effective features.

4.2 Inference Task

Inference can be based on variable correlation analysis, regression equations, or any other form. To finish this task, you are expected to use proper data analysis techniques to identify a subset of attributes that have a significant impact on the appliance usage. **Report** your identified features with statistical evidence and interpret the attribute subsets.

4.3 Some Hints

There are some hints that we summarise based on the past submissions made by previous cohorts.

- Avoid using the absolute path in your Jupyter notebook. Instead, use a relative path (.e.g, "./train_data_withlabels.csv") and place the data file in the same folder where your Jupyter notebook is.
- Avoid just plainly showing the results without meaningful interpretation/discussion (this has been stressed multiple times in both tutorials and lectures). For example, if you use any plot, you will need to clearly discuss the information delivered by the plots in the context of the task.
- Choose the appropriate plots or statistics to show the right information.
- While developing the model, make clear, for example, how the optimal parameters are chosen if there is any, etc.
- Be precise in the use of various tools and the corresponding discussion.
- Avoid submitting an extremely long Jupyter notebook, which could result in a lot of redundant information, easily losing the focus of your work.
- Make sure the logic (or the methodology) you used to develop the model is properly documented.
- Write your discussion using the markdown cells and avoid putting it in the code cell as we will use this to gauge your reasoning skills.
- Make use of the discussion forum and consultations to clear any doubts that you may have regarding the tasks you want to accomplish.
- Before making the final submission, you must make sure that your Jupyter Notebook runs without any errors. A simple step is to click "Kernel → Restart & Run All".

5 Evaluation

The evaluation metric used in test is the F1 score, which is defined as

$$\mathrm{F1} \quad = \quad 2 \times \frac{\textbf{Precision} \times \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}}$$

where

$$\text{Precision} = \frac{\textbf{Number of True Positive}}{\textbf{Number of True Positive and False Positive}}$$

and

$$\text{Recall} = \frac{\textbf{Number of True Positive}}{\textbf{Number of True Positive and False Negative}}.$$

You can use the existing python/R code to compute F1, precision, and recall, for example

- F1 score in Python[11], Precision in Python[12], and Recall in Python[13],
- F1 score in R[14], Precision in R[15], and Recall in R[16].

6 Submission

To finish this data analysis challenge, all the groups are required to submit the following files:

- **"pred_labels.csv"**, where the label prediction on the testing documents is stored.
  - In your "pred_labels.csv", there will be six columns: the first one is the column index from 1 to the number of total load data points 105540. From the second column to the sixth column, they should include labels (0 or 1) for five appliances in order: air conditioner, electric vehicle charger, oven, cloth washer, and dryer. Remember the first row of your "pred_labels.csv" file should be "col_index", "ac", "ev", "oven", "wash", and "dryer.
  - The "pred_labels.csv" must be reproducible by the assessor with your submitted R/Python code.
- The **R/Python implementation** of your **final** classifier with A README file that tells the assessor how to set up and run your code. The output of your implementation must include the label prediction for all the appliances (see the above description about **"pred_labels.csv"**). The use of Jupyter notebook or R Markdown is **not required**. All the files that are required for running your implementation must be compressed into a **zip** file, named as "**groupName_ass2_impl.zip**". Please note that the unnecessary code must be excluded in your final submission. For example, if you tried three different types of models, say logistic regression, SVM, and classification tree, and your group decides to submit classification trees as the final model. You should remove the code for the other models from the submission. **The discussion of the comparison should be included in your report.** *However, you should keep a copy of the implementation used for comparison for the purpose of the interview.*
- **A PDF report**, where you should document in detail the development of the submitted classifier. **The maximum number of pages allowed is 8**. The report must be in the PDF format, named a **"groupdName_ass2_report.pdf"**. The report must include (but not limited to)
  - The discussion of how the data preprocessing/features selection has been done.

[11] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html?highlight=f1#sklearn.metrics.f1_score
[12] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html#sklearn.metrics.precision_score
[13] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html#sklearn.metrics.recall_score
[14] https://www.rdocumentation.org/packages/measures/versions/0.3/topics/F1
[15] https://www.rdocumentation.org/packages/Metrics/versions/0.1.4/topics/precision
[16] https://www.rdocumentation.org/packages/Metrics/versions/0.1.4/topics/recall

- The development of the submitted classifier: To choose an optimal classifier for a task, we often carry out empirical comparisons of multiple candidate models with different feature sets. In your report, you should include a comprehensive analysis of how the comparisons are done. For example, the report can include (but not limited to)
    * A description of the classifier(s) considered in your comparison.
    * The detailed experimental settings, which can include, for example, the discussion of how the cross-validation is set up, how the parameters for the model considered (if applicable) are chosen, or the setting of semi-supervised learning (if applicable).
    * Classification accuracy with comprehensive discussion.
    * The justification of the final model submitted.

  Warning: If a report exceeds the page limit, the assessment will only be based on the first 8 pages.

- A signed group assignment cover sheet, which will also be included in your zip file.
  Warning: typing name is not counted as a signature in the cover sheet.

7 How to submit the files?

The Moodle setup allows you to upload only two files

- "**groupdName_ass2_report.pdf**": A pdf report file, which will be submitted to Turnitin.
- "**groupName_ass2_impl.zip**"': a **zip** file includes
  - the implementation of the final submitted model
  - "predict_label.csv", where the label prediction on the testing documents is stored.
  - the signed grouped assignment cover sheet

While submitting your assignment, you can ignore the Turnitin warning message generated for the ZIP file.

Please note that

- **Only one group member needs to upload the two files. But all the group members have to login in to their own Moodle page and click the submit button in order to make the final submission.** If any member does not click the submit button, the uploaded files will remain as a draft submission. A draft submission won't be marked!
- **The two files must be uploaded separately.**

8 Academic integrity

Please be aware of University's policy on academic integrity. Monash University takes academic misconduct[17] very seriously. You can learn from the above materials and understand the principle of how the analysis was done. However, you must finish this assessment with your own work.

[17] https://www.monash.edu/students/study-support/academic-integrity