COSC2673 | Machine Learning

Assignment 2 | Decision Trees | Multi-Layer Perceptrons

Date Due: Wednesday October 10th, 11.59pm (Week 12)

---

1 Introduction

Critical analysis is a very important aspect of using machine learning algorithms. That is, it is insufficient to simply execute the machine learning algorithms. Instead, it is important to have a good understanding of many concepts, including:

- The nature of the learning problem and the available training data
- What class(es) of problems each algorithm is suited towards
- How the features of the learning problem and data set are used by each algorithm
- Why each algorithm may succeed or fail at learning a sufficient model

This assignment will require you and a group partner to analyse and compare the performance of two machine learning algorithms on two separate data sets for supervised learning problems. You will produce a written report of your findings. The *emphasis* of the marking criteria is on your analysis of the data sets, learning problems, and machine leaning algorithms.

1.1 Learning Outcomes

This assignment supports the following Course Learning Outcomes:

- Understand the fundamental concepts and algorithms of machine learning and applications
- Understand a range of machine learning methods and the kinds of problem to which they are suited
- Set up a machine learning configuration, including processing data and performing feature engineering, for a range of applications
- Apply machine learning software and toolkits for diverse applications

1.2 Group Work

This assignment will be completed in pairs (groups of 2). The course form may be of assistance in finding a group partner. You will need to register your groups on Canvas, under the Assignment 2 section. Register your group by no later than ***Thursday September 13th (Week 8)***.

If you are unable to find a partner, please discuss this with Tim ***as soon as possible and before*** the group registration deadline.

If at any point you have problems working with your partner, please inform Tim, Jeff, Xiadong, or Arian as soon as possible, so that any issues may be resolved.

1.3 Assessment Tasks

In this assignment you will complete three tasks:

1. (10 marks) Supervised learning problem with decision trees for predicting sale prices of for the Melbourne property market (houses, units, etc.), in Section 2.
2. (10 marks) Supervised learning problem with a multi-layer perceptron for image classification with the MNIST fashion dataset, in Section 3
3. (5 marks) Comparison and analysis of the two supervised learning problems, in Section 4

Your submission will consist of three components, detailed in Section 5:

- A written report (maximum 12 pages).
- Two code files, with one code file for each supervised learning task.

Your report and code files are due ***Wednesday October 10th, 11.59pm (Week 12)***.

The marking criteria is detailed in Section 6. Unless there are exceptional circumstances, both members of the group will receive the same mark.

1.4 Relevant Lecture/Lab Material

This assignment covers lecture and lab material for Weeks 5 to 10 (inclusive). You may find that you will be unable to complete some of the activities until you have completed the relevant lab work. Specifically, the Melbourne house prices problem (Section 2) requires material from Week 5 and 9, and the MINST image classification problem (Section 3) requires material from Week 10.

2 Decision Tree Learning

In this task you will investigate the prices of properties sold within the Melbourne region from 2016 - 2018. The sale prices are represented by price bands (ranges), for example $200K-$400K ($200,000 - $400,000) or $1M-$2M ($1,000,000-$2,000,000). Using Python and Scikit Learn (`sklearn`), you are required to train *at least two decision tree classifiers*, to predict the sale price band of future property sales.

The challenge of this task is not in training a decision tree classifier. Instead the challenge lies in:

- Working on a real-world data set
- Pre-processing the data set into a suitable format for use with a decision tree classifier
- Selecting appropriate features for the decision tree to use
- Constructing additional features for the decision tree to use
- Handling missing attributes
- Selecting appropriate examples from the data set for training and testing

You are required to analyse the classifiers that you train in your report. Your report must also contain an ultimate conclusion of the classifier (if any) that you would use to predict the sale price bands of upcoming properties in Melbourne for the remainder of 2018.

2.1 Data set

The Melbourne property price data set is provided in the `property_prices.csv` file on Canvas. The file contains approximately 35,000 records, which has been collected from real-world data of the Melbourne property market from 2016-2018[1]. There is only one data file contained training data. There is no separate test data file.

The data set contains:

- Target class: `price_bands`
- 20 Attributes. A full description of the attributes is provided in the `property_prices_names.txt` file available on Canvas.

Before training any classifiers, spend time to familiarise yourself with the data set. ***This is a real-world data set***. It is highly noisy, contains inconsistencies and many examples have missing attributes. This data set has not been cleaned or nicely prepared for you. Therefore you will need to carefully examine the data set.

Like real-world problems, there is no test data file for you to use for a final evaluation of your classifier. In this task you will need use the available data to make evaluations about data that will appear in the future.

2.2 Task

You are required to train at least two *decision tree classifiers*, and evaluate the performance of the classifiers. The purpose of the classifiers are to *predict the price band of properties that have been sold*. The classifier can then be used to predict the sale price band of future property sales. In your report you are required to describe, in no more than 5 pages, how you trained the classifiers, justify any decisions you made, and provide an analysis of the classifiers. Questions you may wish to consider in your analysis are suggested below. In your report you must also make an ultimate conclusion. Choose a classifier, that in your judgement, is the best suited for predicting the sale price of Melbourne properties in the coming months, with a justification of your choice. You will also need to submit the Juypter notebook/python code that you used.

To complete this task, it is recommended that you adopt the following workflow, of three key steps:

1. Pre-process the data set
2. Train a classifier
3. Evaluate the classifier

There are 10 marks for this task, broken down into 5 marks for the method, and 5 marks for the analysis. More information is given in Section 6.

[1]The data set was originally collected by Tony Pino and provided through Kaggle. It has been modified for this assignment

2.2.1 Pre-processing the data set

As previously noted, this is a real-world data set, and must be pre-processed before a decision tree classifier can be learnt. You may wish to consider:

1. There may be many examples (rows) that are not relevant to the learning problem. Including these examples in the training data may prevent you learning an accurate model and should be removed.
2. Many examples have attributes with missing values. `sklearn`'s algorithms do not automatically handle missing values. You will need to make a judgement of how to handle examples with missing attributes, either by removing them, or generating appropriate values.
3. Feature selection, that is, limiting the classifier to use a subset of the attributes. There may be attributes that are not relevant to the learning problem or provide duplicate information. Also if there are too many attributes, it may not be possible for the classifier to find a reasonable model.
4. Feature construction, that is, generating new features. Some attributes may not be in a suitable format. Thus you may use the existing attributes to generate new features for use by the classifier.

2.2.2 Training the classifier

Once the data has been pre-processed, a classifier can be trained. You may wish to consider:

1. The number of examples from the pre-processed data set that are used for training. That is, constructing training and testing data sets.
2. Choosing suitable configuration parameters for the decision tree classifier, such as the maximum tree depth, or the minimum number of examples in a node.
3. Conducting tuning over the parameters of the classifier.

2.2.3 Evaluating the classifier

Once you have trained a classifier, you will need to evaluate its performance. You may wish to consider:

1. Constructing a separate training and test data set.
2. Examining the confusion matrix and classification report.
3. Using k-fold cross validation.

2.2.4 Additional Hints

Some additional hints for getting up and running:

- This is a very open-ended task. You are asked to provide your judgment. There is no correct answer or "best" model.
- The data file is provided in a CSV (comma separated) format. This can be easily loaded into spreadsheet programs such as Excel, and exported back as a CSV. You may find it easier to use these programs for feature construction, compared to `sklearn`.
- CSV formatted files can be loaded by Pandas using:
  ```
  pd.read_csv("file.csv")
  ```
- The data set contains a variety of different data types. Take care in how you choose to handle this.
- To find a sufficient model, you will probably need to train more that two decision trees. However, you don't have time to investigate every possible combination of features, and classifier parameters. Be smart in your choices, but also be careful about making broad generalisations about what "will work".

2.3 Extensions

You are welcome to explore algorithms beyond a simple decision tree classifier. (Reports that receive the top marks are expected to do so.) However to provide reasonable constraints, in making your ultimate conclusion, you are required to stick to methods which utilise decision trees. These algorithms include:

- Ensemble methods:
  - AdaBoost
  - Bagging
  - Classifier Voting
- Random Forest decision trees

You are welcome to explore different forms of the classification task. For example, rather than training a single classifier that classifies every band at once, you may wish to investigate if it is possible to training a binary classifier that predicts if a property sale will fall within a single band.

You are welcome to explore unsupervised learning methods to identify the most informative attributes. This can include clustering methods, or principle component analysis.

2.4 Analysis and Report

The focus of this task is on your critical analysis of the data set and of the classifiers that you train. That is, providing clear and reasonable justifications for choices that you have made throughout your investigation. As mentioned in the hints, there is no correct answer. Marks are awarded for clear and reasonable justifications.

At a minimum your report should contain:

- An analysis of at least two decision tree classifiers
- A comparison between the effectiveness of the classifiers
- An ultimate conclusion

However, good analysis is presented in a simple manner. Don't detail everything you did, instead provide only describe the most informative experiments you conducted, and the most informative considerations that you have made throughout your investigation.

In conducting your analysis, you may wish to consider the following questions:

- Are the price bands a useful format for predicting sale prices for properties?
- What attributes, and attribute values, are the most informative? How did you conclude that these are the most informative attributes?
- What additional attributes or information may help improve the model?
- Are decision tree's a good algorithm to use for this learning task?
- If you removed examples (rows) from the training data, which rows did you remove and why?
- If you generated values for the missing attributes, why did you choose your approach? Could you have been be more intelligent in how you generated the values?
- If you did not generate values, how would you handle predicting the sale price band for a property that was missing values for some attributes?

2.4.1 Ultimate Conclusion

Out of all of the classifiers that you trained during your investigation, which classifier (if any) would you use to attempt to predict the sale price of Melbourne properties for the remainder of 2018? Provide a clear analysis and justification behind your choice.

3 Multi-Layer Perceptron

In this task you will need to design a Multi-Layer Perceptron (MLP) for a supervised image classification problem using the MNIST fashion dataset. This tasks has three components:

1. Designing a MLP coded in Python and Keras
2. Experimenting with different configurations/hyper-parameters for the MLP, and
3. Analysing the performance of the MLP's.

The supervised learning problem is to correctly label images from the MNIST fashion dataset, and to predict the labels of the unseen images in the test data set. One aim of this assignment is for you to become familiarised with running experiments with a MLP. Thus you are required to evaluate different setups of a MLP, especially with different numbers of hidden nodes in a single hidden layer. Since running experiments with a MLP is still mostly a trial-and error effort, you will need to determine which setup will give you the best performance, that is, classification accuracy, given a limited computational budget.

3.1 Data set

The MNIST fashion dataset consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from one of 10 classes. The data is available on canvas with the name `fashion_mnist`, and can be loaded the same way as the MNIST digits dataset.

The MNIST fashion dataset is very similar to the MNIST handwritten digit dataset, the one which was used for Lab in week 10. It shares the same image size (28x28) and the structure of training (60,000) and testing (10,000) splits. For further information about the dataset, see: `https://github.com/zalandoresearch/fashion-mnist`.

3.2 Tasks

There are 3 tasks in this part, which accounts for a total of 10 marks. The mark allocation of the 3 tasks are 2, 6 and 2 marks, respectively. Your report must contain, in no more than 5 pages, details on how you complete these 3 tasks. In addition to recording your experimental results as required, you will need to provide result analysis, including answering the questions asked below (but not limited to them). You should focus on providing insights on the behaviours of the MLP given its setups, and how you identify the best MLP model, based on what observations of the results. You may need to include some visualization of the results, in order to aid your illustration and explanation.

1. **Designing a MLP in Python and Keras code**: this includes data loading and preprocessing, model building, training, testing, and visualization, etc. For detailed information, please follow Week 10 lab exercises (in particular, exercise 2.2 is very similar to this task). Write in less than half page in your report about your experience in designing the MLP in Keras. Please include the source code in a separate file which will be one part of your submission. See Section 5 for submission instructions.
2. **Experimenting with a single-split of the data set**: the training data set (of 60,000 examples) needs to be split into two sets, the training and validation sets (with a training-validation split ratio of 90:10). You need to experiment with changing only one hyperparameter, i.e., the number of nodes in the hidden layer with a fixed number of epochs of 10. We suggest you experiment with the number of hidden nodes in the range from 78 to 784 (i.e., the number of input nodes).
   You are required to carry out 5 runs of experiments and complete your results in the table below. Your aim here is to identify the optimal (or close to optimal) number of hidden nodes for this MLP model in providing the best performance in classifying images on this data set.

Run no.	Epochs	No. of hidden nodes	Training Accuracy %	Validation Accuracy %
1	10	78		
...	10	...		
5	10	784		

   You will need to provide result analysis based on the above table. In your analysis you may wish to consider the questions such as, but not limited to:
   - How is the training process is carried out in the MLP?
   - What would be the potential cases for overfitting or underfitting?
   - What would be the training error curve vs validation error curve in a graph?
   - What can you conclude based on the observation of the curves in the graph?
   - Is it always good to use too many epochs for training?
   - How do we find out the appropriate number of epochs for the best classification performance?
3. **Experimenting with k-folds cross-validation**: in this case, we set k to 5 (in order to save time). The data set is randomly split into k parts (or folds), set one fold aside for testing, train the MLP on the remaining $k-1$ folds and evaluate it on the test fold. This process is repeated k times until each fold has been used for testing once.
   Since from the single-split data experiment in Task 2 the optimal number of hidden nodes has been identified. In this part, you can assume the number of hidden nodes is fixed with this number. Now you will need to use the k-folds cross-validation procedure to get a more reliable estimate of the performance of the MLP model. Please record your runs (on using different folds) in the following table:

Fold	Epochs	No. of hidden nodes	Cross-validation Accuracy %
1	10	identified from Task 2	
...	10	...	
5	10	...	
		Mean	
		Standard deviation	

   Using the results of the cross-validation, in your report:
   - Explain the reasoning behind k-folds cross-validation, and in what circumstances it is more beneficial.

- Explain the results in the above table, and whether or not the k-folds cross-validation has been shown effective.
- Assuming that now you have found the best MLP, evaluate this MLP on the test set and record the test accuracy in your report.

4 Compare, Contrast, and Review

In this final task, in no more than 2 pages of your report, provide an analysis comparing and contrasting the two machine learning algorithms that you have investigated. 5 marks are allocated to this task. In particular, consider the following questions:

- What learning problems are each of these algorithms (decision trees and MLP) suited towards?
- What conclusions from the two supervised learning tasks can you use and apply on learning problems that you might encounter in the future?
- What are the strengths and limitations of each algorithms on the learning problems you have investigated?
- Could you use a decision tree leaner for MNIST fashion data set image classification problem?
  - If yes, how do you expect the decision tree will perform, such as its classification accuracy?
  - If no, then why is it not possible to use a decision tree learner?
  - For either answer, are the features of the MINST fashion data set useful for a decision tree? Alternatively, what additional features could be constructed?
- Could you use a MLP for the prediction task on the Melbourne property price data set?
  - If yes, how do you expect the MLP with perform, such as its classification accuracy?
  - If no, then why is it not possible to use a MLP?
  - For either answer, are the features of the Melbourne property price data set useful for a MLP? Alternatively, what additional features could be constructed?
- What alternative machine learning algorithms do you think would result in a better (or equivalent) performance on each of the data sets?

5 Submission

Your submission is due ***Wednesday October 10th, 11.59pm (Week 12)***. Only one member of your group should submit the necessary files. (Please do not spread your submission across both members of your group.)

Your group must submit three files:

1. A *single* report file (PDF format), no more than 12 pages long, named "`report_studentID1_studentID2.pdf`". The report should contain your conclusions and responses for each of the three tasks:
   - Supervised learning for predicting the sale price band of Melbourne properties
   - Supervised learning for image classificatin with the MNIST fashion data set
   - Compare, Contrast, and Review of the learning problems
2. The python code (or Jupyter Notebook) for the Decision Tree task, named "`dt_studentID1_studentID2.ipynb`".
3. The python code (or Jupyter Notebook) for the MLP task, named "`mlp_studentID1_studentID2.ipynb`".

6 Marking Criteria

The marks are divided among the tasks as follows:

Marks	Task
5	Decision Tree task: code & methodology
5	Decision Tree task: report
2	MLP task 1: Keras code
6	MLP task 2: Single-split data
2	MLP task 3: k-folds cross-validation
5	Compare, Contrast, and Review task: report
25	Total

Unless there are exceptional circumstances, both members of the group will receive the same mark.

Each task will be assessed according to a similar standard. Use the following rubric to help you determine how to approach the assignment.

Criteria	**Excellent**	**Good**	**Fair**	**Poor**
Code, Approach & Methodology	Code is styled and organised well, following general good programming practices. Methodology and logic of the approach is easy and clear to follow.	Code is styled and organised reasonably, following general good programming practices. Methodology and logic of the approach is generally possible to follow after some work.	Code is styled and organised poorly, not following general good programming practices. Methodology and logic of the approach is difficult to comprehend.	Code is styled and organised poorly, not following general good programming practices. Methodology and logic of the approach is highly difficult to comprehend.
Report & Critical Analysis	Report is clear and concise, and well structured. It is easy to fully understand the rationale for approaches taken. Approaches are described well, training and testing setup is described, and parameter settings are justified. Tables, figures and other visualisation are easy to read and interpret.	Report is reasonably clear, possibly verbose, and reasonably well structured. It is reasonably easy to understand the rationale for the approaches that are taken. Approaches are described well, training and testing setup is described, and parameter settings are mostly justified. Tables, figures and other visualisation are easy to read and to interpret.	Report is difficult to follow in places, verbose and does not flow well. It is adequately structured, but it is difficult to understand the rationale of the approaches that are taken. Approaches are described but not in much detail, training and testing setup may be described, and parameter settings are not really justified. Tables, figures and other visualisation are either too small or difficult to interpret.	Report is difficult or impossible to follow. It is barely structured and it is difficult to understand the rationale of the approaches that are taken. Approaches are barely described, training and testing setup is likely to be missing, and parameter settings are not justified. Tables, figures and other visualisation are either too small or difficult to interpret.