COMP9417 - Machine Learning
Homework 2: A look at Feature Importance

Introduction In this problem set, you will explore feature importance methods applied to different machine learning models and compare their effectiveness in selecting relevant features. The focus will be on empirical evaluation and interpretation rather than theoretical derivations.

Points Allocation There are a total of 30 marks.

- Question 1 a): 2 marks
- Question 1 b): 2 marks
- Question 1 c): 3 marks
- Question 1 d): 3 marks
- Question 1 e): 1 mark
- Question 1 f): 3 marks
- Question 1 g): 2 marks
- Question 2 a): 1 mark
- Question 2 b): 3 marks
- Question 2 c): 3 marks
- Question 2 d): 1 mark
- Question 2 e): 3 marks
- Question 2 f): 3 marks

What to Submit

- A **single PDF** file which contains solutions to each question. For each question, provide your solution in the form of text and requested plots. For some questions you will be requested to provide screen shots of code used to generate your answer — only include these when they are explicitly asked for.
- **.py file(s) containing all code you used for the project, which should be provided in a separate .zip file.** This code must match the code provided in the report.
- You may be deducted points for not following these instructions.

- You may be deducted points for poorly presented/formatted work. Please be neat and make your solutions clear. Start each question on a new page if necessary.
- You **cannot** submit a Jupyter notebook; this will receive a mark of zero. This does not stop you from developing your code in a notebook and then copying it into a .py file though, or using a tool such as **nbconvert** or similar.
- We will set up a Moodle forum for questions about this homework. Please read the existing questions before posting new questions. Please do some basic research online before posting questions. Please only post clarification questions. Any questions deemed to be *fishing* for answers will be ignored and/or deleted.
- Please check Moodle announcements for updates to this spec. It is your responsibility to check for announcements about the spec.
- Please complete your homework on your own, do not discuss your solution with other people in the course. General discussion of the problems is fine, but you must write out your own solution and acknowledge if you discussed any of the problems in your submission (including their name(s) and zID).
- As usual, we monitor all online forums such as Chegg, StackExchange, etc. Posting homework questions on these site is equivalent to plagiarism and will result in a case of academic misconduct.
- You may **not** use SymPy or any other symbolic programming toolkits to answer the derivation questions. This will result in an automatic grade of zero for the relevant question. You must do the derivations manually.

When and Where to Submit

- Due date: Week 7, Monday **March 31st**, 2025 by **5pm**. Please note that the forum will not be actively monitored on weekends.
- Late submissions will incur a penalty of 5% per day **from the maximum achievable grade**. For example, if you achieve a grade of 80/100 but you submitted 3 days late, then your final grade will be $80 - 3 \times 5 = 65$. Submissions that are more than 5 days late will receive a mark of zero.
- Submission must be made on Moodle, **no exceptions**.

Question 1. Explore Model-Based Feature Importance

Throughout this question, you may only use Python. For each sub-question, provide commentary (if needed) along with screenshots of the code used. Please also provide a copy of the code in your solutions.py file. For fitting models, always use a random seed (or random state) of 4 for reproducibility.

(a) Generate a dataset of two classes using `sklearn.datasets.make_classification`. It should have 1000 observations, 20 features. Set 5 of those features to be *informative* (important), and the rest as *redundant*. Be sure to set the `shuffle` parameter to False, so that the informative features are listed first. Normalize your data using `sklearn.StandardScaler()`. Then, fit a decision tree (using entropy as the criteria for splits) to a shuffled version of the data[1] using `sklearn.tree` model `DecisionTreeClassifier`, and using its `feature_importances_` method, report how many of the actually important features are found in the top 5 important features by the decision tree. Plot a histogram with x-axis showing the features ranked in decreasing order of importance, and the y-axis showing the feature importance score. Use a random seed of 0 when generating the data for reproducibility. Use a random seed of 0 when shuffling the data, you can use `shuffled_idxs = np.random.default_rng(seed=0).permutation(X.shape[1])`.

(b) Provide a detailed explanation of how the feature importance (a.k.a. Gini importance) in the previous question are computed; use formulas to explain the exact calculation. Further, answer the following:

1. What feature importance score is assigned to a feature that is not used for any splits of the tree. Why?
2. What does a feature importance of 0.15 mean?

(c) In order to obtain a more accurate picture of how good decision trees are at finding important features, we will repeat the experiment in part (a) a large number of times. Repeat the experiment a total of 1000 times. In the i-th experiment, use a random seed of i when creating the data set, where $i = 1, 2, \ldots, 1000$. For each trial, record how many of the actually important features are identified. Provide a histogram of this metric over the 1000 trials. What do you think about the ability of decision trees to pick out the top features? Report the average number of good features recovered over the 1000 trials.

(d) Repeat part (c), but now use logistic regression with no penalty. Do this once with and once without scaling the feature matrix. As a feature importance metric, use the absolute value of the coefficient of that feature. Plot a histogram as before and report the average number of features recovered over the 1000 trials. Compare the scaled and non-scaled versions. How does logistic regression compare to decision trees?

(e) Does scaling features affect the result for decision trees? Explain.

(f) We now want to assess how often the two models (Decision trees and logistic regression (with scaling)) identify the same features as being important. Using the set-up of part (c), for each trial, record the number of overlaps for the top-5 ranked features for each of the two models. Plot a histogram of the number of overlaps over all trials. For example, if on a particular trial, DT has $[1, 2, 3, 4, 5]$ in its top-5, and Logistic regression has $[1, 2, 6, 7, 8]$, the number of overlaps for this trial is 2.

(g) The approaches considered so far are called "model-based" feature importance methods, since they define importance with respect to a particular algorithm/model being used. Discuss some

[1] The reason we do not shuffle the data when creating it is that we want to be able to know which of the features are the most important (first 5). We do not want to give the algorithm the ordered features as this may inflate the algorithm's ability to find important features, it may just break ties by looking at which features come first.

potential disadvantages of using a model-based approach if your goal is to uncover truly important features, referring to the previous exercises for evidence. For example, suppose that you are studying a rare genetic disease and that the 20 features represent specific genetic features, only 5 of which are truly associated with the disease. Further, discuss the effect of the number of redundant features used when creating the data set.

Question 2. Greedy Feature Selection

We now consider a different approach to feature selection known as backward selection. In backward selection, we:

1. start with all features in the model
2. at each round, we remove the j-th feature from the model based on the drop in the value of a certain metric. We eliminate the feature corresponding to the smallest drop in the metric.
3. we repeat step 2 until there are no features left.

(a) Why do you think this is referred to as a greedy feature importance algorithm? What do you think are some of the pitfalls of greedy algorithms in this context?

(b) Using the same set-up as in Question 1 part Q1 (a) write code implementing the backward elimination algorithm. Use a logistic regression model with no penalty, and the same metric as in Question 1 part (d). Be sure to generate the data without shuffling but then to shuffle the data before fitting the model. Report the remaining features at round 15 (that is, when only 5 features are left). How many of these are actually important features?

(c) Repeat part (a) for 1000 trials (similar to what is done in Q1 (c)). Plot a histogram of the number of important features recovered, and report the average number of recovered features.

(d) Another approach is called best subset selection. This model generates all possible subsets, trains a model on each subset, evaluates the performance and returns the subset with the highest performance. For example, at the t-th round, we consider all subsets with t features. How does this algorithm compare to backward selection? Will it always outperform backward elimination? What are some disadvantages of this approach?

(e) Implement best subset selection in code. Repeat part (c) using your best subset implementation. For computational reasons, set all parameters as in Q1 part (a), but with only 7 features, 3 of which are to be taken to be informative, and the rest to be redundant. Plot a histogram as before and report the average number of recoveries. Comment on your results.

(f) An alternative approach to feature importance is known as the Permutation Feature Importance score, implemented in `sklearn.inspection.permutation_importance`. Read the documentation and provide a detailed explanation of how permutation importance works. Compare it to the techniques studied so far in this homework, and explain why we refer to this as a model-independent metric. Do you think it's more or less fair to compare logistic regression and decision trees using this metric? Finally, using the sklearn implementation, re-do part Q2(c) using this new feature importance metric. Similar to before, use 20 features, with 5 to be set as informative and the rest as redundant.