COMP 309 — Machine Learning Tools and Techniques
Assignment 3: Kaggle Competition

Student ID: 300417869
Name: Karu Skipper

Core: Exploring and understanding the Kaggle process [50 marks]

Business Objective:

Develop the best possible machine learning system to predict the completion times of given tramping tracks in NZ.

Dataset Exploration:

Let's analyse our raw CoreDataSet-Train.csv dataset to understand what we will be investigating in this question.

Id	difficulty	Shape__Length	X	Y	Class
1	Easy	6067.516674	168.6052622	-43.98688424	0
2	Easy	1190.301175	174.9241203	-41.34988879	0
3	Easiest	516.548047	169.5614119	-46.5176704	0
4	Advanced	11422.56164	174.802487	-38.47091454	1
5	Advanced	3330.867075	168.772812	-45.137258	1
6	Easiest	718.0697865	170.1131962	-46.00161732	0
7	Advanced	4557.861652	172.5518639	-40.63409605	0
8	Advanced	2992.429545	177.8007948	-38.98481859	1
9	Advanced	3284.209167	171.22407	-43.63237	1
10	Easiest,Easy	975.8485552	171.3275657	-42.11516818	0
11	Easiest	5921.0808	174.1083856	-39.25751982	0
12		3741.219226	174.4652588	-36.84250676	1
13	Intermediate	71926.56678	172.3028297	-40.88644937	2
14	Advanced,Expert	11579.62589	167.4592865	-45.83573596	1
15	Advanced	32189.20895	171.0091276	-43.10568786	2
16	Advanced	125082.131	172.4395031	-41.31655782	2
17	Advanced	18593.83517	168.9647944	-45.15107353	1
18	Easy	329.6915735	167.1564114	-45.46365128	0
19	Advanced	1195.007303	168.8168876	-45.05986931	1
20	Advanced	35089.00364	174.8585441	-39.34062725	2
21	Easy	1706.756362	172.9402163	-40.83876811	1
22	Easy	1590.882815	172.8618034	-40.88149487	0
23	Easiest	565.4305424	169.4859153	-46.5040545	0
24	Advanced	2946.382344	177.3881146	-38.27966379	1
25	Easy	17562.7012	172.7835552	-34.45340293	1
26	Advanced	4047.873045	176.3006879	-39.62729289	1
27	Intermediate	4833.243419	168.1267263	-44.81812323	1
28	Advanced	4694.175046	170.4064419	-45.81564491	1
29	Easy	719.6814706	173.6275539	-35.23992101	0
30	Easy	3965.725559	176.2644738	-39.72037491	1
31	Expert	16096.67815	172.643854	-42.37819934	1
32	Easy	9261.72863	171.8812823	-41.609507	1

<u>The six attributes of a given instance:</u>
This section was important to investigate as it provides insight before we apply algorithms to the dataset.

1: Id)
Looking at the figure above, the first attribute is the *'id'* of the instance, which lists each instance (tramping track) in order. {1, 2, 3 ...}.

2: Difficulty)
The second column identifies the *'difficulty'* of the track, which classifies each instance into different categories including; {'Easiest,' 'Easy,' 'Intermediate,' 'Advanced,' or 'Expert.'} *There are also combinations of these categories to help further increase accuracy of the difficulty of the track e.g. {'Intermediate, Advanced.'}*

3: Shape Length)
The third column identifies the *'shape length'* of the instance. This feature will be important to investigate and keep in our dataset as it provides a length of each track. *Which could affect the order a person may finish a given track, changing the time class it may be associated with.*

4 & 5: X and Y)
The next two attributes are X and Y values of each instance. I am unsure how relevant this attribute may be; however, it could affect the difficulty of the course.

6: Time Class)
Finally, the attribute in the last column is the 'time class' the instance falls into. This is one of the more important attributes in our investigation. This class ranges from {0 – 2}, based off the time a person may take to complete a given track.

<u>Highlights of the dataset exploration:</u>

- The last attribute '*class*' is of numerical type, this has been changed to nominal so we can classify our data. *Not changing this would mean classification methods would not apply to our dataset, which means we would not be able to make predictions for the time class.*

- I've had to change the combination of difficulty attributes e.g. "Advanced, Expert" as converting this to a .csv format confuses what attribute this difficulty is under (because of the comma). *This would mean an extra 'empty' attribute would be added to the end of the instance. This would incorrectly classify this instance, either ignoring it or adding incorrect values to our classification. Resulting in data loss or even accuracy loss, which is not ideal for our predictions.*

- The X and Y ranges are very different for each instance, regardless of what difficulty the track is. *This means that our model may have a harder time classifying our data. For e.g. Instance 1's X and Y values are very similar to Instance 2's X and Y, even though they are the same difficulty. Feeding this into an MLP for example, may change what time class the predicted instance may fall into.*

- *I have removed ID from the dataset as it isn't relevant to our investigation of helping us classify the instances.*

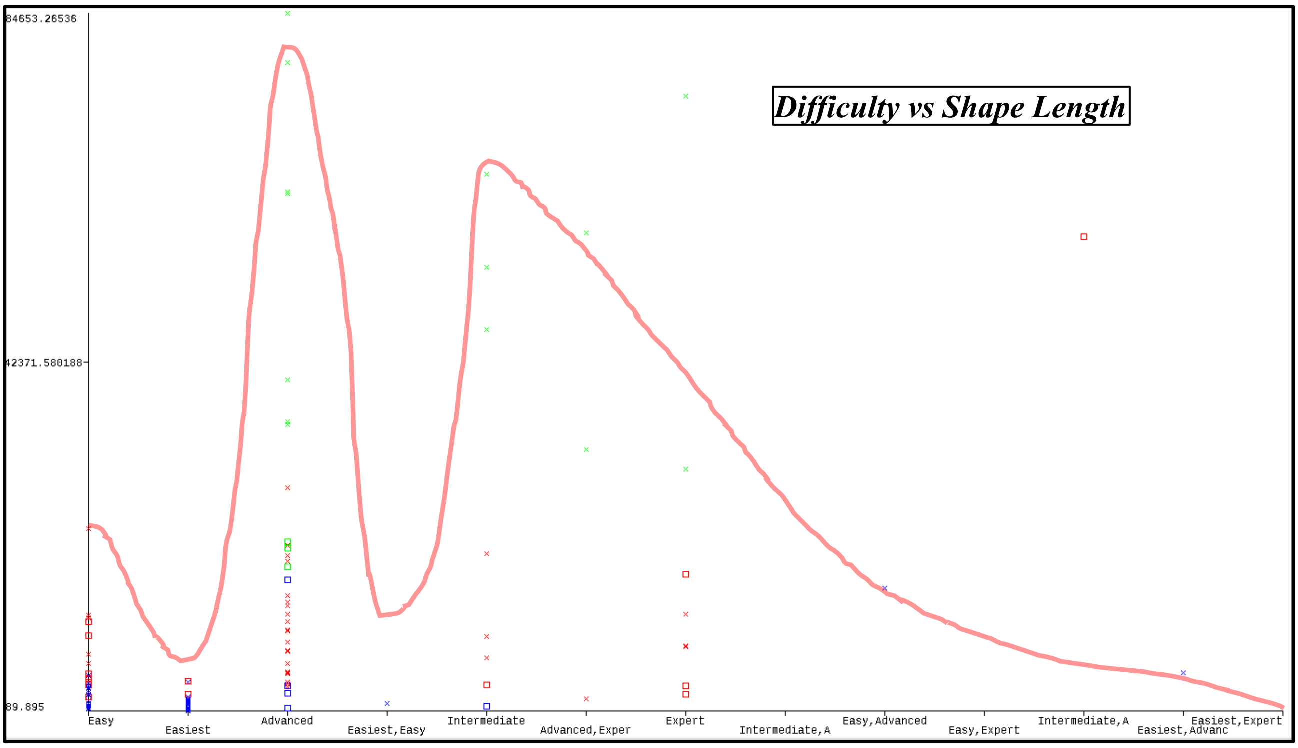
Patterns between attributes of an instance (Using WEKA and scikit):

Let's look at patterns between the attribute's difficulty and shape_length that help classify our instances.

From quick investigation, I noticed a large amount of '*easier tracks*' seem to fall into the lower-range of the *'shape_length'* attribute and vice versa for '*more advanced*' tracks. (Except for some outliers).

If we investigate this further, we can visually see the difference between the first '0' time class and the second '1.', based off *difficulty* and *shape_length*. However, it is harder to see the difference between the next time class '2' as the *shape_length* ranges are very similar. This is where other attributes would contribute to helping classify the data.

From the graph below we can see a significant amount of '*advanced*' to '*expert*' instances are greater in '*shape_length*' than easy to easiest.



This graph indicates that harder tracks are usually in the upper-range of *shape_length* and the easy tracks are usually in the *lower-range*.

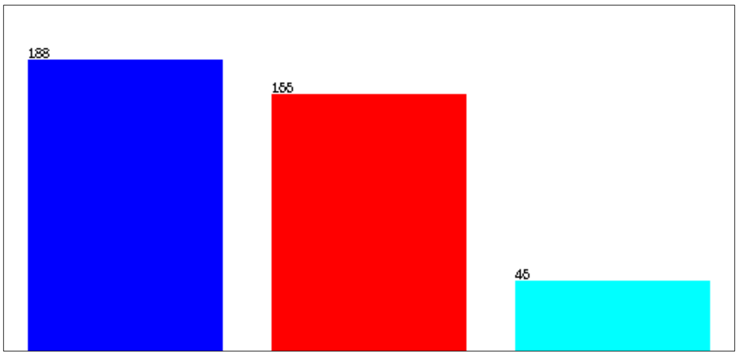
Let's account for other complicated patterns that could impact our investigation.

<u>For this section I am using scikit to quickly analyse unusual patterns that may be occurring.</u>

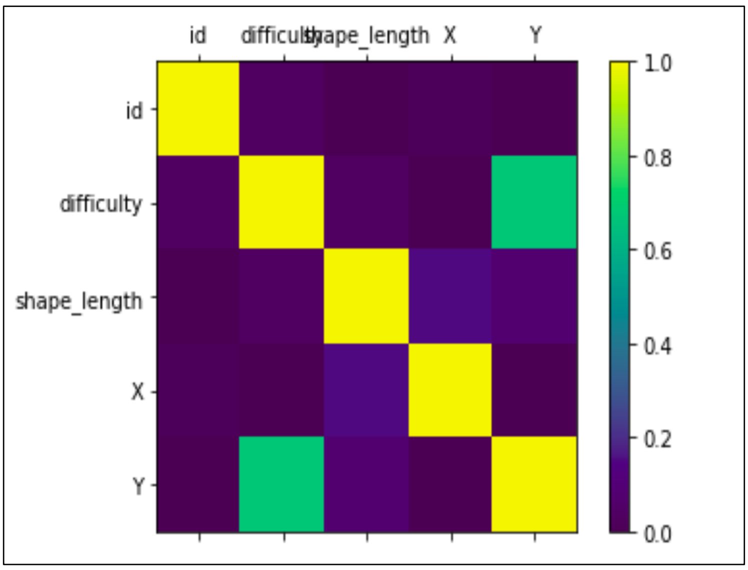
First, I ran a null scan and found there is only one missing value for difficulty in the dataset. This has no pattern in relation to other attributes in the dataset and will be removed in pre-processing.

Another pattern to note is the class imbalance that is in our training set. This affects the accuracy of our results and can be highly unreliable when testing.

```
In [5]: data.isnull().sum()
Out[5]: Id              0
        difficulty      1
        Shape__Length   0
        X               0
        Y               0
        Class           0
        dtype: int64
```



One important finding of my work;
Below is a Correlation Matrix Plot of every attribute in the dataset (I am using the scikit-learn tool to display the plot). This graph will tell us if there are any important correlations between attributes.



Visually, there is a positive correlation of about 0.7 (strength) between the *'difficulty'* of a course and the *'Y value'* of a course. (The greater the y-value, the greater the difficulty). Other than this, there are no visible correlations I can identify from the graph.

To conclude, from this investigation I found '*advanced*' to '*expert*' instances are greater in '*shape_length*' than *'easy'* to *'easiest'*. But also, the instance count is significantly lower for more 'advanced' tracks, which could affect the reliability of these results. I have also found a correlation between the *'difficulty'* of a course and the *'Y-value'* of a course. This evidence helps our algorithm make predictions of each time class we are classifying the instance into. If we relate back to our business understanding, we can safely use this data in relation to predict a given track time in NZ.

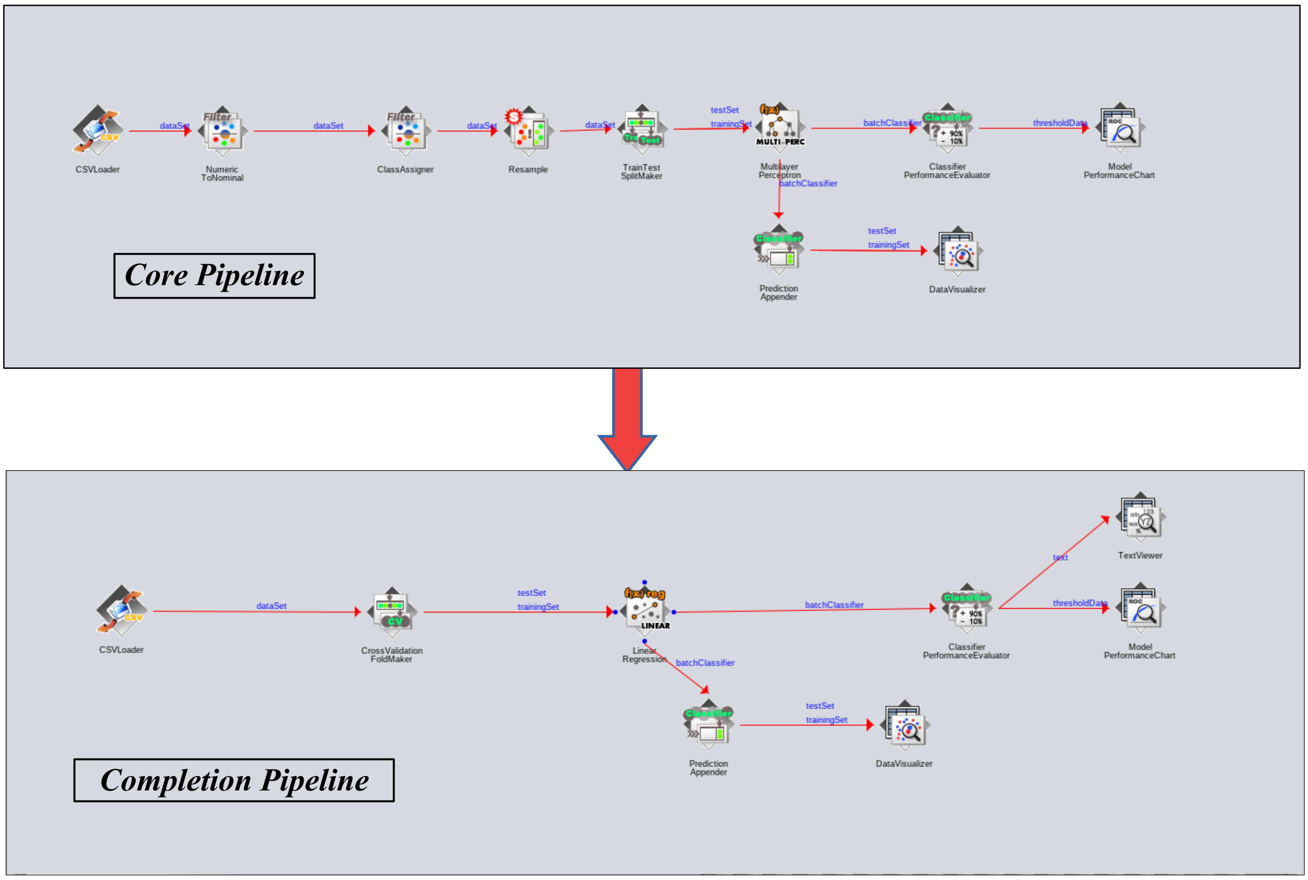
2.3 Completion: Developing and testing your machine learning system [40 marks]

Datasets I will be analysing in this section:
Training set one: Challange-WB-T1
Training set two: Challange-WB-T2

The First System (Single training set)

Initial design in WEKA (Pipeline saved in '/completion/Pipelines and Models/').

Note: This is the original system I used for the core part of the assignment, from this I will be altering and adapting it to suit the completion section.



Description of the System
Above is the *'new'* Pipeline that takes a singular dataset (Challange-WB-T1) which is split into its own training and test set and parsed through the Linear Regression algorithm. I have chosen this initial system because of the simplicity of its design and architecture.

This system was easy to use, small and robust. Because of this, I could experiment with different parameters and Pre-Processing techniques quite easily. Meaning it wasn't hard to output a model that we could use on our test set. It was also easy to abstract and alter from the original Core Pipeline.

How we can improve this?
The Linear Regression Model outputs a correlation coefficient of about 0.409, with the single T1- Training set. This model performs below average against the online Kaggle Competition (Completion) test, with a result of around 407.30889. This system will need improvements if we are to increase the accuracy of this model.

Insight from Core System
Because we are using different data for the Completion section of the assignment, we can no longer apply some Pre-Processing techniques we were using in Core. The data in the Core section of the assignment is time classification prediction, whereas Completion is more regression-based. This means we must change what technique we apply to our datasets. For this second system, I changed from an MLP to Linear Regression.

The Second System (Combined Training Sets)
(Using WEKA and scikit learn)
The second system included exploring different algorithms that we could apply to our two training datasets.

Pre-Processing of the Second System:
Before we do this, we must consider the removal of unnecessary attributes and data that could affect our model. Now that we are working with two training sets which contain more attributes and instances, we need to remove anything that could be taking up space or affecting our results. This was one for the issues with the dataset provided.

Refining data into similar completion times:
This was the biggest issue with the provided training datasets. I identified this attribute as 'messy' due to the variation in 'time completions' that contained string extensions, making our data hard to apply an algorithm to. To fix this, we use a Pre-Processing technique called 'data cleansing.' (described later).

I decided to combine both the training sets to give a greater instance count, as well as instance/attribute variation. Because of the high number of unnecessary attributes, I decided to manually remove them (during pre-processing) before adding them into the Pipeline. This makes it easier to manipulate data once it's been loaded. In summary, combining these datasets came with the positive result of more variation, larger data and more instances. However, this lead to some overfitting, unbalancing the variation of our data.

Here is data before and then after applying excel functions and WEKA pre-processing techniques.

completionTime
30min
20min
2 hr
3 hr circuit
4 - 5 days
6 - 8 hr
1 hr 30 min
1 hr 30 min
30 min to 1 hr
1 hr return
3 hr
40 min return
20 min - 1 hr

completionTime
240
330
960
90
45
60
150
150
1920
60
480
1200
30

Here is the function I used for identifying certain strings within a cell.

=ISNUMBER(SEARCH(substring, text))

In doing this, I accounted for walking 'days' as 8 hours, which was then converted into minutes. I repeated this for other values of this attribute, looking for keywords such as 'hr,' 'min,' returning a value in minutes. After this, if this returned true for substrings such as 'hrs,' it would be times by 60 to convert it into a separate time class.

In this second system, I explored familiar techniques like the '*Random Forest*' algorithm, which is part of the Tree category. I chose this because our data's completion time wasn't categorical, it was numerical based. This should create a better accuracy model.

Comparison to the First System:

Comparative to the first system that included only used one training set, I have combined both training sets into one. This system scored slightly higher than the original system, resulting in 332.22026 on the leader board. That's a 75.082 decrease in mse. However, we are trying to aim for the lowest possible result and the best model, and I believe this can be achieved through multiple iterations of this system.

More experiments (Sci-kit Learn):
I was interested to test different programs to create the best possible model. This led me to experiment with sci-kit learn and its machine learning tools. Firstly, I made predictions using Linear Regression but noticed other algorithms were also suited to the datasets criteria. (I have listed the python program in the '*\completion\Systems\'* folder).

--Images and output needed

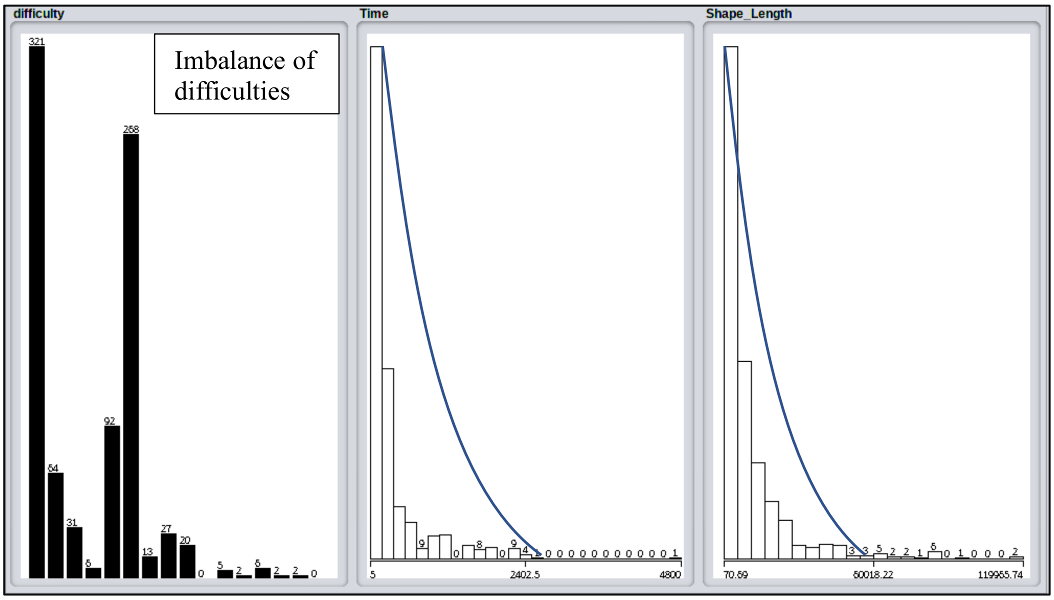
The Third System (Merging outsourced datasets)

Outsourced dataset reference (DOC_Tracks):
http://doc-deptconservation.opendata.arcgis.com/datasets/59a193dfa57d4f87b7ef157a412907d3_0

Description of the System
This is where I used knowledge from the previous systems to improve the accuracy of my model. This included an outsourced dataset to help with my predictions and correlation coefficient. To iterate over what this system includes, I will be combining the two given datasets, applying extra attributes from the outsourced dataset and running the *'Random Forrest'* algorithm, which will hopefully lower the mse and raise my ranking on the Kaggle leader board.

Pre-Processing of the third system
Because of the previous systems applied techniques, there wasn't much to alter for this systems pre-processing. However, I had to merge an outsourced dataset, (which was done through the WEKACLI) . This included merging *'shape_length'* from the *DOC_Tracks.csv* to try and increase the score and model accuracy against the given test set. In the second system we combined the datasets, which gave a greater instance count but also resulted in a greater imbalance between difficulties.

Before implementing our outsourced data, I needed a solution for the imbalance of instances in our dataset. Below is an example of the three attributes included in the T2 training set (After attribute removal). Visually, we can see a heavily left-skewed graph for time and shape length (Longer time, longer shape length). But 'difficulty' seems to be fluctuating. This means we have to apply a 'Resample' to the datasets so we can have fair testing. Otherwise, outliers could affect our prediction and model. This technique was applied through *WEKA -> Filters -> Unsupervised -> Instance -> Resample.*



Comparison to the two other systems:
This system was abstract and operated quite differently to the other systems but contained a lot of similar techniques of data usage. Most importantly, it resulted in similar results to the second system. I have created a pipeline in ‘*\completion\ Systems\’* using sci-kit learn and WEKA.

--Images and output needed use sci kit

<u>The System of the Three Systems and Why:</u>

<u>System two:</u>
System two was the best of the three discussed above, with both model accuracy and leader board ranking. System one is unreliable because it utilises the only one training set. System three is complex and balances variables but still outputs similar results to system two. System two still contains the robustness of system one and maintains the simplicity without altering attributes too much. System two, uses two training sets, limited filters and few pre-processing techniques.

Between these two datasets we only use five attributes for our findings. This is a low amount of attributes for a combined amount of 900 instances. Although this is true, the attributes we are using are more than sufficient for our findings. Adding another dataset ended up confusing my model and not supporting my business understanding.

System two wasn’t overfitting data and I believe was the most reliable of the created models. Other systems could be an issue as they are unreliable or refined to the specific test set.

2.4 Challenge: Reflecting on your findings [10 marks]