Machine Learning Techniques for Fish Breeding Decision Making

Rose Taylor
School of Engineering and Computer Science
Victoria University of Wellington
Wellington, New Zealand
taylorrose3@myvuw.ac.nz

***Abstract*— The New Zealand Institute for Plant and Food Research has been working on creating breeding programs for the Australasian Snapper (Chrysophrys auratus) to breed snappers that mature faster and are high quality. One of the breeding program goals is to select individuals that produce quick-to-mature offspring. To accomplish this, they collected the genomic makeup of snappers into a dataset. However, the collected data has missing values in some features, which require imputation to enable use of those features to classify fish that grow faster and slower. As the genes responsible for controlling the growth rate in Snapper are currently unknown, the dataset must maintain as many of the features as possible to enable identification of the genes most likely to control the snappers' growth rate. This project investigated whether the data imputation methods used impacted the ability of a machine learning classifier to predict the growth rate and, if so, how different imputation methods performed. This project implemented five imputation methods, specifically Most Frequent imputation, K-Nearest Neighbour (KNN) imputation, Multiple Imputation by Chained Equations (MICE), a KNN approach using domain information, and a cascading KNN imputation method using domain information. The KNN and MICE approaches have two different parameter settings for imputation. This project evaluated these imputation techniques using a Random Forest classifier. The results showed that all imputation methods are robust to the test train split and random state used in the random forest classifier. The classification accuracies were similar between the imputation methods. Despite differences being displayed in split datasets, the complete datasets p-value calculations confirmed no significant differences in overall result. These results indicated that domain-based imputation approaches did perform better than other imputation techniques indicating that using domain-based imputation techniques could improve the overall classification accuracy. Lack of significant differences between the classification accuracies are caused by the number of features being so great that there is little overlap in the features selected by the Random Forest classifier and the features that are selected by the majority of the trees help account for majority of the classification accuracy.**

Keywords—data imputation, machine learning, fish, KNN, MICE

I. Introduction

This project is part of the Australasian Snapper (Chrysophrys auratus) breeding program begun in 2016 by The New Zealand Institute for Plant and Food Research Limited [1]. The breeding program aims to identify the genotypes responsible for controlling the growth rate in snapper fish so that when planners breed snapper, the offspring grow to a harvestable size faster [1]. The snapper growth rate is slow. It can take 3-5 years for a snapper to grow large enough to be legally caught commercially for food [2], [3]. This project aims to solve the problem of how to handle missing data within the genetic dataset. Since the genes responsible for influencing the growth rate in snapper are, as yet, unknown, the handling of missing data is vital to creating a model that can accurately predict the growth rate of a snapper based on its DNA [1].

This project investigated whether the use of different data imputation methods impacts the ability of a machine learning classifier to predict the growth rate and, if so, how different imputation methods perform. The project implemented and evaluated different techniques for imputing the missing data values within the original dataset. This project is part of an existing project being worked on by multiple researchers, both within Victoria University of Wellington and The New Zealand Institute for Plant and Food Research Limited. This project directly built on the project "Machine Learning Techniques for Fish Breeding: Finding Genetic Variants Related to Growth" by Ze Chen [4]. This project is crucial as there are many missing values within the genomic dataset, which contains many features. The project aimed to determine which genes are most likely responsible for the growth rate in snapper fish. Therefore, features (genes) cannot be removed from the dataset because of missing data as it is currently unknown which genes are responsible for Snapper growth rate.

In [4], the missing data was imputed by using with the most frequent value in each feature. Because this data will be used in machine learning models to identify which genes are responsible for controlling the growth rate in snapper, it is necessary to replace missing values in a way that will not lead to distortion of any of the attributes of the data. Distortion will lead to misclassification of the features responsible for controlling the growth rate of snapper. This project implemented a variety of data imputation methods to create new datasets and evaluated the datasets to assess how the accuracy of the imputation methods.

Successfully identifying data imputation methods that work with this type of data will benefit other projects with similar datasets with missing values. Despite the dataset being high dimensionality, with over 18000 features, keeping as many features as possible is vital. Because there are only 1200 instances, 1100 of which are usable, performing data imputation ensures that those remaining instances are high quality. Replacing the missing values with the most frequent value in the feature was not ideal, as it distorted the data. More complex data imputation methods consider and evaluate the most likely value of the instance, so those imputed features are less likely to distort the data than using the most common value does.

The model created in this project can be used to improve the breeding programmes of Snapper fish for food by identifying individuals with a higher genetic likelihood of growing faster and producing offspring that will grow faster. Fish will reach a harvestable size sooner by growing quicker, allowing them to be harvested for food sooner [2]. The overall project of classifying the genes responsible for growth in Snapper will be able to help in achieving two of the United Nations Sustainable Development Goals, specifically, goal 2, "End, hunger, achieve food security and improved nutrition and promote sustainable agriculture" and goal 14, "Conserve and sustainability use the oceans, seas and marine resources for sustainable development" [5], [6]. These goals are supported as growing fish faster means more fish are an adequate size for food sooner (goal 2). Fish growing faster commercially would also help prevent illegal fishing as less fish would need to be illegally fished (goal 14).

To discover whether the imputation method influenced the classification accuracy 5 different methods were used for imputing missing values: most frequent, KNN, MICE, domain KNN, domain KNN version 2. All versions of KNN imputed datasets using two different values of k (3 or 5) and for MICE two different values for the maximum number of iterations were given (10 and 50).

For this project, a domain-based weighted function was created. This takes positional information relating to each feature to create weights for a specific selected feature based on the difference in position relative to other features. This domain-based weight function ensured that features further away from the selected feature have less impact on the imputation of the current feature and closer features have more impact on the imputed value. This was based on biological concepts of crossover and inheritance of DNA where genes closer to the current gene/feature are more likely to be inherited together and so are less likely to be different from each other [7].

This was applied to two domain-based versions of KNN created for this project. The first version imputes each missing value for each fish using the domain-based weight function. The function is repeatedly run for every feature with a missing value in each instance. The second version of the domain based KNN imputes the features with the fewest missing values first and for every feature imputes the values for all instances.

Stratified 10-fold Cross Validation was conducted using a Random Forest classifier 30 times on every dataset to test whether classification accuracy was robust to different random state for the test split and random forest. Friedman's Test was then applied to identify whether there was a significant difference in classification accuracy between the imputed datasets.

Overall, the results showed that machine learning imputation methods used did not have a significant impact on the classification accuracy of the model despite there being a significant difference in the average accuracy of the chromosome/scaffold datasets. Regardless of that, in general the domain-based imputation methods perform better than non-domain-based imputation methods. All the imputation methods models outperform the most frequent imputation model. The most frequent imputation technique was used as the baseline method for this project due to being the technique used for imputation in [4]. MICE with a maximum of 50 iterations performed the best overall compared to the other methods but this came at a cost, as MICE was time and resource intensive. As there were differences in the imputed values, the features selected by the Random Forest classifier were investigated. It was found that the majority of the features used contained imputed values and there was overlap in the selected features. Potentially the features selected by 5 or more different imputation methods be considered good starting predictors for the class of growth rate of the fish.

II. Related Work

A. *Machine Learning for Analysing Fish Related Data*

Machine learning has been a useful tool for a variety of different tasks relating to fish over the years.

There are a wide variety of fish classification tasks that have occurred using machine learning on fish-related data. When it comes to classification-based studies several focussed on classifying the presence of schools of fish based on different data collection methods [8], [9]. Paper [8] used machine learning techniques to classify whether clusters recorded by Multibeam Echosounder were fish or other natural occurrences such as bubbles. Paper [9] used data collected from a Drifting Fish Aggregating Device and used machine learning to classify whether groups of fish were present or not. These two papers show how machine learning can be used to help classify fish in environments. There have also been some papers that classified different species of fish based on spectroscopic data recorded [10], [11]. These are more general applications of machine learning and how they relate to fish.

There are also papers which looked at using more specific data regarding fish which can be used in machine learning classifiers. Paper [12] uses data recorded from tagged fish and environmental data to classify the fate of individual fish during migration. Paper [13] uses machine learning techniques to create a classifier able to identify the age of fish based on provided information such as scales, length, weight and vertebrae. This shows that there are a variety of different ways to use machine learning to classify specific information about fish using other physical and biological related fish data.

There have also been uses of machine learning to predict other fish related things. Paper [14] uses neural networks to predict the physical characteristics length, circumference and weight of a variety of fish species based on images. Paper [15] used chemical and physical data related to fish muscle information in order to predict the mobility type of the given fish. Paper [16] has used machine learning to generate biomarkers to identify fish sex and thermal stress information. Two papers have also used machine learning to gain a better understanding of how different environmental factors affect fish [17], [18].

In recent years machine learning has been a vital tool in helping researchers understand and find links between a variety of different information about fish for many different purposes. There is demonstrated merit in using machine learning to identify genes and justifies the idea of using machine learning to identify genes related to growth.

B. *Exisiting Genetic Based Imputation Techniues For Genetic Data*

The features within the genomic dataset, "data.csv" and "new_data.csv", can be broken down into two different

categories: they are either part of a chromosome or a scaffold. The dataset called "Snapper_SNP_locations.xlsx" stores the chromosome or scaffold the feature belongs to and positions the genes have on the chromosome or the relative positions they have in the scaffold [19]. The scaffold positions are estimates of the distances between each chromosome in the scaffold and therefore should not be used when imputing the data. This is because scaffolds are collections of DNA sequences which were unable to be read as part of a full chromosome and has been stated during our discussions with experts from New Zealand Institute for Plant and Food Research Limited [19].

There is the potential to use the positional information about where the genes are on the chromosome in data imputation methods due to the existence of haplotypes.

Haplotypes are sets of DNA variants in a chromosome which are usually inherited together due to their close location on the chromosome [20], [21]. There are three different methods for deriving haplotypes but the most applicable form for this project would be population inference, which looks at other genomic information in the population to find haplotype matches [21]. Haplotypes have been used to help impute missing values in genomic data in other studies seeking to do this. Two of these studies were focused on imputing missing human genomic data, such as The Cancer Genome Atlas datasets and the 1,000 Genomes Project [22], [23]. There have also been studies on imputing data in livestock population data and for dairy cattle breeding that used haplotypes [24], [25]. In paper [25], haplotypes between the parents and offspring were compared and if there was an identifiable match the missing values would be imputed based on that. However, in this project we do not know whether the parents and offspring are included in the data, so are prevented from using haplotype imputation in this way. Haplotypes cannot be applied directly to this project as haplotypes need to be found using industry experts. However, the general concept that DNA variants are likely to be inherited together can be used in the project as it means that fish with similar DNA on a chromosome are likely to be related, so this concept can help with imputation of missing values.

In paper [22], a study was conducted into using variational auto-encoders (VAE) on datasets which are related to human DNA, such as The Cancer Genome Atlas datasets and DNA methylation datasets. Like this project's dataset, the datasets looked at in this paper are purely numeric [22]. VAE are able to learn about distributions of variables to ensure that the output generated is like the input and have been used on genomic data before [22]. However, research is needed on some more specific types of genomic data [22]. Overall, they found that VAEs were able to outperform KNN in multiple different situations. However, the differences between the datasets they used and this project's dataset is that only 15% of the features in their dataset have missing values with on average 8.5% of the values missing [22]. This is significantly different to this dataset which has a wider range of missing values, in some cases features can have over 60% missing values and more than 15% of the features have missing values. Because of having more missing values, the reported accuracy of VAEs were not applicable to this project.

C. Machine Learning Impiutation Methods

There are two main different types of data imputation methods: single imputation methods and multiple imputation methods [26]–[29]. Single imputation methods impute a missing value once whereas multiple imputation methods impute a missing value multiple times in order to find the most likely value [26]–[29]. Single imputation has flaws due to the fact it can be inconsistent in producing results if the missing data isn't missing completely at random, which can lead to biased results [26], [27]. Multiple imputation methods are less likely to be biased due to the missing value being calculated multiple different times and so allow for more variability in the results produced [26]–[29]. In this project a variety of both single and multiple imputation methods are implemented.

The single imputation methods considered for use in here were:

- K-Nearest Neighbour, which looks at the k instances most similar to the instance with missing value and imputes the missing value using the average or most frequent value in those k instances [30], [31].
- Weighted Nearest Neighbours (NN), which is a version of KNN that uses a weighting function that weights the features of the instances differently depending on a given criteria [28], [30].

The multiple imputation methods considered for use in here were:

- MICE, which fills in all the missing values as a base then for every instance makes the values missing and runs regression to calculate what those missing values in the instance would be [29]. It does this a number of times, equal to the number of iterations given and produces the final result based on the regression model [29].
- SICE, which is an extension of MICE as it runs MICE a number of times and calculates the mean or most frequent result from the different MICE results [26].
- Multiple Imputation using Nearest Neighbours, which repeats a nearest neighbour method multiple times while using the newly completed values when computing the next instances missing values [27].

The reasons for considering these methods are they have some existing implementation in Python libraries such as scikit-learn. These methods allow for the ability to incorporate domain knowledge gained from research and from talking with people from The New Zealand Institute for Plant and Food Research Limited. Weighted NN is an example of a method where domain knowledge regarding the positions of genes on the chromosome can be implemented using a custom method.

D. Evaluation and Benchmarks

The goal of this project was to discover whether the data imputation method, used on the missing data, created a significant difference between the classification accuracies of the model. The main evaluation statistics used are the classification accuracy value and standard deviation of the classification accuracies value.

To determine whether there is a statistically significant difference between the imputed values, Friedman's Test will be used. Friedman's test is able to be used on any number of groups. It uses ranked classification accuracies between the different imputation methods to determine if there is a statistical difference between the groups being looked at [32].

This test is useful for this project as it compares more than two different groups to determine if there is at least one group that is more different from the others. This allows for easy comparison between the different generated datasets.

The baseline model that results are compared to in this project is the model using imputed values with the most frequent value. This project expanded and supports a current ongoing project which used the most frequent value as the imputation type. This also means the accuracy has been compared to a model with a very fast implementation time, so conclusions can be drawn about the value of using a more computationally intensive imputation method.

E. Tools

This project was coded in Python. This project is part of a suite of related projects already using Python. Regardless, if this project had been started from scratch Python would have been chosen as it has a variety of available libraries, such as scikit-learn, which provided valuable tools [33]. Scikit-learn is a Python library which has a variety of data imputation method implementations which can be used and customised with different input parameters, along with a variety of other methods which can be used to help format the datasets and functions for calculating metrics on the data imputations [34], [35]. Using Python also allowed for custom imputation methods and additional methods to be created from scratch as needed. Modifications could be made to existing methods, providing more control over how those techniques were implemented.

Microsoft Excel was also used in this project, as more complex and detailed visualisations were able to be generated using it, as was a simple implementation of difference checks and p-value calculations.

III. Design

A. Overall Approach

The approach taken to solve this project was to implement a simple data imputation technique, a single imputation method and a multiple imputation technique to analyse the accuracy of the results. This provided a good starting point allowing additional techniques to be implemented later and providing an understanding of what techniques would be good for incorporating domain knowledge into. When the imputation methods were affected by given arguments, such as KNN results being influenced by the selected value of k, two different values were used to allow for a wider variety of results and to see if some of the methods had better potential depending on the parameters. Initially KNN and MICE were the methods selected to be compared to the baseline. KNN is a single imputation method and MICE is a multiple imputation method.

B. Dataset Design

The datasets themselves are made up of features, which are individual genes identified within recorded DNA from Snapper fish. These genes can be one of three ordinal numeric values, 0, 1 or 2 and missing values are indicated by -1. Within this project they are treated as if they are numbers as they have different meanings. When the value is 0 then the DNA for that individual's gene is identical to the baseline recorded individual. When the value is 1 the DNA for that individual's gene is slightly different to the baseline individual. When the value is 2 the DNA for that individual's gene is significantly different to the baseline individual. For each gene 0, 1 and 2 represent the same difference in DNA, meaning it is a different genotype to the baseline fish DNA. This means that distance measures such as Euclidean distance can be used, as the different values have inherent meaning, denoting that 0 is different to 1 and is more different to 2, the reverse can be said for 2 and, 1 is similarly different to 0 and 2.

There are two classes fish can be classified into for this project, slow or medium. These labels were based on the calculated growth rate of the fish, which was determined by changes in weight and length over time. There was a fast category, but these fish were removed from the data as there were too few of them to be of any use for accurate classification modelling. These labels were provided by the work done in [4].

To help with potential memory issues due to the size of the datasets, the imputation techniques were executed on split datasets, meaning the full dataset was broken down by different chromosomes and scaffolds. Once all the split datasets had the missing values imputed these datasets were recombined to form a complete dataset for analysis purposes. Splitting the dataset like this allowed for the most relevant features to be kept together and allowed for better implementation of the domain specific imputation approaches. The dataset was split into 27 different datasets which are shown in Table I along with the total number of features each chromosome/scaffold contains and the number of the features with no missing values or missing values.

For a better understanding of the generated datasets, both split and complete, stratified 10-fold cross validation was performed on a Random Forest classifier to determine their overall test accuracies. Stratified k-fold cross validation keeps the proportion of classes the same in the test set to keep the test instances the same as the population proportion. Two experiments were done to ensure that the results were not random state dependent on the Random Forest Model seed or random state for the test split of the data. This code [36] was used to conduct both experiments one after another to ensure that less time was spent generating results overall. The results of the two different tests then had Freidman's Test performed on them to determine whether there was a significant difference between the datasets generated, both for the split and complete datasets. A Random Forest classification model was chosen due to the ability to ensure that overfitting would not occur on the training data. This was accomplished by ensuring that new subtrees could only be created when there were more than 10 individuals that needed to be classified.

To gain information about the features selected by the Random Forest classifier both the random state for the test split and the random state for the Random Forest seed were set. This can be seen in code [37]. This allowed for a fair comparison between the features selected due to the same random seeds being used for the elements that were random state dependent.

The criteria for choosing which methods to use as a base for implementing domain knowledge were the time it took for the imputation method to run, the amount of memory it needed to complete and the simplicity of adding domain knowledge to the imputation method. These criteria were chosen to ensure that computation time and power usage to ensure code was environmentally friendly as possible.

TABLE I. TABLE OF THE SPLIT DATASETS AND THE BREAKDOWN OF FEATURES WITH AND WITHOUT MISSING VALUES

Chromosome/Scaffold Dataset Name	Number of Features with No Missing Values	Number of Features with Missing Value	Grand Total
CAURATUSV1 SCAFFOLD	124	3305	3429
LG1	44	630	674
LG10	43	666	709
LG11	43	759	802
LG12	29	514	543
LG13	49	658	707
LG14	41	523	564
LG15	29	558	587
LG16	43	627	670
LG17	43	461	504
LG18	34	639	673
LG19	38	527	565
LG2	41	726	767
LG20	36	494	530
LG21	29	552	581
LG22	37	592	629
LG23	22	404	426
LG24	23	310	333
LG25	6	52	58
LG3	47	713	760
LG4	51	649	700
LG5	46	685	731
LG6	35	583	618
LG7	46	680	726
LG8	38	664	702
LG9	31	514	545
SUPER SCAFFOLD 182	3	7	10
Grand Total	1051	17492	18543

IV. IMPLEMENTATION

A. Machine Learning Impuation Techniques Implemented

Initially the Most Frequent Value, KNN and MICE techniques were implemented to provide general statistics of performance. Imputation using the Most Frequent Value was completed by calculating the most frequent value for each feature using the non-missing values and substituting that value in for the missing values for each feature.

KNN imputation was more difficult to implement due to needing to code the function from scratch. The version provided by scikit was unable to be used in this context due to it imputing the missing values using the average value of the nearest neighbours rather than providing the option to impute using the most frequent value, which is needed due to the categorical nature of the data. So KNN was programmed from scratch for this project [38]. For KNN, imputation was conducted with the value of k set to two different values, 3 or 5. This meant that the imputation method either selected 3 or 5 individuals that were most similar (nearest) to perform imputation. Doing this provided some insight into how different parameters would perform in terms of classification accuracy. The distance measure used for KNN was Euclidean distance as the data values have an inherent distance built into them as discussed earlier.

MICE used to two different maximum iteration sizes, 10 and 50. This was done to ensure that the MICE regression model could converge and provide accurate imputation. However, MICE was difficult to implement for several reasons. The amount of time it needed to complete a run meant that a lot of time was required to run the algorithm. This was especially troublesome when the end was reached for all but a few chromosomes' due to the amount of RAM required not being available, so that the run crashed. As the imputation technique created regression models to impute the missing values, the results from the imputation technique had to be rounded later to ensure that the final imputed values were one of the allowed categorical values of 0, 1, or 2. Accomplishing this meant making a method to round values less than 0.5 to 0, values greater than and equal to 1.5 to 2 and then rounding values equal to or greater than 0.5 and less than 1.5 to 1 and can be seen in [39]. Also, due to MICE using a regression model to impute the values, an implication is made that there is a relationship between the features, so this method was not used to impute the missing values in the scaffold datasets, only the chromosome datasets as discussed earlier.

Despite producing datasets which were more accurate, MICE was not used for domain-based imputation changes due to being too time consuming, memory consuming and being a complicated method to implement domain knowledge in. When the runs were set to 50, two datasets, LG7 and LG8, could not be imputed due to the required resources being too great for 32GB of RAM. This also meant that SICE was not implemented in this project as it performs MICE multiple times to get the most frequent value. This would require significant power and memory resources that were inaccessible for this project.

B. Domain Imputation Methods

KNN was selected to apply domain knowledge to. It was ideal due to having a shorter computation time, requiring fewer resources and being more easily able to have domain knowledge applied to it. To implement domain knowledge into KNN a custom method was created to turn the feature positional information provided into a weighted function which was used to determine the most similar individuals to the individual being looked at. This function combines domain knowledge with machine learning knowledge. The function takes a feature that is currently being imputed and then determines the relative distance between the selected feature and all other features within a chromosome. Then to calculate the proper weighted value the distance value is inverted. This can be seen in (1):

$$Weight = 1 / distance \quad (1)$$

This allows for the features that are further away from the selected feature to have a less significant impact on whether

an individual is determined to be like another. This is based on the biological concept of crossover as genes that are closer together have a higher likelihood of staying together during crossover than genes that are further away. This means that each feature can have different individual fish selected as the most similar depending on the feature being imputed, as the weighted value for each feature changes depending on the selected feature. This helps accommodate crossover as features that are further away as less likely to be the same as the features closer to the feature due to crossover. The domain weight function can be seen in the code for the two domain versions of KNN [40], [41].

This domain-based weight function was implemented in two versions of KNN created for this project.

The first version created looked at individual fish and imputed all the missing values independently in that fish [40]. This meant that each fish and each feature within that fish was looked at separately. A downside to using this approach was that depending on the number of fish with no missing values there was potential for only a very limited amount of information to be used. If there were not more fish than the k being used for the number of nearest neighbours, then the method would look at all fish but only the genes with no missing values at all. This meant that for some datasets there was a significant amount of information that was not used due to there not being enough complete fish.

The second version of the domain based KNN approach was created to attempt to fix the lack of information that was an issue in the first version because of the significant number of features with >0 - <2.5% missing values, as shown in Fig. 1. The second version of the KNN approach used domain knowledge to weight the importance of the features using a waterfall or cascading approach [41]. The method starts by imputing features with the fewest number of missing values first. These features and instances are added to the available information that can be used by the KNN to predict the other missing values. The potential flaw with this approach is that errors could be propagated. This means that if a missing value is incorrectly imputed the likelihood that future imputed values are incorrectly imputed also increases. However, since the features being imputed to start with have so few missing values, that is less likely to happen than if the features with the most missing values were imputed first. If the features with the most missing values were imputed first, then the probability of errors occurring and propagating through the future imputations would be higher as there would be very little information available to be able to impute many missing values accurately. This version of the method provides the

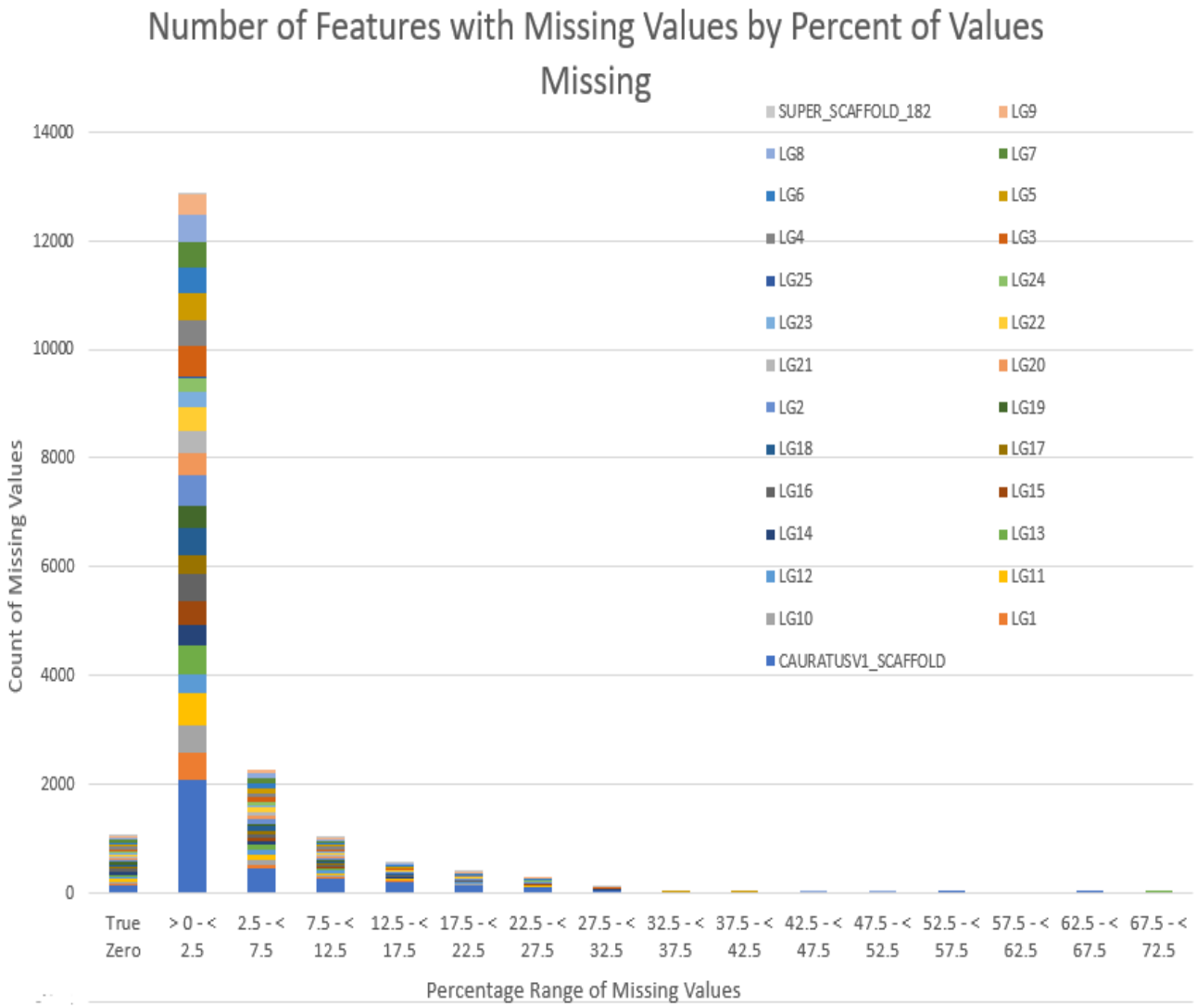


Fig. 1. Stacked Bar Chart Displaying the Average Accuracy of the Test Data when 30 different random states were used to split the data by split dataset

best potential for ensuring errors are less likely to be propagated and that there is more available information for imputing features with higher proportions of missing value.

The two Scaffold datasets could not be imputed using these methods. This is because the positional information provided for the features in those datasets is not actually position but relative position to each, because the chromosome those features belong to cannot be identified. This means that the complete datasets for the domain based approached do not have any of the scaffold dataset features added to them. Rather than impute the values using a different method and adding that into the combination a decision was made to keep the datasets purely made of one imputation type to ensure that the results would not be affected by using a combination of imputation methods.

V. Evaluation

A. *Classification Accuracy and Random Forest Statistical Significance*

The main goal of this project was to research whether the imputation performed on the dataset affected the classification accuracy, evaluation was dome by using the imputed datasets in a model and evaluating the classification accuracy. A Random Forest classifier was used as previous work on this over-arching project had shown that using a Random Forest classifier produced accuracy of around 65%.

The first test was conducted to see if the test data split influenced the results. This was done by applying a stratified test splitter on the imputed datasets 30 times with different random states set on the Stratified 10-fold Test Splitter and was implemented using the scikit library. The random state for the Random Forest classifier was fixed for the experiment. Fig. 2 shows the average test accuracy calculated from running this experiment on the complete datasets ordered from the most accurate to the least accurate. This shows that the MICE 50 method produced the most accurate results, of the methods tested, and performed better than the benchmark imputation method, the Most Frequent. Only one method in this experiment was outperformed by the baseline imputation method. This was Domain KNN 5. This could be due to 5 individuals being too many to be able to accurately select the value most likely to be the actual gene, which negatively affected the accuracy.

Several p-values were calculated using Friedman's Test. The first was a p-value for indicating whether there was a significant difference between the individual chromosome/scaffold datasets. This value was 4.2248E-05, indicating that there is a significant difference between the individual datasets and their classification accuracies due to the p-value being significantly smaller than 0.05. The second was a p-value indicating whether there was a difference between the individual chromosome/scaffold datasets and the complete datasets. This value was 0.00017206, indicating there was still a significant difference between the datasets, but the complete datasets decreased the overall difference between all the datasets. This decrease in difference between the imputation types caused by the complete datasets was

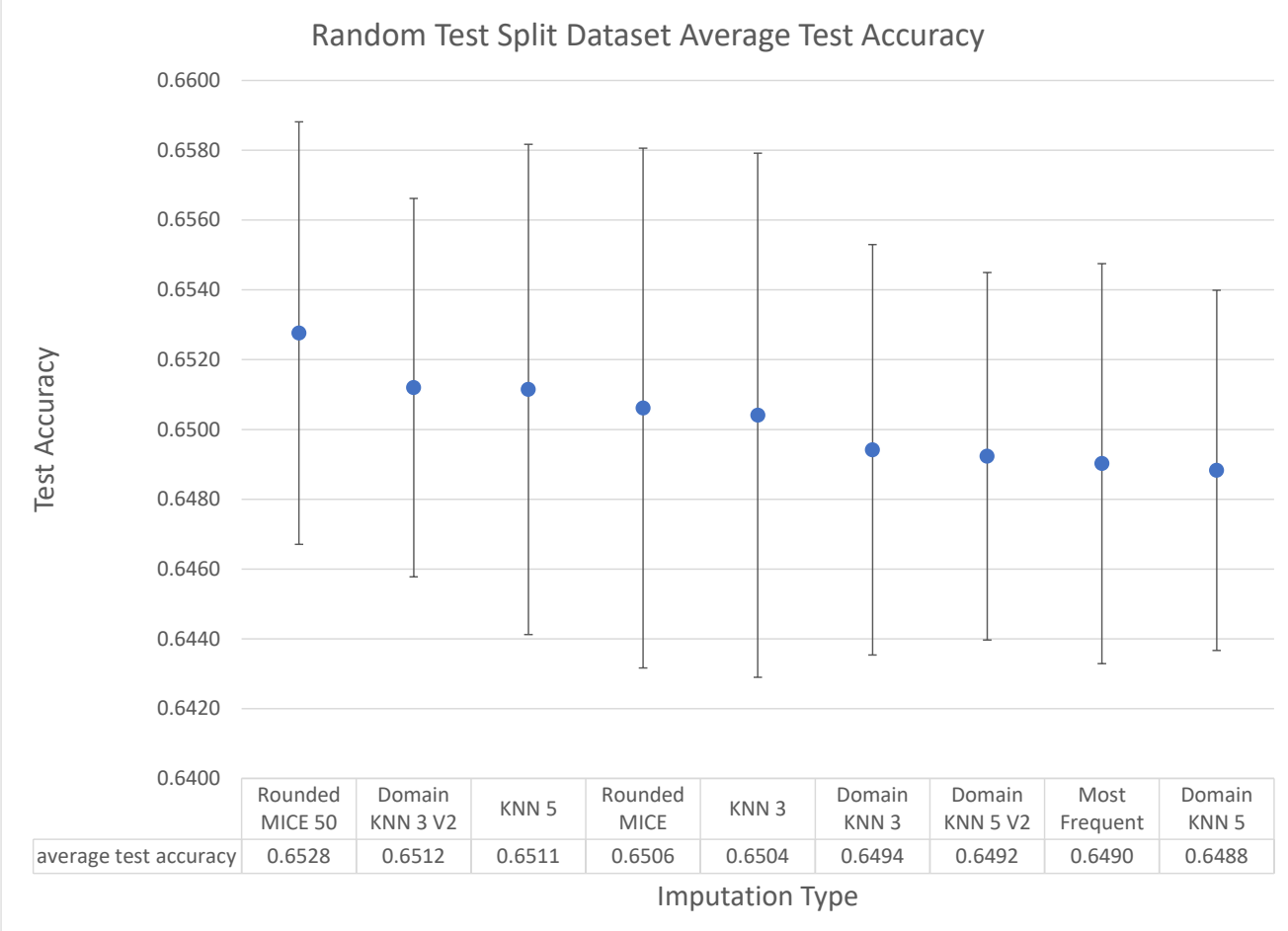


Fig. 2. Graph Displaying the Average Accuracy of the Test Data when 30 different random states were used to split the data

confirmed by not only the results shown in Fig. 2, due to the overlapping standard deviations and similar average accuracies for the complete datasets, but also by the p-value calculated for just the complete datasets, which was 0.43347012, indicating that there was no significant difference between the accuracy results of the complete datasets at all.

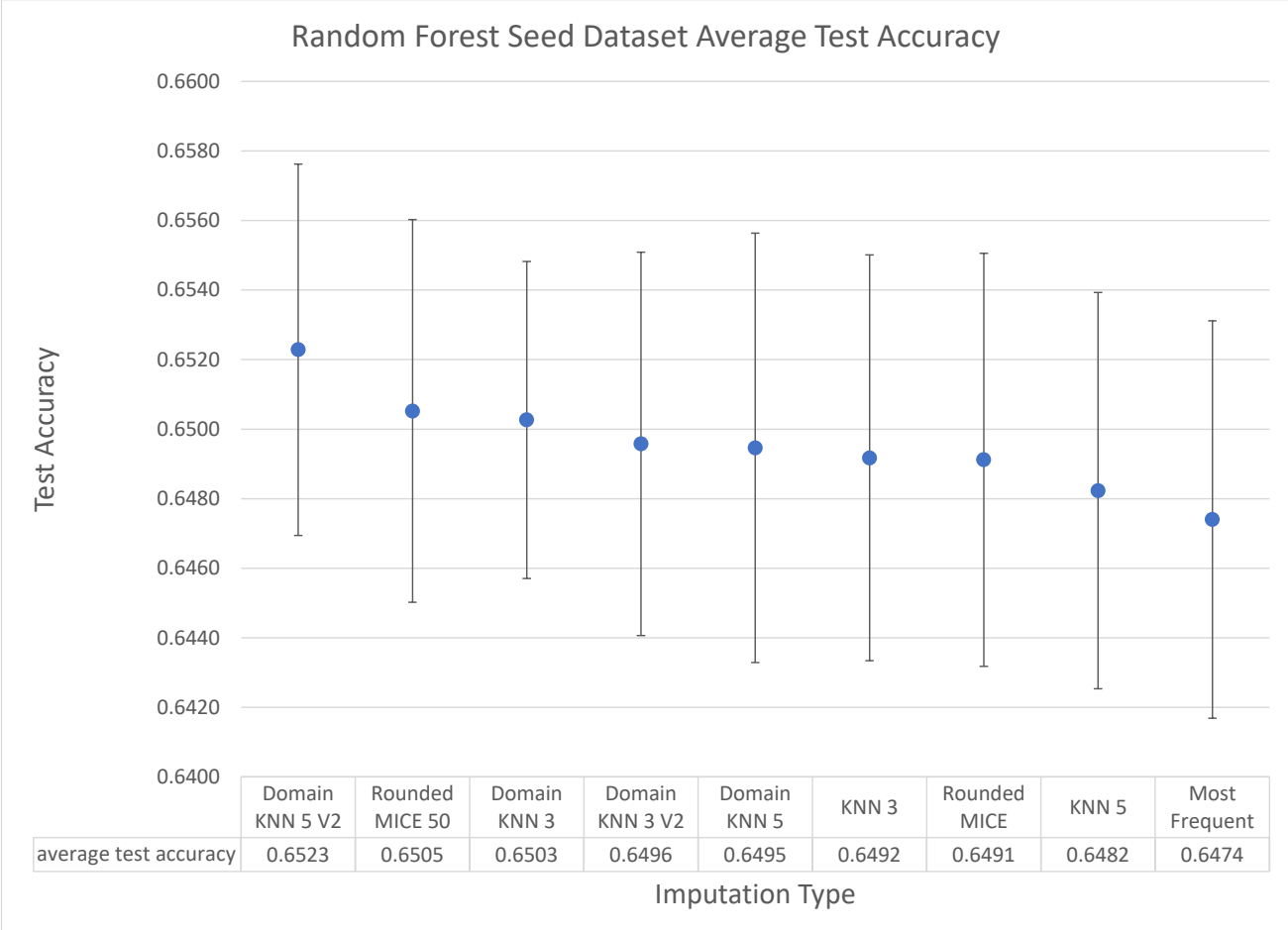
Overall, it was found the test split did not have a significant effect on the classification accuracy. This was shown in Fig. 2 by the overlapping standard deviations for each method and the Friedman's Test p-value for the combined datasets being greater than 0.05.

The second test was conducted to see whether the random state of the Random Forest model significantly affected the results of the imputation methods. This was conducted because random forests are known to be random state dependent. This was done by performing stratified 10-fold cross-validation on the imputed datasets 30 times with different random state sets for the Random Forest seed. The random state for the Stratified test splitter was fixed for the experiment.

Fig. 3 shows the average test accuracies calculated from running this experiment on the complete datasets ordered from the most accurate to the least accurate. This shows that the domain KNN 5 version 2 method produced the most accurate results out of the different imputation methods and performs better than the benchmark imputation method, the Most Frequent value. The experiment also indicated that no other imputation method was outperformed by the benchmark imputation method of the Most Frequent.

Several p-values were calculated using Friedman's Test. The first was a p-value for indicating whether there was a significant difference between the individual chromosome/scaffold datasets. This value was 0.001085608, indicating that there is a significant difference between the datasets and their classification accuracies due to being significantly smaller than 0.05. The second was a p-value indicating whether there was a difference between the individual chromosome/scaffold datasets including the complete datasets. This p-value was 0.003303644, indicating that there was still a significant difference between the datasets, but the complete datasets decreased the overall difference between all the datasets. This decrease in difference between the imputation types caused by the complete datasets was confirmed by the results shown in Fig. 3 due to the overlapping standard deviations, similar average accuracies for the complete datasets but also the p-value calculated for just the complete datasets which was 0.43347012, indicating that there was no significant difference between the results at all.

Overall, it was found that the Random Forest seed did not have a significant effect on the classification accuracy. This was Random Forest seed did not have a significant effect on the classification accuracy, which is shown in Fig. 3 by the overlapping standard deviation between the results, the small standard deviation of each imputation implementation and the Friedman's Test p-value for the combined datasets being



	Domain KNN 5 V2	Rounded MICE 50	Domain KNN 3	Domain KNN 3 V2	Domain KNN 5	KNN 3	Rounded MICE	KNN 5	Most Frequent
average test accuracy	0.6523	0.6505	0.6503	0.6496	0.6495	0.6492	0.6491	0.6482	0.6474

Fig. 3. Graph Displaying the Average Accuracy of the Test Data when 30 different random states were used to split the data

greater than 0.05 which indicated there was no significant difference between the accuracy results of the combined datasets.

Together these results indicate that despite having a statistically significant difference between the average accuracy of the chromosome/scaffold datasets, there is no statistical difference found between the different combined datasets' average accuracies. The results also show that the different random states for the test split produces more variable results than the different random forest seed. That is the p-values for the random test split are smaller than the p-values generated for the different random forest seeds. This indicates that the test split has more impact on the results than the random forest seed does and means that care is needed when splitting data into the training and test datasets for classification.

Table II shows the average rank of each imputation method based on how well it performed in each test. Overall, the domain-based approaches performed better than the benchmark imputation method of the Most Frequent Value, which has an average rank of 8.5, indicating that it produced the least accurate classifiers overall. MICE with 50 iterations performed the best out of all the different imputation methods. However, MICE is not the recommended imputation method for this data as it is not resource-effective when compared to the other methods. Either of the domain KNN version 2 approaches would be more suitable due to their rank being lower than the Most Frequent and general machine learning imputation methods, excluding MICE 50.

B. Feature Analysis

As the results of the complete datasets were similar, a test was conducted with a fixed random state for both the test split and the Random Forest seed. This meant that the results of the features selected by the Random Forest classifier could be compared.

To examine in more depth the features chosen by the random forest classifier, the top 200 features selected in the random forest trees based on the complete datasets were recorded and compared to identify any overlaps in the features selected. A breakdown of the features used and whether those features contained imputed values was also completed. As each complete dataset had a random forest tree generated that made a total of 9 tree and a potential for there to be 1800 unique features selected since 200 features were selected from each tree.

Table III shows that despite the fact that 1800 features were recorded in total from the 9 trees, 200 from each dataset, there were only 956 unique features used by the Random Forest classifiers. Of the features selected by the Random Forest classifier, 891 had imputed values, with only 65 features having complete data. This is not surprising because, as shown in Fig 1, there are significantly more features with missing values in them than features with no missing values. Due to the number of features, this means that features with missing values are more likely to be selected than features without missing values. However, what is surprising is that most of the complete features used only occur in one tree's top 200 features, shown in Table III, whereas 18 of them are implemented in multiple trees' top 200 used features.

Despite the fact the results from the Friedman's Test experiments showed that there were no significant differences, the majority of the features in the top 200 for each method are independent of one another, indicated by 761 of 956 features only occurring in one tree, indicated in Table III. This could potentially indicate that the features which occur in multiple imputation methods' top 200 features could signal they may be good indicators for growth rate classification and explain why the results are so similar. If all the trees pick the same base features to build Random Forest trees from, then selecting similar features early on at the start of the tree would help explain why the results end up being so similar as having the same starting features would mean that small differences

TABLE III. TABLE OF THE AVERAGE ACCURACIES OF THE TWO TESTS AND THE AVERAGE PERFORMANCE RANK

Imputation Type	Test Split Average Test Accuracy	Random Forest Average Test Accuracy	Average Rank
Most Frequent	0.649022823	0.647403443	8.5
KNN 3	0.650407464	0.649176577	5.5
KNN 5	0.65114509	0.649461717	5.5
Rounded MICE	0.650615302	0.649119096	5.5
Rounded MICE 50	0.652763054	0.650522143	1.5
Domain KNN 3	0.649416412	0.650265319	4.5
Domain KNN 5	0.648829143	0.649461717	7
Domain KNN 3 V2	0.651200306	0.649576962	3
Domain KNN 5 V2	0.649232359	0.652283384	4

TABLE II. TABLE OF THE SIMILARITIES BETWEEN THE TOP 200 FEATURES IN RANDOM FOREST TREES

Number of Trees	Has Imputed Values	No Imputed Values	Grand Total
9	5	0	5
8	16	2	18
7	31	2	33
6	28	5	33
5	35	2	37
4	33	3	36
3	16	1	17
2	13	3	16
1	714	47	761
Grand Total	891	65	956

between features selected would occur later in the tree and produce less of an impact on the classification accuracy than the features selected at the start of the tree.

The main issue appears to be the number of features in this dataset with very few instances in comparison. There are around 1100 instances with 18543 features indicated in Table I. This means it is a high dimensionality dataset and so implementing feature selection on the different imputed datasets could produce results where the imputation methods matter based on selected features. However, investigating that was outside the scope of this project.

VI. Conclusions and Future Work

A. Conclusions

Overall, five different data imputation techniques were implemented to analyse whether machine learning imputation techniques had an effect on the classification accuracy of models for classifying the growth rate of Snapper fish. This was done to help future research being conducted in this area with this data. Two of the methods used in this paper have been created specifically for this project taking into account specific domain knowledge.

Despite the results showing that there is no statistically significant difference in the accuracy results generated, a recommendation would be to use a domain-based machine learning method, either domain KNN or domain KNN version 2. These methods produced better results than more general imputation methods, except for MICE 50. However, MICE 50 is not recommended due to the time it takes for the imputation to complete and the resources it requires. If the risk of propagating errors using domain KNN version 2 is not acceptable then domain KNN should be used. However due to how domain KNN version 2 is implemented it is less likely to propagate errors due to the order of the features being imputed.

Despite machine learning imputation techniques not producing much difference in classification accuracy, this may not be true using biology imputation techniques. These need to be conducted by experts in genomics and are currently being done by researchers at the New Zealand Institute for Plant and Food Research Limited. Datasets generated using pure biological imputation techniques were not able to be generated in time to be analysed for this project so there is no information about how accurate the results generated may be. However, based on the results indicating that the domain-based machine learning imputation techniques produced more accurate results, there is a high likelihood that the biological imputation would generate more accurate results.

B. Future Work

Two different ideas would have been implemented in this project if there had been more time and would be worth investigating.

The first is implementing feature selection using the different imputation methods and analysing differences between the selected features for each imputation method. Others in the over-arching project have implemented feature selection using allele frequency. Allele frequency calculates the relative frequency of alleles in specific genes [42]. Feature selection occurs by comparing the allele frequencies between different genes, and genes with the same allele frequency are considered identical, so only one gene stays in the final dataset. The reduced number of features to be considered by the classification model may improve the results.

The second is to include the "well" feature in the original dataset as a stand-in for the environment. The idea of the project is to create a model which can predict fish growth rates regardless of their living environment. However, studies indicate that the growth rate of fish is influenced and affected by the environment [43]–[45]. As the fish were grown in partially controlled tanks, with non-heated seawater brought into them, the lack of environmental features within the data could explain why the accuracy of the different types of imputation produced little difference in the overall accuracy of the model. Testing datasets including an environmental variable to compare their accuracy could be enlightening as current datasets are dependent solely on biology, despite the growth rate of fish not being dependent solely on biology [43]–[45]. The environment a fish lives in is a significant factor that influences how that fish grows. Fish with good growth genes may not grow quickly in some environments. A further idea, beyond using the well as a stand-in for the environment, would be to access the recorded temperature data and include it in the model to see if that improved the classification accuracy. As this data is currently not accessible, more time will be needed to complete this analysis.

A future ENGR489 project could be an investigation and analysis of the available environmental data and how it affects classification accuracy. This topic would be unique for future research as it is currently unknown what results would come from doing this, and it combines biology with machine learning and data analysis.

Acknowledgment

I thank Julie Blommaert, Linley Jesson, Chris Van Houtte and Maren Wellenreuther from The New Zealand Institute for Plant and Food Research Limited for their invaluable help with this project by answering domain-specific questions.

I thank Ze Chan, Jessie Dong and Xingsi Xue for giving me advice, listening to ideas, and providing code as a base to start the project.

I thank Yi Mei for being an invaluable supervisor and providing much-needed advice, guidance, and feedback during this project.

References

[1] D. T. Ashton, P. A. Ritchie, and M. Wellenreuther, 'High-Density Linkage Map and QTLs for Growth in Snapper (Chrysophrys auratus)', *G3 GenesGenomesGenetics*, vol. 9, no. 4, pp. 1027–1035, Apr. 2019, doi: 10.1534/g3.118.200905.

[2] 'Snapper: Adult', NIWA. Accessed: May 28, 2023. [Online]. Available: https://niwa.co.nz/fisheries/ecosystem-influences-on-snapper/life-cycle/adult

[3] 'Snapper – New Zealand Sport Fishing Council', New Zealand Sport Fishing. Accessed: May 28, 2023. [Online]. Available: https://www.nzsportfishing.co.nz/fisheries/species/snapper/

[4] Z. Chen, 'Machine Learning Techniques for Fish Breeding : Finding Genetic Variants Related to Growth', *unpublished*, Accessed: Oct. 15, 2023. [Online]. Available: https://vuw-my.sharepoint.com/personal/meiyi_staff_vuw_ac_nz/_layouts/15/onedrive.aspx?csf=1&web=1&e=164FBg&OR=Teams%2DHL&CT=1684818739694&clickparams=eyJBcHBOYW1lIjoiVGVhbXMtRGVza3RvcCIsIkFwcFZlcnNpb24iOiIxNDE1LzIzMDQwMjAyNzA1IiwiSGFzRmVkZXJhdGVkVXNlciI6dHJ1ZX0%3D&cid=113cc70e%2Db405%2D44c2%2Db475%2Db008fbf6886f&FolderCTID=0x0120004F62EAF011AF8F488DE069F402E87C58&id=%2Fpersonal%2Fmeiyi%5Fstaff%5Fvuw%5Fac%5Fnz%2FDocuments%2FData%2DScience%2Dfor%2DGenomic%2DFish%2DBreeding%2DPFR%2DVUW%2FFishBreeding%2FZeChen%2FReport%2FSu

mmer%5FResearch%2Epdf&parent=%2Fpersonal%2Fmeiyi%5Fstaff%5Fvuw%5Fac%5Fnz%2FDocuments%2FData%2DScience%2Dfor%2DGenomic%2DFish%2DBreeding%2DPFR%2DVUW%2FFishBreeding%2FZeChen%2FReport
[5] 'Goal 2 | Department of Economic and Social Affairs'. Accessed: Sep. 19, 2023. [Online]. Available: https://sdgs.un.org/goals/goal2
[6] 'Goal 14 | Department of Economic and Social Affairs'. Accessed: Sep. 19, 2023. [Online]. Available: https://sdgs.un.org/goals/goal14
[7] 'Linkage', Genome.gov. Accessed: Oct. 15, 2023. [Online]. Available: https://www.genome.gov/genetics-glossary/Linkage
[8] A. Minelli, A. N. Tassetti, B. Hutton, G. N. Pezzuti Cozzolino, T. Jarvis, and G. Fabi, 'Semi-Automated Data Processing and Semi-Supervised Machine Learning for the Detection and Classification of Water-Column Fish Schools and Gas Seeps with a Multibeam Echosounder', *Sensors*, vol. 21, no. 9, Art. no. 9, Jan. 2021, doi: 10.3390/s21092999.
[9] Y. Baidai, L. Dagorn, M. J. Amande, D. Gaertner, and M. Capello, 'Machine learning for characterizing tropical tuna aggregations under Drifting Fish Aggregating Devices (DFADs) from commercial echosounder buoys data', *Fish. Res.*, vol. 229, p. 105613, Sep. 2020, doi: 10.1016/j.fishres.2020.105613.
[10] F. Monteiro *et al.*, 'Classification of Fish Species Using Multispectral Data from a Low-Cost Camera and Machine Learning', *Remote Sens.*, vol. 15, no. 16, Art. no. 16, Jan. 2023, doi: 10.3390/rs15163952.
[11] L. Ren *et al.*, 'Rapid identification of fish species by laser-induced breakdown spectroscopy and Raman spectroscopy coupled with machine learning methods', *Food Chem.*, vol. 400, p. 134043, Jan. 2023, doi: 10.1016/j.foodchem.2022.134043.
[12] D. V. Notte, R. J. Lennox, D. C. Hardie, and G. T. Crossin, 'Application of machine learning and acoustic predation tags to classify migration fate of Atlantic salmon smolts', *Oecologia*, vol. 198, no. 3, pp. 605–618, Mar. 2022, doi: 10.1007/s00442-022-05138-3.
[13] S. Benzer, F. H. Garabaghi, R. Benzer, and H. D. Mehr, 'Investigation of some machine learning algorithms in fish age classification', *Fish. Res.*, vol. 245, p. 106151, Jan. 2022, doi: 10.1016/j.fishres.2021.106151.
[14] N. Bravata, D. Kelly, J. Eickholt, J. Bryan, S. Miehls, and D. Zielinski, 'Applications of deep convolutional neural networks to predict length, circumference, and weight from mostly dewatered images of fish', *Ecol. Evol.*, vol. 10, no. 17, pp. 9313–9325, 2020, doi: 10.1002/ece3.6618.
[15] F. Wei, K. Ito, K. Sakata, T. Asakura, Y. Date, and J. Kikuchi, 'Fish ecotyping based on machine learning and inferred network analysis of chemical and physical properties', *Sci. Rep.*, vol. 11, no. 1, Art. no. 1, Feb. 2021, doi: 10.1038/s41598-021-83194-0.
[16] A. Valdivieso, D. Anastasiadi, L. Ribas, and F. Piferrer, 'Development of epigenetic biomarkers for the identification of sex and thermal stress in fish using DNA methylation analysis and machine learning procedures', *Mol. Ecol. Resour.*, vol. 23, no. 2, pp. 453–470, 2023, doi: 10.1111/1755-0998.13725.
[17] R. Gutha, S. Yarrappagaari, L. Thopireddy, K. S. Reddy, and R. R. Saddala, 'Effect of abiotic and biotic stress factors analysis using machine learning methods in zebrafish', *Comp. Biochem. Physiol. Part D Genomics Proteomics*, vol. 25, pp. 62–72, Mar. 2018, doi: 10.1016/j.cbd.2017.10.005.
[18] J.-H. Hu, W.-P. Tsai, S.-T. Cheng, and F.-J. Chang, 'Explore the relationship between fish community and environmental factors by machine learning techniques', *Environ. Res.*, vol. 184, p. 109262, May 2020, doi: 10.1016/j.envres.2020.109262.
[19] J.-F. Flot, H. Marie-Nelly, and R. Koszul, 'Contact genomics: scaffolding and phasing (meta)genomes using chromosome 3D physical signatures', *FEBS Lett.*, vol. 589, no. 20PartA, pp. 2966–2974, 2015, doi: 10.1016/j.febslet.2015.04.034.
[20] 'Haplotype', Genome.gov. Accessed: Jun. 01, 2023. [Online]. Available: https://www.genome.gov/genetics-glossary/haplotype
[21] J. C. Roach *et al.*, 'Chromosomal Haplotypes by Genetic Phasing of Human Families', *Am. J. Hum. Genet.*, vol. 89, no. 3, pp. 382–397, Sep. 2011, doi: 10.1016/j.ajhg.2011.07.023.
[22] Y. L. Qiu, H. Zheng, and O. Gevaert, 'Genomic data imputation with variational auto-encoders', *GigaScience*, vol. 9, no. 8, p. giaa082, Aug. 2020, doi: 10.1093/gigascience/giaa082.
[23] C. Vasilopoulou, B. Wingfield, A. P. Morris, and W. Duddy, 'snpQT: flexible, reproducible, and comprehensive quality control and imputation of genomic data'. F1000Research, Jul. 14, 2021. doi: 10.12688/f1000research.53821.1.
[24] R. Antolín, C. Nettelblad, G. Gorjanc, D. Money, and J. M. Hickey, 'A hybrid method for the imputation of genomic data in livestock populations', *Genet. Sel. Evol.*, vol. 49, no. 1, p. 30, Mar. 2017, doi: 10.1186/s12711-017-0300-y.
[25] A. Klímová, E. Kašná, K. Machová, M. Brzáková, J. Přibyl, and L. Vostrý, 'The use of genomic data and imputation methods in dairy cattle breeding', *Czech J. Anim. Sci.*, vol. 65, no. 12, pp. 445–453, Dec. 2020, doi: 10.17221/83/2020-CJAS.
[26] S. I. Khan and A. S. M. L. Hoque, 'SICE: an improved missing data imputation technique', *J. Big Data*, vol. 7, no. 1, p. 37, Jun. 2020, doi: 10.1186/s40537-020-00313-w.
[27] S. Faisal and G. Tutz, 'Multiple imputation using nearest neighbor methods', *Inf. Sci.*, vol. 570, pp. 500–516, Sep. 2021, doi: 10.1016/j.ins.2021.04.009.
[28] A. Dubey and A. Rasool, 'Efficient technique of microarray missing data imputation using clustering and weighted nearest neighbour', *Sci. Rep.*, vol. 11, no. 1, Art. no. 1, Dec. 2021, doi: 10.1038/s41598-021-03438-x.
[29] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, 'Multiple imputation by chained equations: what is it and how does it work?', *Int. J. Methods Psychiatr. Res.*, vol. 20, no. 1, pp. 40–49, Mar. 2011, doi: 10.1002/mpr.329.
[30] A. W.-C. Liew, N.-F. Law, and H. Yan, 'Missing value imputation for gene expression data: computational techniques to recover missing data from available information', *Brief. Bioinform.*, vol. 12, no. 5, pp. 498–513, Sep. 2011, doi: 10.1093/bib/bbq080.
[31] W.-C. Lin and C.-F. Tsai, 'Missing value imputation: a review and analysis of the literature (2006–2017)', *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 1487–1509, Feb. 2020, doi: 10.1007/s10462-019-09709-4.
[32] D. G. Pereira, A. Afonso, and F. M. Medeiros, 'Overview of Friedman's Test and Post-hoc Analysis', *Commun. Stat. - Simul. Comput.*, vol. 44, no. 10, pp. 2636–2653, Nov. 2015, doi: 10.1080/03610918.2014.931971.
[33] 'scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation', scikit-learn. Accessed: May 29, 2023. [Online]. Available: https://scikit-learn.org/stable/
[34] '6.4. Imputation of missing values', scikit-learn. Accessed: May 29, 2023. [Online]. Available: https://scikit-learn/stable/modules/impute.html
[35] '3.3. Metrics and scoring: quantifying the quality of predictions', scikit-learn. Accessed: May 30, 2023. [Online]. Available: https://scikit-learn/stable/modules/model_evaluation.html
[36] R. Taylor, 'Cross_fold_Baseline_RF'. Oct. 04, 2023. Accessed: Oct. 15, 2023. [Online]. Available: https://gitlab.ecs.vuw.ac.nz/course-work/project489/2023/taylorrose3/engr489project/-/blob/main/Cross_fold_Baseline_RF.py?ref_type=heads
[37] R. Taylor, 'Baseline_RF_Results'. Oct. 04, 2023. Accessed: Oct. 15, 2023. [Online]. Available: https://gitlab.ecs.vuw.ac.nz/course-work/project489/2023/taylorrose3/engr489project/-/blob/main/Baseline_RF_Results.py?ref_type=heads
[38] R. Taylor, 'KNNMostFrequent'. Jun. 02, 2023. Accessed: Oct. 15, 2023. [Online]. Available: https://gitlab.ecs.vuw.ac.nz/course-work/project489/2023/taylorrose3/engr489project/-/blob/main/KNNMostFrequent.py
[39] R. Taylor, 'rounding'. Sep. 17, 2023. Accessed: Oct. 15, 2023. [Online]. Available: https://gitlab.ecs.vuw.ac.nz/course-work/project489/2023/taylorrose3/engr489project/-/blob/main/rounding.py?ref_type=heads
[40] R. Taylor, 'KNNPositionBased'. Aug. 04, 2023. Accessed: Oct. 15, 2023. [Online]. Available: https://gitlab.ecs.vuw.ac.nz/course-work/project489/2023/taylorrose3/engr489project/-/blob/main/KNNPositionBased.py?ref_type=heads
[41] R. Taylor, 'KNNPositionBasedVersion2'. Aug. 15, 2023. Accessed: Oct. 15, 2023. [Online]. Available: https://gitlab.ecs.vuw.ac.nz/course-work/project489/2023/taylorrose3/engr489project/-/blob/main/KNNPositionBasedVersion2.py?ref_type=heads
[42] N. Rezaei and M. Hedayat, 'Allele Frequency', in *Brenner's Encyclopedia of Genetics (Second Edition)*, S. Maloy and K. Hughes, Eds., San Diego: Academic Press, 2013, pp. 77–78. doi: 10.1016/B978-0-12-374984-0.00032-2.
[43] L. F. Canosa and J. I. Bertucci, 'The effect of environmental stressors on growth in fish and its endocrine control', *Front. Endocrinol. Lausanne*, vol. 14, pp. 1109461–1109461, 2023, doi: 10.3389/fendo.2023.1109461.
[44] T. Flikac, D. G. Cook, W. Davison, and A. Jerrett, 'Seasonal growth dynamics and maximum potential growth rates of Australasian snapper (Chrysophrys auratus) and yellow-eyed mullet (Aldrichetta

forsteri)', *Aquac. Rep.*, vol. 17, p. 100306, Jul. 2020, doi: 10.1016/j.aqrep.2020.100306.

[45] D. Parsons *et al.*, 'Snapper (Chrysophrys auratus): a review of life history and key vulnerabilities in New Zealand', *N. Z. J. Mar. Freshw. Res.*, vol. 48, no. 2, pp. 256–283, Apr. 2014, doi: 10.1080/00288330.2014.892013.