Binary Classification of Clonal Haematopoiesis of Indeterminate Potential

ADS2001: Data Challenges 3

Macey Jackson, Nick Phan, Poornisha Senthil Kumar, Sam Cropman, Wei Jie Huang

Table of Contents

EXECUTIVE SUMMARY

Hematopoietic stem cells produced in human bone marrow specifically give rise to other types of blood cells throughout the lifetime of an organism; this production of blood cells is termed hematopoiesis. CHIP (**C**lonal **H**ematopoiesis of **I**ndeterminate **P**otential) is a condition describing the development of detectable clones with somatic mutations in hematopoietic cells, leading to clonal expansion. Although these somatic mutations are commonly acquired during human aging, the prevalence of CHIP increases the incidence of cardiovascular disease and blood cancers, such as leukaemia and myelodysplasia, in otherwise normal, healthy people. The aim of the project is to classify if patients, denoted by a unique identifier (*MSID*), are either CHIP positive or negative (denoted as 'control'), within supplied datasets.

In addition, there exists two sets of DNA sequencing data, both originating from the same source of blood samples, but read over two different exome sequencing machines. This resulted in varying levels of false negative reads, where Control classifications of patients could be defined as CHIP. Consequently, of the sub-sampled eighty-eight participants (44 CHIP, 44 Control), there exists a slight class imbalance of more CHIP classifications than depicted; a sub goal of the project is to identify which patients were falsely misclassified incorrectly as Control through machine learning and modelling.

By applying SKLearn models to the dataset after preprocessing, it's uncovered that KNearestNeighbour, AdaBoost and Decision Tree classifiers perform the best, scoring accuracies of 81.5%, 80.7% and 78.8% respectively when trained on the first dataset, and 63.0%, 68.9%, 66.1% respectively when trained on the second dataset. Examining the feature importance attributes of AdaBoost and Decision Tree classifiers, both placed almost identical weightings on variable importance; the most notable features include *DP, HIAF, REFBIAS*

and *ODDRATIO*, the two former variables relating to the DNA sequencing read itself, the latter indicating the quality and reliability of the read.

The accuracy differences led to the conclusion that the second dataset contained a greater amount of sequencing noise than the first dataset; this statement is supported by further data analysis of SHAP scores. We utilise SHAP values (average of the marginal contributions across all permutations), to demonstrate how much each predictor contributes, either positively or negatively, to the target variable. SHAP values are primarily used to increase the model's transparency and to indicate a feature's impact on the model prediction. Analysing SHAP values with *QUAL* as a primary variable, it's concluded that a larger variance in *QUAL* arises, resulting in lower accuracy in model predictions.

In addition, by examining the trends and weighted averages of model inaccuracies when grouped by participant's *MSID*, a set of patients had been continuously misclassified by the binary classification models. This led to the conclusion that particular patients had been misclassified across both datasets. When the models ran again with these labels flipped, the accuracy of models had increased by a range of 1-5%.

INTRODUCTION

The Binary Classification of Clonal Haematopoiesis of Indeterminate Potential project incorporated numerous domains of data science as well as bioinformatics. It was apparent that machine learning would be required to address the aim with data exploration and cleaning in conjunction. As the data consisted of numerous categorical and continuous features, comprising many bioinformatic domain specific features, knowledge of the meaning and impact of these were required.

The ASPREE (ASPirin in Reducing Events in the Elderly) trial was conducted to look at a cohort of healthy elderly and to observe the long-term effects of low dose aspirin. led by

Monash University in Australia and the Berman Centre for Outcomes and Clinical Research Institute in the USA. The study findings were published in the New England Journal of Medicine, September 2018, although further study findings are still being prepared for publication. There were over 12,000 participants for which a blood sample was taken. For approximately 2000 participants, a list sensitive method to detect clonal haematopoiesis was performed. This data was collected using sequencing technology (NovaSeq) and was supplied in TSV files after general preprocessing.

‘Clonal Haematopoiesis of indeterminate potential is an aging-related phenomenon in which hematopoietic stem cells contribute to the formation of a genetically distinct subpopulation of blood cells’ *Gondek LP, DeZern AE (2020).* Haematopoiesis encapsulates the formation, development and differentiation of blood cells. However, in the aging-process, this leads to certain biological repercussions when the formation of genetically distinct blood cells arsies. These are identified by mutated nucleotides on strands of DNA, compared to the reference genome, hence why the individuals had their blood sampled again three years after their ‘baseline’ sample was taken.

This project asked whether the prevalence of chip changes over the course of three years, evidenced by both samples from each participant. In order to address this, the data supplied featured numerous variables that can be explored to determine the appearance of CHIP in a sample. Scientific research coincided with what was expressed in the data, with certain genes expressing mutations signifying the beginning of clonal haematopoiesis. These indicators were explored through data analysis and machine learning, to identify which features alluded to an individual being ‘CHIP positive’.

The dataset was derived from a variant calling software and was annotated with features associated with the variants. These features included, but were not limited to;

- Participant ID – *MSID*
- Allele Frequency - *AF*
- High Quality Allele Frequency (*HIAF*)
- Reference Bias – *REFBIAS*
- Variant Bias - *VARBIAS*
- Genomic Location – *loci*
- Sample Classification - *chipOrControl*
- Sample Timepoint
- Gene Name (*SYMBOL*) – *Gene* associated variant

Through online research, it was evident that there were particular genes that began to express variants signalling the beginning of the clonal haematopoiesis. These genes were featured in the many unique *SYMBOL's* in the dataset. Further, from the definitions, *HIAF* was considered a better indicator than *AF*, pertaining to a High Quality Allele Frequency read. *REFBIAS* and *VARBIAS* were seemingly related, as they represented the strand's bias, whether it was drastically 'mutated' or not.

As there was inherent sequencing noise in the dataset, it was apparent that this had to be limited to achieve accurate machine learning results. To account for and remove the sequencing noise from the dataset, exploratory analysis would be necessary to find features that indicated CHIP. Furthermore, individual samples were labelled incorrectly, with some features of a sample suggesting a CHIP positive result, despite being labelled as control. This noise and labelling error needed to be dealt with through the aim of the project, as achieving a model that could predict CHIP in a sample is heavily affected by such obstacles. This was addressed via changing the binary classifications of labels, with the grouping of patient *MSID*'s. Further, through model comparisons, by looking at the model inaccuracies, and addressing them, small improvements in accuracy were achieved.

DATA QUALITY

DNA sequencing reads of patients are collected via whole exome sequencing, drawn from a blood sample of a patient. Hence, data collection is affordable, thus resulting in over one million reads of sequencing data for a subsample of 88 patients who participated in Monash's

ASPREE study. However, although many data points were collected, many were unreliable to work with. In particular, many duplicate rows of data existed, due to many sequencing reads of the same blood sample in identically targeted areas. By dropping duplicate rows, the number of reads within a dataset drops approximately from one million to twelve-hundred thousand. Diving deeper into the dataset itself, many variables, such as *QUAL*, *BIAS*, *REFBIAS*/*VARBIAS* and *ODDRATIO* provide an indication of whether a DNA sequencing read is reliable or not.

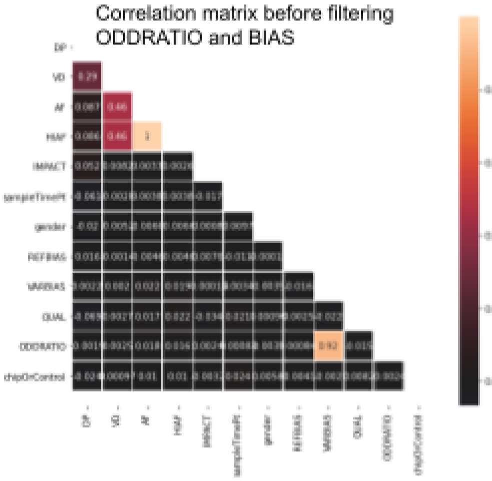
QUAL, referred to as quality score throughout, is an indication of the quality of a read, it represents the probability of a read being correct. When analysing quality scores, the higher the quality score was, the higher the probability of the read being correct. Similarly, the lower quality score indicates a higher probability that the read is incorrect, so we began analysing the scores to determine the quality of the reads in our data. ("Phred-scaled quality scores", GATK Team, 19/5/21). The formula to find the accuracy (A) given the quality score (Q) is:

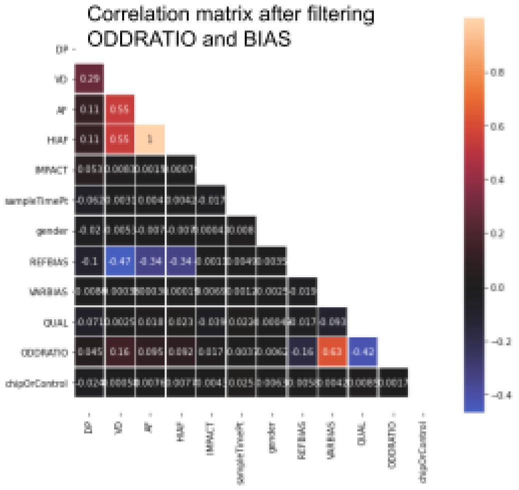
$$A = 1 - 10\text{^}(- (Q/10))$$

Using this formula, we found that the accuracies across both datasets ranged from approximately 99.9% to 99.99% which is excellent. With further investigation into the quality values, particularly looking at differences between the datasets, we found that the mean quality score for dataset 1 was greater than that of dataset 2; 35.4 and 32.1 respectively with standard deviations 2.68 and 4.48. From this observation we determined that dataset 2 contained greater levels of sequencing noise than that of dataset 1, due to higher variability.

BIAS was another variable which was indicative of the quality of the data. The *BIAS* value was given to us as a ratio of *REFBIAS* : *VARBIAS* or reference : variant, the values of the *BIAS* ratio were either 0, 1 or 2. 0 indicates that bias could not be determined, 1 means only one orientation is observed and 2 means both orientations are observed. To further explain, *REFBIAS*, which is also a ratio, counts the number of reads from each strand that matches the reference and we would expect an equal number of reads to each DNA strand, that is strand1:strand2. This is the same for *VARBIAS*, but it counts the number of reads for each strand that matches the variant instead of reference. In summary the ratios indicate how many reads were different, the closer the ratio is the less variation there was. The *BIAS* value indicates whether strand bias is present, strand bias having a negative impact on the quality of the data. If the bias ratio is 2:1 or 1:2 this indicates strand bias and if we have a 2:2 or 1:1 ratio this suggests no strand bias. If there is a 0 in the ratio then bias could not be determined for either reference or variant so the presence of strand bias cannot be determined.

Further analysis brought to attention the correlation between *ODDRATIO* and the bias variables, the odd ratio indicating the confidence of the variant call. We found that the *ODDRATIO* was strongly correlated to the variant bias as per unfiltered correlation matrix below. This encouraged more analysis into *ODDRATIO* and we found that an *ODDRATIO* of 0 resulted in a *BIAS* ratio of 2:1 or 1:2, these ratios being indicative of strand bias. When removing *ODDRATIO* values of 0 we observed that almost all strand bias was removed from the data and correlation between *ODDRATIO* and *BIAS* along with *REFBIAS* increased as seen below.


Correlation matrix before filtering ODDRATIO and BIAS


Correlation matrix after filtering ODDRATIO and BIAS

Another variable which could potentially be reducing the quality of the data we are working with is the correlation between specific *Genes* and the variant being identified as CHIP positive. Within the data there were 5 *Genes* which were found to have a high correlation to CHIP, that is a large percent of CHIP positive variant reads were found to have one of five *Genes* associated with the variant. Because of this strong correlation we chose to use only these five genes' data in our models to improve accuracy, although this did have the desired impact on our accuracies there is the possibility that it reduced the quality of our data. This is because these five genes were over sampled in the dataset because they have been specifically targeted in DNA sequencing for CHIP. This over-sampling in order to focus on these specific *Genes* could have potentially caused some imbalance within the dataset.

LABEL NOISE

As a reminder, within both datasets, grouping the eighty-eight participants by their unique *MSID* demonstrates no inconsistencies between dataset 1 and dataset 2 classifications (44 CHIP, 44 Control). In addition, within each individual dataset, each read of a particular patient presents the same label; thus for every unique *MSID,* no inconsistent classification arises. From these two observations, it's hypothesised that the label noise is large enough to span both datasets, hence, we cannot ascertain that a particular machine produced a ground truth classification.

There were two methods in uncovering label noise, the first was utilising a Python package, cleanlab, which wraps around sklearn machine learning models working with label noise probabilities as an input; the second consisted of further exploratory analysis of modelling accuracies and averages, inferring structured bias across multiple patients through *MSID* groupings, which will be discussed further in results.

As described in its documentation, cleanlab is a framework for confident learning, analogous to how PyTorch and TensorFlow are frameworks for deep learning. Thus, the purpose of the library is to find label errors in datasets, and aid machine learning models with noisy labels in mind. Integrating the package was straightforward, as confident learning required no hyperparameter tuning. Through cross validation to obtain predicted probabilities out-of-sample, it can directly estimate the joint distribution of noisy and true labels, thus output a noise matrix representing the predicted probabilities for a classification. Furthermore, the calculated probabilities for each element in the noise matrix does not assume randomly uniform label noise within the implementation, hence increasing the likelihood of detecting structured bias across *MSID* groups. By utilising cleanlab within both CHIP datasets, the expectation is that the package's methods will improve the accuracy scores of our binary classification models.

MODEL DEVELOPMENT

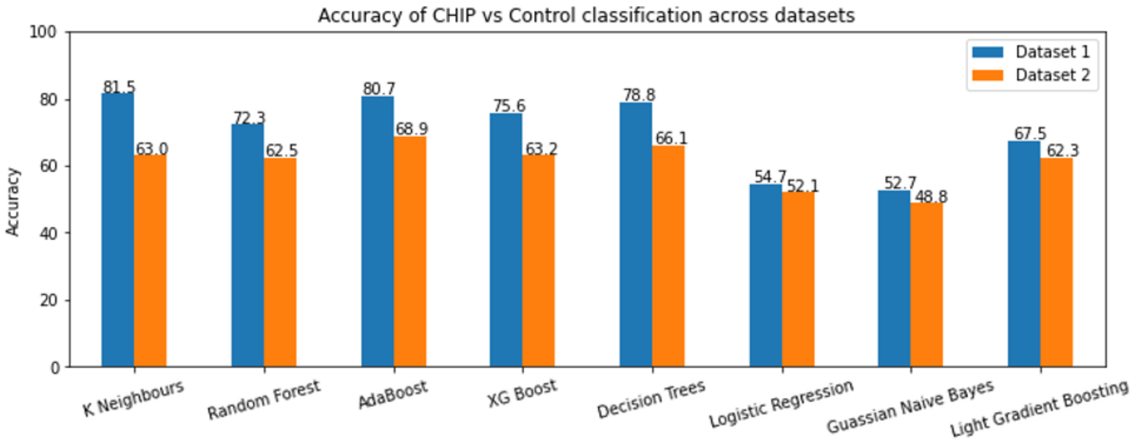
Before we were able to produce the final models and accuracies that will be discussed in this report there were several pre-processing steps which were carried out to get our data to a state where modelling could occur. These steps include converting strings to floats and One Hot Encoding, removing outliers and separating unique elements of a variable.

To reduce the amount of time spent cleaning and manipulating the dataset before modelling we made a python file called ‘preprocessing’ containing all of the functions needed so that we could call this file instead of having multiple cells of code. The first function in this file is remove_redundant() which drops duplicate rows, rows with blank or unknown *chipOrControl* values and drops the ‘*d.barcode*’ column. The second function is loci_split(), which splits the *‘loci’* variable into three separate columns, chromosome, chromosome location and nucleotide. ratio_to_int() converts *BIAS, REFBIAS* and *VARBIAS* ratios from string to float values. It works by converting the string data type into an integer, where the string is in the format 'x:y'. The output would then be an integer x divided by an integer y., which iterates over every value of a specified variable in the dataframe to convert the string ratios to float values.

Another element of preprocessing included One Hot Encoding, where we converted the *chipOrControl* values once unknowns were removed to 1 if they were CHIP positive and 0 if they were CHIP negative. As these were categorical values, we needed to give an integer value to each possible value of each categorical variable, one hot encoding was also applied to the 5 genes selected from the correlation with CHIP positive variants.

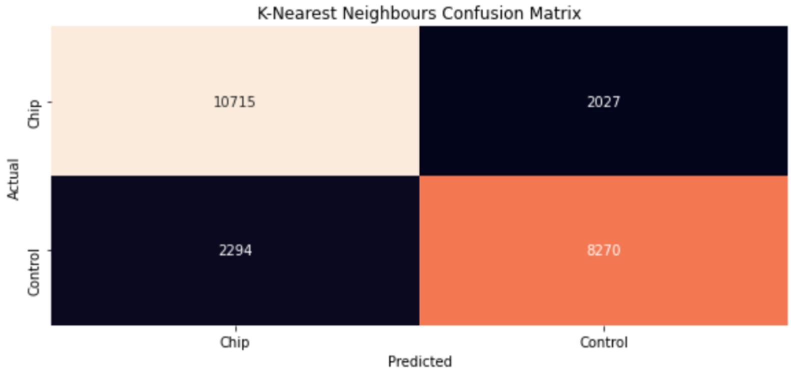
RESULTS

Our group applied a large variety of different models and classifiers to our dataset to find the best one which resulted in the highest accuracy. In the end, K-Nearest Neighbours and AdaBoost resulted in the highest accuracies of around 80%.



Accuracy of Our Classifiers Across Datasets

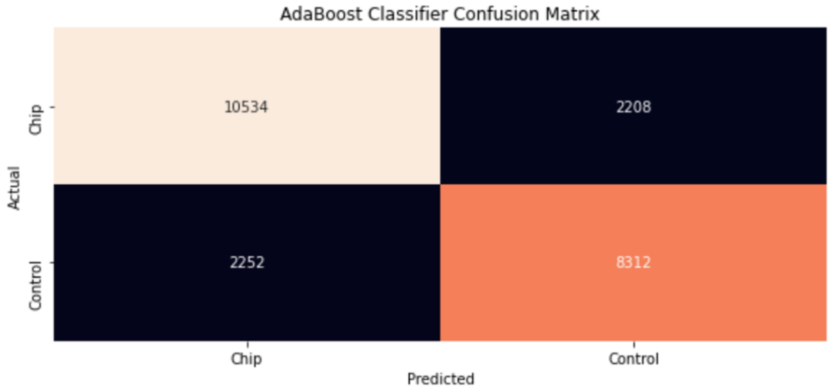
K-Nearest Neighbours was the most accurate classifier with an accuracy score of 81.5%. K Nearest Neighbours is a simple supervised machine learning algorithm that operates by calculating the distance of a data point to all old existing training points (*Harrison, 2018*). It then selects K nearest data points, where K is an integer, and then assigns this new data point the same class as the class of majority of K nearest data points, that being either CHIP or Control.



Confusion Matrix for K-Nearest Neighbours

K Nearest Neighbours is much faster than most algorithms because it does not require training before making predictions. *fig 5* within the appendix shows the highest accuracy came from when K was set to equal 4, which meant that for the best results, each new point was determined by its four nearest neighbours. *fig 6* in the appendix highlights the results of cross validating the model, where 4 remains the optimal number of neighbours..

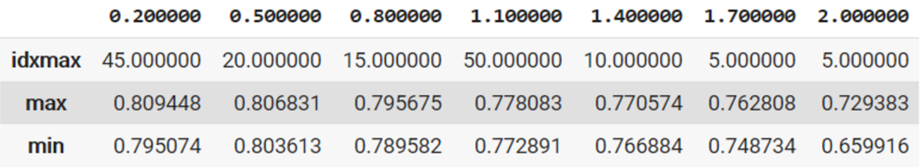
AdaBoost was our second most accurate classifier with an accuracy score of around 81%. AdaBoost works by combining multiple weaker classifiers into a single strong one. After every iteration, it associates weights with each observation and then increases the weights on difficult to classify or incorrectly classified instances while decreasing the weights on those already handled well. This weight system allows for the model to focus more on the incorrect predictions rather than wasting memory on already known instances.



Confusion Matrix for AdaBoost

After testing each base estimator for AdaBoost, it was discovered that using Decision Trees as our estimator led to the highest results before implementing any hyperparameters. It was also discovered that the algorithm 'SAMME' had a higher accuracy than 'SAMME.R' for our model. SAMME.R is the default algorithm for AdaBoost, it uses probabilities as estimates as to whether a patient is CHIP or Control. Contrast this to SAMME, which outputs either 0 or 1 instead of a probability. SAMME.R is normally seen as the better algorithm as it typically converges faster than SAMME. However, in this case, the SAMME algorithm is better.

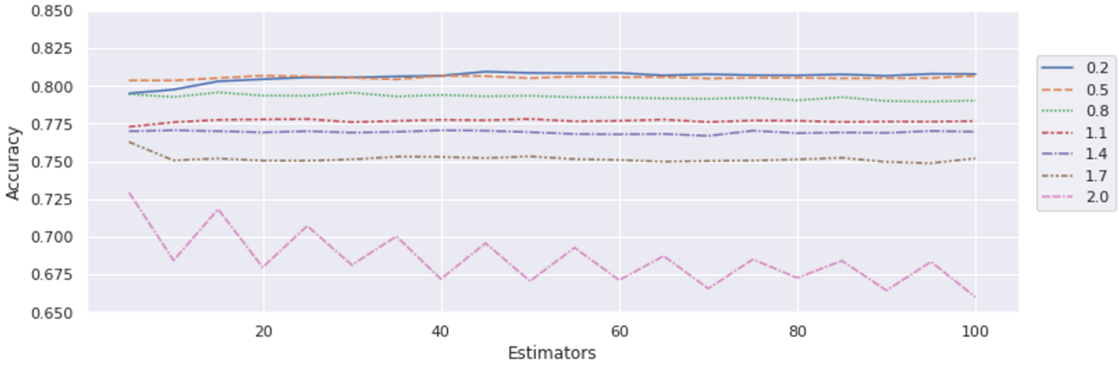
Apart from the algorithm, the two next most important hyperparameters for AdaBoost are the learning rate and the number of estimators. Therefore, to obtain the optimal hyperparameters, we went through each combination to see which resulted in the highest accuracy. From the table below, the optimal learning rate for our AdaBoost classifier was 0.2 with 45 iterations. Similarly, using a learning rate of 0.5 also led to accuracies of over 80%.



	0.200000	0.500000	0.800000	1.100000	1.400000	1.700000	2.000000
idxmax	45.000000	20.000000	15.000000	50.000000	10.000000	5.000000	5.000000
max	0.809448	0.806831	0.795675	0.778083	0.770574	0.762808	0.729383
min	0.795074	0.803613	0.789582	0.772891	0.766884	0.748734	0.659916

Accuracies for Different Learning Rates and their Optimal Estimators

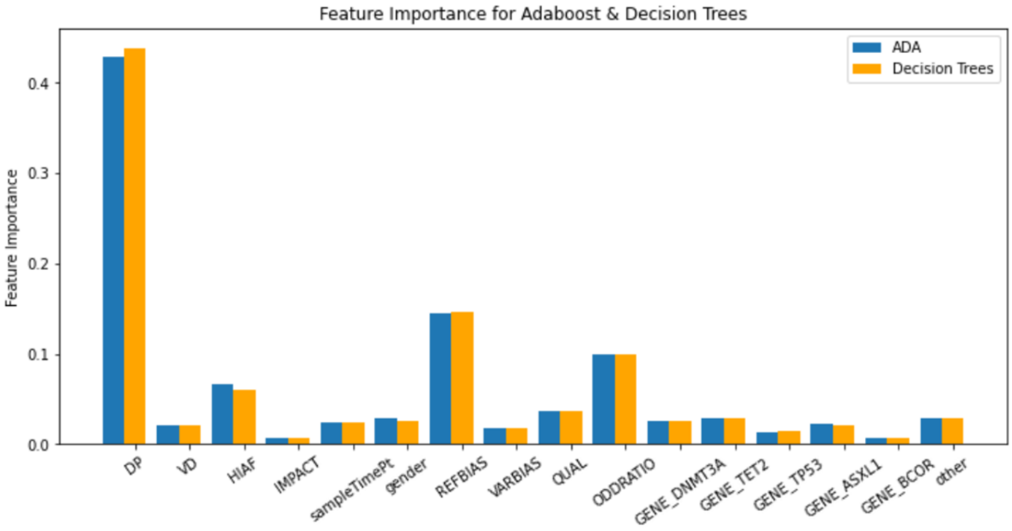
Using the Seaborn library, the accuracy vs number of estimators was also graphed out for each of the learning rates. The impact of the number of different estimators can be seen by the graph and *fig 7* in the appendix. In general, the number of estimators does not have a large effect on the accuracy.



Graph of the Different Learning Rates and their Optimal Estimators vs Accuracy

Feature Importance

Feature importance provides a weighting of features which were most vital in classifying a patient as CHIP positive or Control. AdaBoost and Decision Tree Classifiers unanimously agreed upon *DP, REFBIAS, ODDRATIO* and *HIAF* as top four most important features.



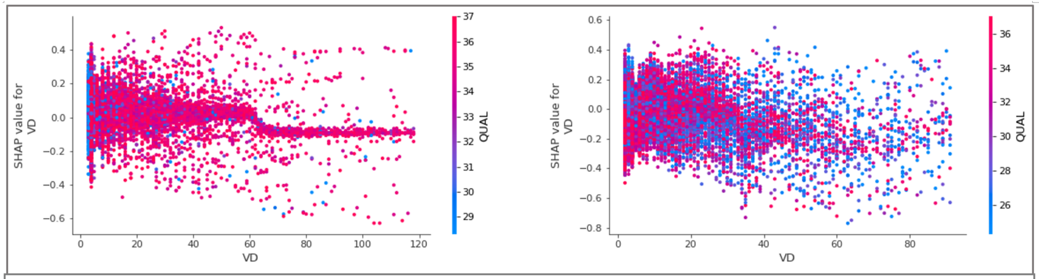
Graph of AdaBoost & Decision Trees Feature Importance

SHAP Values

To explore the reasons for these lower accuracies, SHAP values are used to plot figures, providing a visual representation of the differences between the two datasets. If positive, then the chances of the sample being CHIP positive are high, and the variable chosen as the predictor represents the range of values for which the model has predicted CHIP or control *(Molnar, C. (2021))*.

The SHAP values were studied on three different models namely the KNN (Knearest Neighbours) model, Random forest classifier, and AdaBoost classifier. Firstly, a variable importance plot is created for the two datasets depending on the choice of the model. *fig 1* within the appendix shows the variable importance plot for the KNN model on the two datasets where it can be observed that *DP, VD* and *QUAL* were the variables with the highest feature importance.

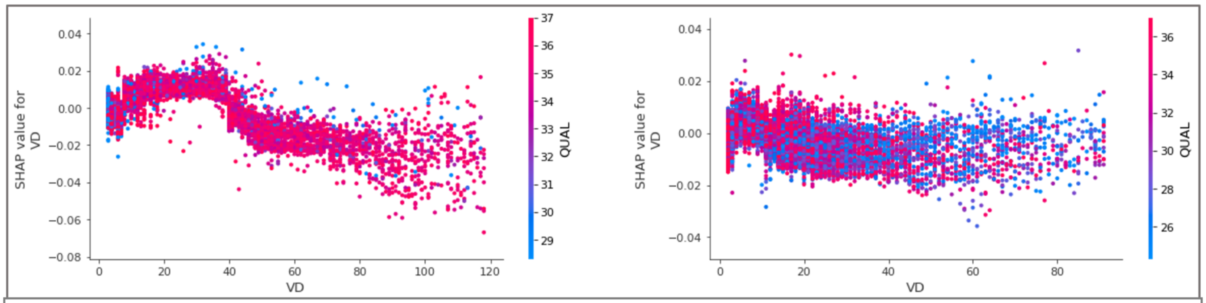
A dependence plot shows the marginal effect two features have on the model's predicted outcome. The following dependence plots for three models illustrate the relationship between the target variable (*chipOrcontrol*) and a feature (*VD*). The colour of the data points correlates to the feature *QUAL*, and the left y-axis shows the SHAP values for *VD*. *VD* and *QUAL* are chosen as the interaction variable as they have a high feature importance and provide a clearer graph compared to the other variable's graphs. Since the variable *QUAL* demonstrates the quality of the data that was read by the machine, it is the most beneficial feature to examine the reason for lower accuracies



KNN model's dependence plot for dataset1 (left) and dataset2 (right)

Upon referring to the KNN model's dependence plot for dataset 1, it can be observed that when *VD* ranges from 60-120, the data points are clustered towards a negative SHAP value implying an increased chance of predicting the variant as control (CHIP negative). In contrast, the dependence plot for dataset2 displays the points more widely spread out than dataset1, thus making it difficult to predict the outcome.

A similar result from KNN model's plot can be seen in the dependence plot for the random forest classifier.

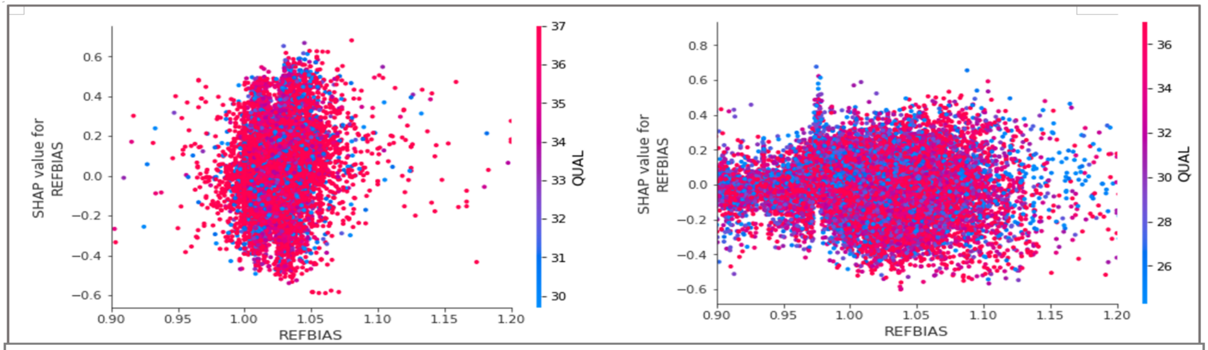


Random forest classifier's dependence plot for dataset1 (left) and dataset2 (right)

The figure shown above provides a more distinct shape to the graph compared to the plot for KNN model. Not only does this plot show the *VD* values that have a negative impact on the model's outcome, but it also shows a clear positive impact on the model's outcome. In cases where the *VD* ranges from 0-40 (in dataset1), the data points lie around a positive SHAP value suggesting a greater chance of predicting CHIP positive variants. However, for dataset 2, the points are varied around different SHAP values but mostly lying around zero.

Aside from the larger variance that can be clearly seen in dataset 2, the colour of the data points also shows a significant difference between the two datasets. The points in dataset1 are mostly coloured in pink correlating to a *QUAL* value of 35 and above. This high number for *QUAL* implies more accurate reads of the data. However, in dataset 2 the points are dominated by the blue colour suggesting lower *QUAL* value and thus a less accurate data readings. These factors could be a possible reason for the lower accuracies in the second dataset.

The final dependence plot is on the AdaBoost classifier where the model's feature importance displayed *REFBIAS* with a higher importance than *VD*. For this reason, the interaction variables used for this dependence plot are chosen as *REFBIAS* and *QUAL*. Nevertheless, the outcome is similar to the results seen from the plots for other models.

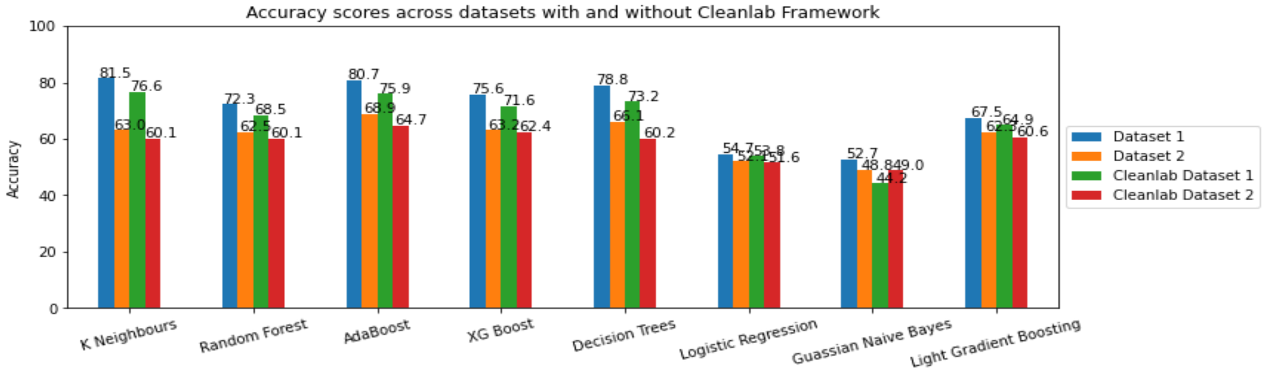


ADA boost classifier's dependence plot for dataset1 (left) and dataset2 (right)

Like the plots for the other models, the dependence plot for AdaBoost classifier shows a larger variance in the second dataset. In dataset 1, the points are clustered around a *REFBIAS* value of 1.00 to 1.10, however in dataset 2 they are widely spread. Moreover, like mentioned above the *QUAL* values for dataset1 are higher than dataset 2 (represented by the pink dots). Thereby, representing the less accurate reads of data for dataset2.

Label Noise and Modelling Improvements

Though the accuracy of classification models are favourable, they can be significantly improved if the inherent sequencing noise of the machines are taken into account. Although utilising cleanlab had the potential to generate good results, it produced less than favourable outcomes. Specifically, all models had accuracy scores decreased by around 3-5%.



There are two possible reasons for this result:

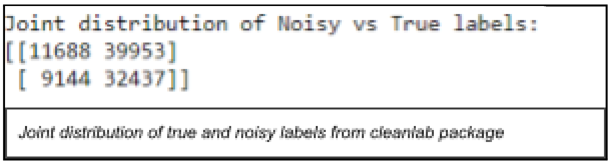
1. The dataset had little to no label noise, resulting in worse performance as cleanlab changes true positive classifications to a falsified label.
2. Input parameter, *est_nm,* contained inaccurate values, resulting in the model predicting label noise incorrectly.

The former reason is very possible. The project introduction brief stated that misclassifications were possible within the control subclass, but not a certainty.

In particular, a correspondence with Nick Wong echoes this line of reasoning:

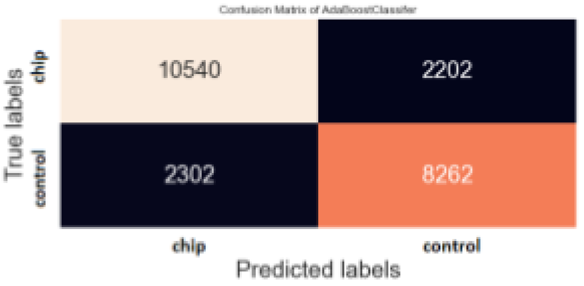
"[Classifications are] a 'truth set' but with limitations. One being, that you have seen control samples where they may actually be CHIP positive and I am open to accept that, if you are confident that the calls on those are real and not noise."

This further exemplifies the possibility that sequencing noise did not affect the labelling, alluding to a "truth set" where label noise was not present. The latter reason is also probable, whereby if the predicted probability (given in the form of a noise matrix) of labels is not inputted with a high degree of accuracy, then the model itself will logically fail to improve. Although we cannot ascertain the validity of the noise matrix, the joint probability distribution of noisy and true labels, P(s,y), which encapsulates the label noise within the class-conditional m x m matrix, can be uncovered as an attribute (*Curtis G. Northcutt, 2021).*



```
Joint distribution of Noisy vs True labels:
[[11688 39953]
 [ 9144 32437]]
```

Joint distribution of true and noisy labels from cleanlab package



Confusion matrix of classifications of AdaBoost

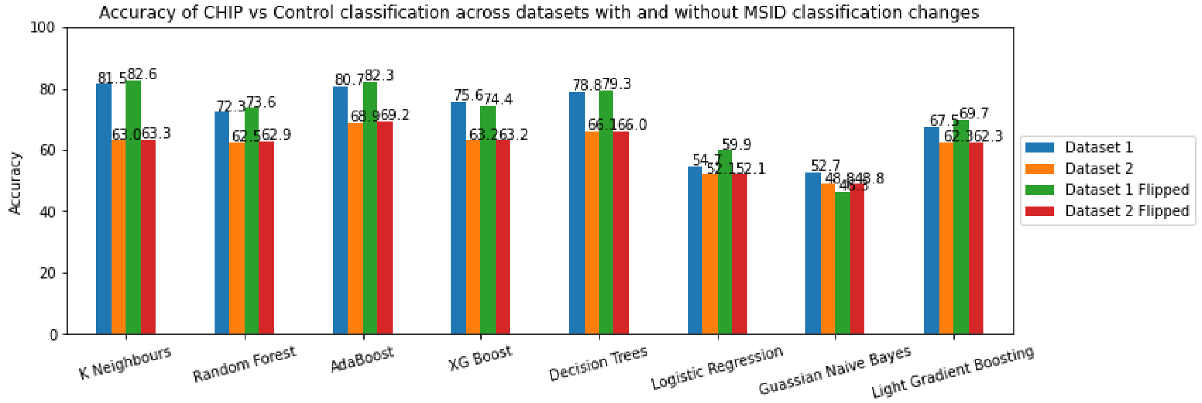
The differences between the confusion matrix of previous models and the joint distribution attribute elucidates a cause for decreased model accuracy. Particularly, the rate of true positive and true negative is roughly equal within the confusion matrix, yet differ vastly within cleanlab's joint distribution input; the lower accuracies within cleanlab's wrappers can be inferred from this distinction.

The second method of uncovering label noise was based on the assumption that structured group bias was present within both datasets. Through grouping model accuracies by *MSID,* visualisation of where models classify incorrectly can be taken into account. *fig 2* within the appendix displays the true classifications of testing data against the predicted classifications of the KNearestNeighbours classifier, grouped by *MSID*; the figure highlights the set of misclassifications for a particular group of *MSID*.

Taking this one step further by finding the frequency of model inaccuracies per patient, if a *MSID* is continuously mislabelled, then there is a high chance that label noise exists, and should be changed from Control to CHIP.

By taking the top performing models and finding the error rate, an average can be found for the percentile of misclassifications per *MSID*. Following, a filter can be set for a minimum percentage of errors, as seen in the DataFrame (appendix *fig 3)*.

Furthermore, these findings align with Nick's initial statement, whereby there existed control samples which should have been classified as CHIP. Hence, by concluding that the above *MSID* labels have been misclassified, and by running the numerous machine learning models again, we observe that the accuracy rates have increased.



CONCLUSIONS

Ultimately, the best performing models were KNearestNeighbours and AdaBoost classifiers, achieving an accuracy of over 80% for dataset 1; AdaBoost was also the best performing model for dataset 2 by producing an accuracy of 68.9%. Although other models were implemented, no model performed better than KNN model and AdaBoost classifier due the amount of label noise in the datasets. The variables *DP, REFBIAS, ODDRATIO* and *HIAF* are indicative features for CHIP classification. These features are not a surprise, as depth read *(DP*) and allele frequency *(HIAF*) hold a solid definition in which CHIP is formatively defined (≥ 2% allele frequency), while *ODDRATIO* and *REFBIAS* provides indication on reliability in terms of a DNA sequencing read.

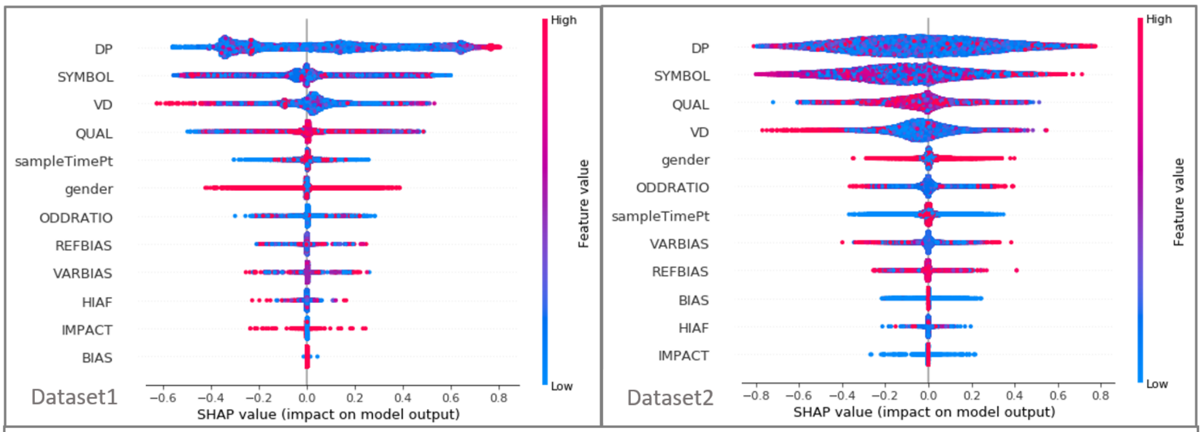
The SHAP values obtained from the top three best performing models provide two possible factors for the second dataset's lower accuracy. These factors are larger variance and low-quality reads of the second dataset. By re-labelling one of the variables (*MSID*) that was misclassified, it was noticed that the accuracy of the models improved slightly by 1-5% (most notably logistic regression).

Major improvements to be considered in this project is to identify the label noise in a more concrete manner, which could increase the accuracies of the models where group bias occurs. Extending upon this concept, applying predictive learning with structured data. In this case, training and testing on separate *MSID*'s, such that the model trains without a particular group; this would reduce overall bias. Furthermore neural networks in a tabular setting may be implemented via Fast.ai and Tabnet, in addition to utilising the PyTorch library. Other exploratory analysis on variables, such as *loci*, (in particular, chromosome location) could be explored in more depth, which may arise in visualising trends of where CHIP is genomically based. In a similar vein, during model development, nucleotide information was scrapped due to time constraints. Given that a genomic mutation may arise through many different methods (base substitutions, deletions and insertions, etc.), investigation into arising trends may give indication into how CHIP is classified.

APPENDIX

fig 1, variable importance plot using SHAP values. The variable importance plot demonstrates the following:

1. Feature importance of the variables in the dataset arranged in descending order.
2. The horizontal axis (x-axis) shows the effect of the variable on the model output.
3. Correlation: Positive and negative impact on the model is shown on the x-axis along with the colours representing level of impact as high or low – red or blue colour, respectively.



KNN model’s variable importance plot for dataset1 and dataset2

fig 2. Labels of testing data and KNearestNeighbours classifier grouped by MSID. The group of MSID (red highlight) in X_test is classified as CHIP, yet the KNearestNeighbour model classifies the group as control.

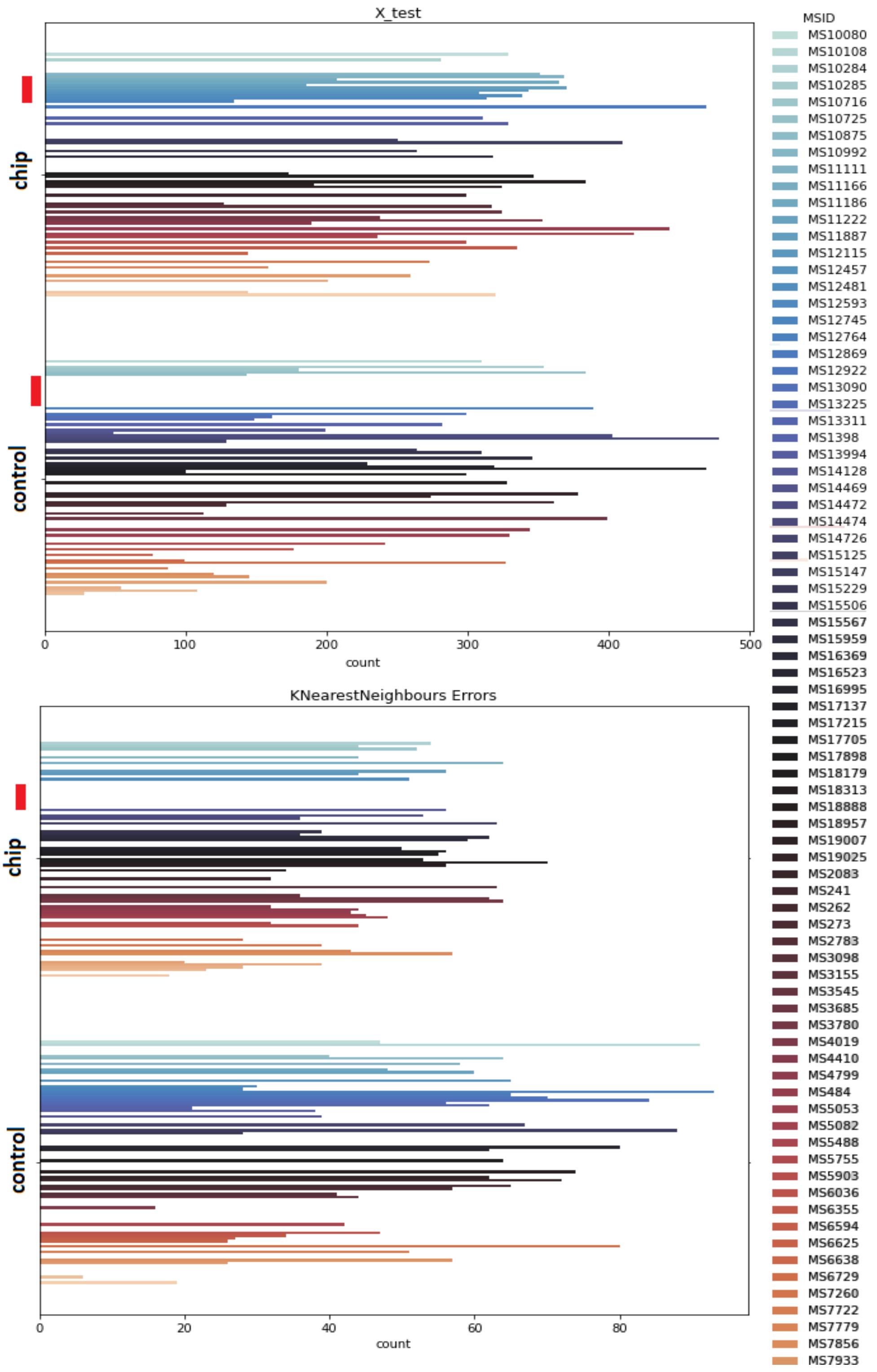
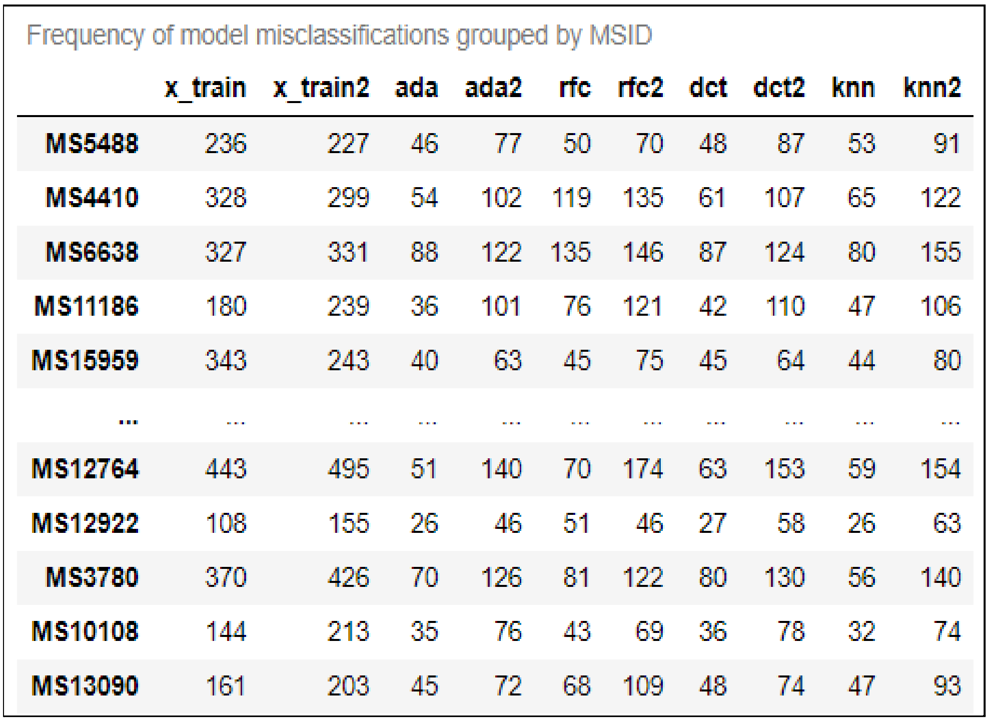


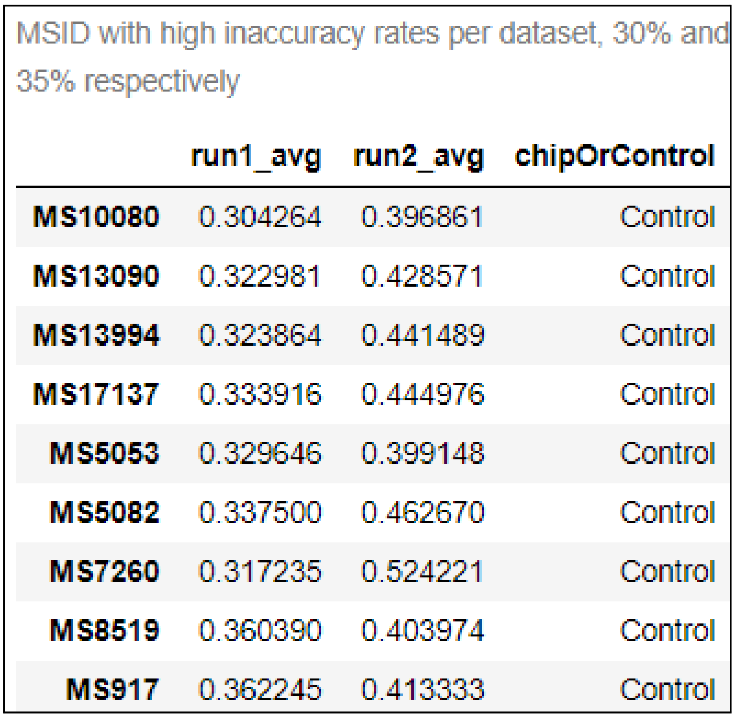
fig 3. DataFrame depicting the frequency of misclassifications across models in both datasets, grouped by MSID.



Frequency of model misclassifications grouped by MSID

	x_train	x_train2	ada	ada2	rfc	rfc2	dct	dct2	knn	knn2
MS5488	236	227	46	77	50	70	48	87	53	91
MS4410	328	299	54	102	119	135	61	107	65	122
MS6638	327	331	88	122	135	146	87	124	80	155
MS11186	180	239	36	101	76	121	42	110	47	106
MS15959	343	243	40	63	45	75	45	64	44	80
...	...	...	...	...	...	...	...	...	...	...
MS12764	443	495	51	140	70	174	63	153	59	154
MS12922	108	155	26	46	51	46	27	58	26	63
MS3780	370	426	70	126	81	122	80	130	56	140
MS10108	144	213	35	76	43	69	36	78	32	74
MS13090	161	203	45	72	68	109	48	74	47	93

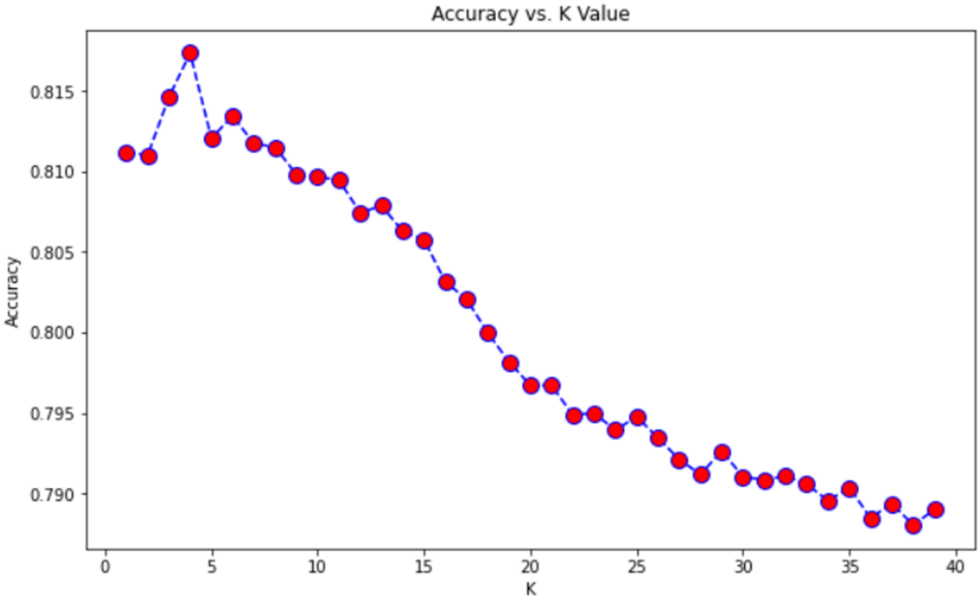
fig 4. Particular MSID which displays high misclassification rates.



MSID with high inaccuracy rates per dataset, 30% and 35% respectively

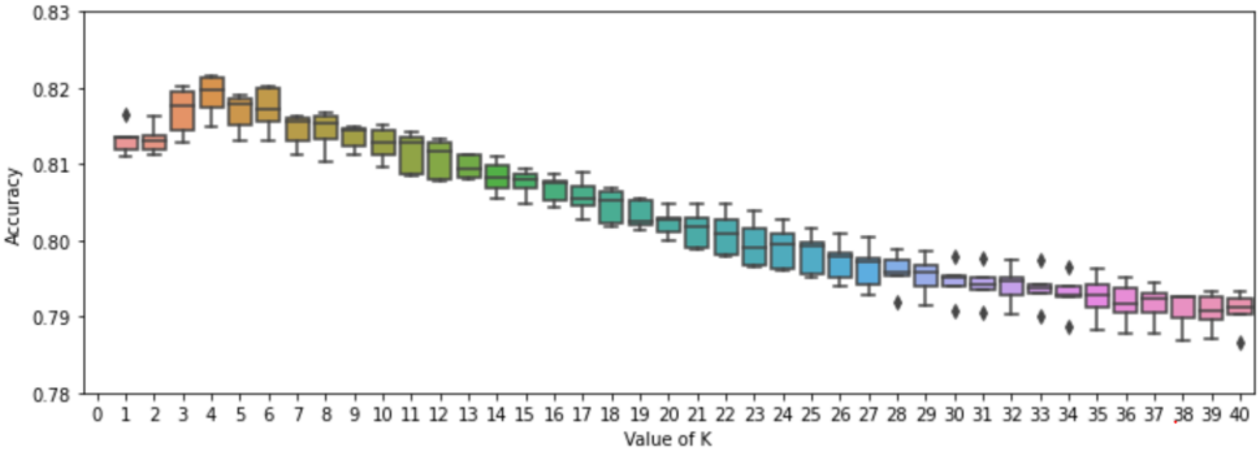
	run1_avg	run2_avg	chipOrControl
MS10080	0.304264	0.396861	Control
MS13090	0.322981	0.428571	Control
MS13994	0.323864	0.441489	Control
MS17137	0.333916	0.444976	Control
MS5053	0.329646	0.399148	Control
MS5082	0.337500	0.462670	Control
MS7260	0.317235	0.524221	Control
MS8519	0.360390	0.403974	Control
MS917	0.362245	0.413333	Control

fig 5. Graph of K Values vs Accuracy for K-Nearest Neighbours



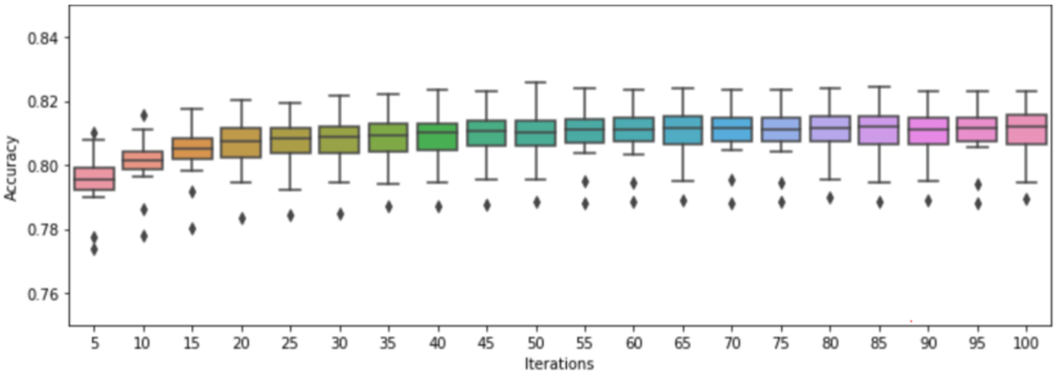
Graph of K Values vs Accuracy for K-Nearest Neighbours

fig 6. Graph of Cross Validation for K-Nearest Neighbours



Cross Validation for K-Nearest Neighbours

fig 7. Graph of Cross Validation for AdaBoost



Cross Validation for AdaBoost and Learning Rate of 0.2

BIBLIOGRAPHY

Curtis G. Northcutt, Lu Jiang, Isaac L. Chuang (2021). Confident Learning: Estimating Uncertainty in Dataset Labels. *Cornell University*. Retrieved May 28th from https://arxiv.org/abs/1911.00068

Explain Any Models with the SHAP Values—Use the KernelExplainer. (2021). Retrieved 28 May 2021, from https://towardsdatascience.com/explain-any-models-with-the-shap-values-use-the-kernelexplainer-79de9464897a

Explain Your Model with the SHAP Values. (2021). Retrieved 28 May 2021, from https://towardsdatascience.com/explain-your-model-with-the-shap-values-bc36aac4de3d

Lai, Z. (2016). AstraZeneca-NGS/VarDict. Retrieved 4 May 2021, from https://github.com/AstraZeneca-NGS/VarDict/wiki/VarDict-Raw-Output

Gondek, L. P., & DeZern, A. E. (2020). Assessing clonal haematopoiesis: clinical burdens and benefits of diagnosing myelodysplastic syndrome precursor states. *The Lancet Haematology*, *7*(1), e73–e81. Retrieved , 27 May 2021, from https://doi.org/10.1016/s2352-3026(19)30211-x

Harrison, O. (2018). Machine Learning Basics with the K-Nearest Neighbors Algorithm. Retrieved 24 May 2021, from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Molnar, C. (2021). 5.10 SHAP (SHapley Additive exPlanations) | Interpretable Machine Learning. Retrieved 28 May 2021, from https://christophm.github.io/interpretable-ml-book/shap.html

Phred-Scaled Quality Scores. (2021). Retrieved 4 May 2021, from https://gatk.broadinstitute.org/hc/en-us/articles/360035531872-Phred-scaled-quality-scores

THE ASPREE TRIAL and ASPREE-XT STUDY - ASPREE Australia. (2020, January 10). Retrieved May 28, 2021, from ASPREE Australia website: https://aspree.org/aus/about-us/the-aspree-study/

Veronica, A. (2021). Understanding Adaboost and Scikit-learn's algorithm:. Retrieved 28 May 2021, from https://medium.datadriveninvestor.com/understanding-adaboost-and-scikit-learns-algorithm-c8d8af5ace10#:~:text=Here%20is%20the%20main%20difference,sample%20belonging%20to%20a%20class

What to know about haematopoiesis. (2017, September 27). What to know about hematopoiesis. Retrieved May 28, 2021, from https://www.medicalnewstoday.com/articles/319544#:~:text=Hematopoiesis%20is%20the%20production%20of,the%20body%20manufactures%20blood%20cells.