**First 2 minutes (approx.) of your presentation -- *Your Motivation***

* **You MUST mention which dataset(s) you used, and what EXACTLY is your problem definition.**
* **This is your chance to explain your MOTIVATION. Don't get carried away; be precise and clear.**

**Next 3 minutes (approx.) of your presentation -- *Set The Stage***

* **Present your Exploratory Data Analysis and some initial data-driven Insights from the dataset.**
* **You MAY also mention how you are planning to set up the Analysis / ML problem for this case.**
* **You MUST mention how you collected / curated / cleaned / prepared the data for this problem.**
* **Did you only use tools and techniques learned in this course? What ELSE did you learn / try?**

**Next 3 minutes (approx.) of your presentation -- *Core Analysis***

* **If you used ML (regression, classification, or something else); mention mainly WHICH one(s).**
* **You may now briefly CLARIFY why and how the ML problem(s) aim(s) to solve your objective.**
* **How did you apply ML technique(s) to SOLVE your problem? Which model(s), how and why?**
* **Did you only use tools and techniques learned in this course? What ELSE did you learn / try?**

**Last 2 minutes (approx.) of your presentation -- *Finish Strong***

* **What is the OUTCOME of your project? Did it solve your original problem? Anything interesting?**
* **What are your data-driven INSIGHTS and recommendations / views towards the target problem?**

***Everything you plan to showcase should be presented within the 10 mins. Plan carefully and smartly.***

**Present the main aspects of your project and data-story in the 10 mins; extra items may be on GitHub.**

**No need to mention who in your team worked on which part of the project -- this should be on GitHub.**

**No need to cite references to other related works in your presentation -- this should be on GitHub too.**

**MOTIVATION**

Genetic disorders make up a large proportion of pediatric and infant deaths around the world. Genetic disorders arise from abnormalities in the genetic material of a person. Our genomic DNA codes for proteins. Mutations or aberrations in the genome lead to mistakes in the proteins produced, and eventually affect bodily functions to various extent. The human genome contains billions of single coding units, every one of which can be mutated and cause genetic disorders. As a result, predicting genetic disorders in a newborn requires expensive whole genome sequencing and testing, and may not be readily available to many parents.

With the Genetic Disorder dataset from Kaggle, we hope to build a machine learning model that can predict genetic disorders in newborns and children, without employing high throughput screening of the genome, but rather, more readily available data such as family history, conditions of the pregnancy and birth, and the health of the mother and child.

**EXPLORATORY DATA ANALYSIS**

[slide: data dimensions, tree showing classes and subclasses]

The Genetic Disorder dataset consists of 45 columns of information on over 22 thousand patients in the USA. One of these columns named Genetic Disorder classifies patients into 3 broad categories of genetic disorders, namely single-gene, multifactorial and mitochondrial disorders. Each broad category is further broken down into 3 specified diseases that patients are diagnosed with, giving us 9 subclasses. Rows with missing subclasses are dropped, while rows with missing classes are filled in based on their subclasses.

[slide: plots of 2 response columns]

Of the 3 broad categories, mitochondrial disorders account for over half of the observations, while multifactorial disorders make up only about 10%, and the rest are single-gene disorders. Among the subclasses, Leigh syndrome and mitochondrial myopathy, both mitochondrial disorders, make up the largest proportions, followed by single-gene disorders cystic fibrosis and Tay-Sachs. It is understandable that subclasses under multifactorial disorders are low on the count list, given the total number is very small compared to the others.

Unsurprisingly, Leigh syndrome and mitochondrial myopathy are the majority among mitochondrial disorders, while cystic fibrosis and Tay-Sachs collectively account for most single-gene disorder cases. There is a small number of multifactorial disorders and most of them are cases of diabetes, while the minority are Alzheimer’s and cancer.

The other 43 columns hold information that can potentially be used as predictors. Many of these containing names and locations are deemed irrelevant to the prediction of the genetic disorders, and are therefore removed from the dataset.

[slide: table listing columns under each group maybe]

The remaining 30 columns can be divided into different groups of information types.

First is columns with basic and medical information on the patient such as age, gender, blood counts. Most of these attributes are uniformly distributed among the patients and seem to make no impact on the type of genetic disorder, except for a group of 5 unspecified symptoms named Symptom 1 to 5. Due to the imbalance in the classes, it is hard to say whether individual symptoms predispose any particular class. However, patients with multifactorial disorders seem to be more likely to show the symptoms than not.

By converting Yes and No to 1s and 0s and summing up the symptoms for each row, new observations can be made. Consistent with the previous observation, patients with multifactorial disorders are likely to show at least 3 out of 5 symptoms, while those with mitochondrial and single-gene disorder tend to show at most 2 symptoms.

Zooming into subclasses,

**Missing Values**

As Genetic Disorder column is directly related to the Disorder Subclass, the rows with missing values in Genetic Disorder were directly filled with values based on the Disorder Subclass according to the table below.

| **Genetic Disorder** | **Disorder Subclass** |
| --- | --- |
| Single-gene | Cystic fibrosis |
| Hemochromatosis |
| Tay-Sachs |
| Multifactorial | Alzheimer’s |
| Cancer |
| Diabetes |
| Mitochondrial | Leigh syndrome |
| Mitochondrial myopathy |
| Leber’s hereditary optic neuropathy |

To fill in missing values in other predictors, we have tried both simply filling in with the mode of the predictor column using SimpleImputer and iteratively impute missing values using IterativeImputer. IterativeImputer uses the multivariate imputation by chained equations approach whereby each predictor is modelled as a function of other predictors and the missing values were predicted imputed. We used Naïve Bayes model in out iterative imputation as we did label encoding just for imputation purpose and Naïve Bayes can ignore order of data. The process repeats until maximum number of iterations decided is reached or convergence occurred where there are hardly any more changes in values as compared to the previous iteration. IterativeImputer yielded higher accuracy in subsequent machine learning models, hence we went with it.

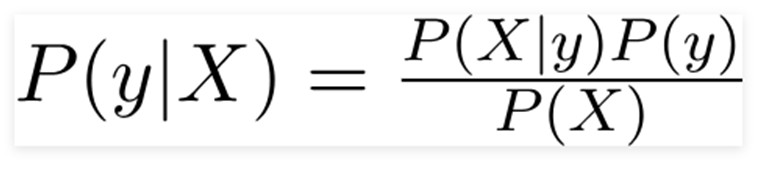
https://machinelearningmastery.com/iterative-imputation-for-missing-values-in-machine-learning/

**CORE ANALYSIS**

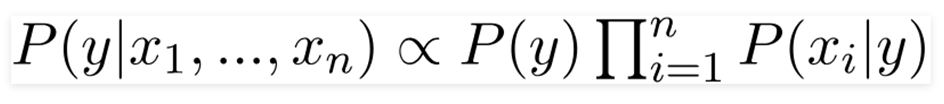
Since there are 3 main classes and 9 subclasses, there is one model to predict the 3 main classes. Then a submodel for each main class to predict the 3 subclasses. One more model is formed to predict the 9 subclasses together.

The main model used was random forest.

**Naïve Bayes**

****

Using Bayes theorem, we can find the probability of response variable happening given that the predictors have already occurred.



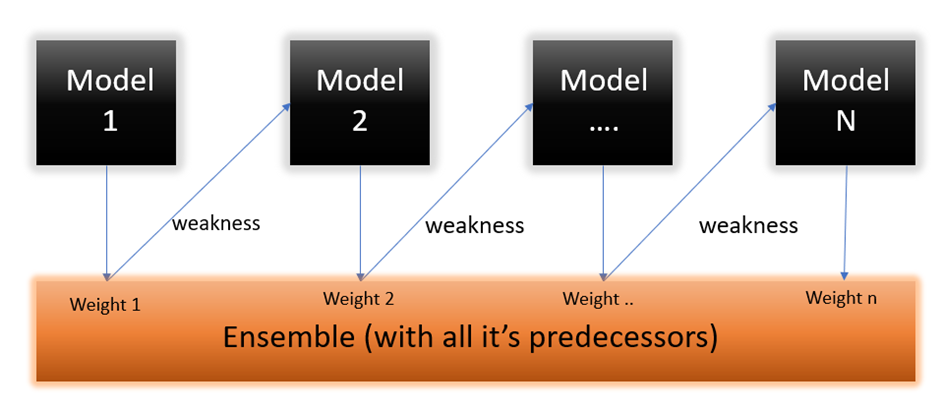
As our predictors are Genetic Disorder and Disorder Subclass, which are multiclass classification problems, the response variable value is determined by the highest probability of y occurring given X.

https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

**AdaBoost**

Boosting algorithm aims to improve the prediction power by converting a number of weak learners to strong learners. For AdaBoost, we used decision trees with 1 levels known as decision stumps. A model is first built and equal weights were given to all the data points. Higher weights were then assigned to points that were wrongly classified. Points with higher weight were then given more importance in the subsequent model as they will be oversampled within the new dataset picked. The process will continue until the error is minimised.

<https://www.youtube.com/watch?v=peh2l4dePBc>



https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/

**Anomaly Detection**

Using unsupervised learning, we also attempted to find out anomalies within our dataset using the PyOD package using K-Nearest Neighbour model. The model hyperparameter was set such that a data point distance to its kth nearest neighbour is viewed as the outlying score and the contamination was set to change from 0.01 to 0.1.

https://towardsdatascience.com/introducing-anomaly-outlier-detection-in-python-with-pyod-40afcccee9ff

**KModes**

KModes was applied in our dataset for Genetic Disorder classification using the KModes package as our dataset consists of categorical variables which are unsuitable for KMeans. KModes uses dissimilarity between data points. An Elbow curve was plotted to find optimal hyperparameter K value which we determined to be 6 for Genetic Disorder. K refers to the number of points picked at random to label as clusters. Dissimilarities were then calculated and each observation was assigned to the closest cluster. New modes of clusters were redefined and the process repeats until there is no more re-assignment of points.

https://www.analyticsvidhya.com/blog/2021/06/kmodes-clustering-algorithm-for-categorical-data/

**Comparison**

The training and testing set were stratified by both Genetic Disorder and Disorder Subclass and different imbalance treatments were conducted. I will explain the effects of stratifying against Genetic Disorder first. For imbalanced data, applying the machine learning algorithms, yielded similar results with signs of overfitting in naïve bayes and adaboost as seen by the training set accuracy being higher than test set with the exception of RandomForest having better accuracies but also with signs of overfitting. Undersampling resulted in more overfitting but without much improvement in accuracy. Random oversampling resulted in overfitting in AdaBoost but also generally improved accuracy in all models with RandomForest performing the best. Oversampling using SMOTE yielded the best results with also AdaBoost fitting but increased accuracies in all models beyond 60% accuracy.

Stratifying against Disorder Subclass for imbalanced data yielded similar results as stratifying against Genetic Disorder. Random undersampling performed worse in all models when stratified against Disorder Subclass with severe overfitting in all models. Meanwhile, random oversampling and SMOTE performed worse in all models when stratified against Disorder Subclass but not many signs of overfitting.

As such, SMOTE and RandomForest seem to be suitable for our data. Dataset was then treated using SMOTE and the RandomForest model was built to predict Genetic Disorder. Using the predicted output of this model, another 3 RandomForest models were used to predict the Disorder Subclass values. The final average accuracy was noted to be 24.1% for train data and 23.8% for test data, which is an improvement over the baseline of 11.1%.

**OUTCOME**

The 30 variables were used for training and prediction of the model. Afterwhich, checking the feature importance of the models shows that the variables: symptoms 1-5 and inherited genes, are more significant for all the disorders. This is the same as what we have found from our EDA.

Using only the more important symptom and gene variables, the accuracy of the prediction was similar to the previous model, except that the total time taken for training and prediction of the model was significantly reduced to half.

However, given that the symptoms and genes are not named in the original dataset description, it is difficult to associate the symptoms and genes with any of the disorders, or explain why the frequency of the symptom and gene variables deemed important for prediction was similar across all 3 disorders.

**README**

**Contributors**

Guan Jia Sheng (U2040851A)

Nguyen Ngoc Minh Truc (U1940862C)

Lim Yong Yee (U2040881F)

4 notebooks:

1. EDA

* Identify pattern within dataset
* Check association between variables and disorders

2. Missing Value Treatment & Machine Learning

* Iterative imputation
* Model family selection
* Model optimisation and building
* Model evaluation
* Based on full dataset without assumptions

3. Replaced Values

* Alternative chained model
* Built with dataset values replaced based on assumptions obtained from EDA

4. Extra

* Fill missing values of predictors with mode of each column
* Comparison of missing value treatment between iterative imputation and mode

**Problem Definitions**

1. Predict classes (Genetic Disorder) and subclasses (Disorder Subclass) in the Genetic Disorder dataset
2. Identify the best model for prediction of classes and subclasses
3. To identify the most important variables for prediction

**Exploratory Data Analysis**

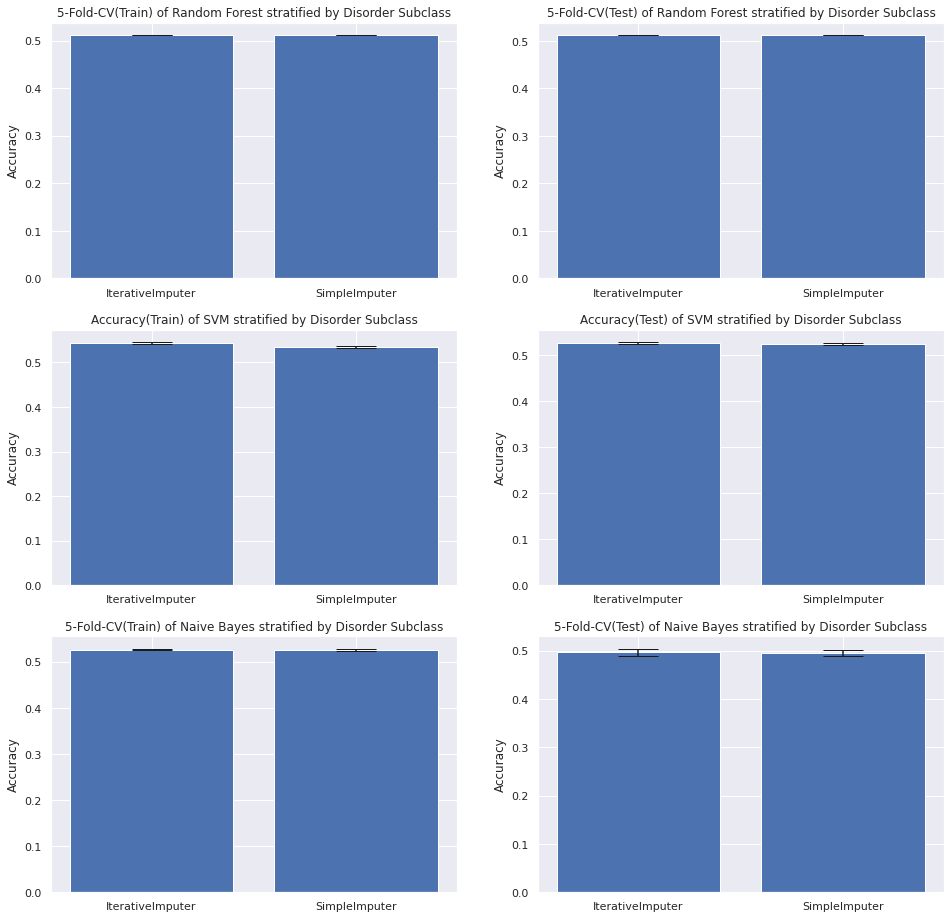
The analysis is found within the EDA notebook.

**Missing Values Treatment**

* Genetic Disorder column directly related to the Disorder Subclass
* Missing values in Genetic Disorder directly filled with values based on the Disorder Subclass

| **Genetic Disorder** | **Disorder Subclass** |
| --- | --- |
| Single-gene | Cystic fibrosis |
| Hemochromatosis |
| Tay-Sachs |
| Multifactorial | Alzheimer’s |
| Cancer |
| Diabetes |
| Mitochondrial | Leigh syndrome |
| Mitochondrial myopathy |
| Leber’s hereditary optic neuropathy |

* Missing values in other predictors - Tried two methods
  + Simply filling in with the mode of the predictor column (SimpleImputer)
  + Iterative imputation of missing values (IterativeImputer)
* IterativeImputer uses the multivariate imputation by chained equations approach
  + Each predictor is modelled as a function of other predictors and the missing values were predicted imputed
  + Naïve Bayes model was used in iterative imputation as label encoding was done initially just for imputation purpose and Naïve Bayes can ignore ordinality of data
  + Process repeats until the maximum number of iterations decided is reached or convergence occurs whereby there are hardly any more changes in values as compared to the previous iteration
* IterativeImputer and SimpleImputer yielded similarity in this dataset when a Random Forest model is built
  + Either one can be used in this dataset

****

**Imbalance Treatment**

**Imbalance**

* Performance of models created were evaluated using original imbalance data
  + The results serve as a control to compare with other imbalance treatment methods
* The imbalanced dataset was hypothesised to produce the worse performing models

**Undersampling**

* Majority classes, from Genetic Disorder or Genetic Subclass, were undersampled randomly without replacement
* The sampling strategy was set to use “not minority” which resamples all classes except the minority class
* Shorter computational time for model training as dataset size was reduced
* May risk removing important data points that should have been used for model training

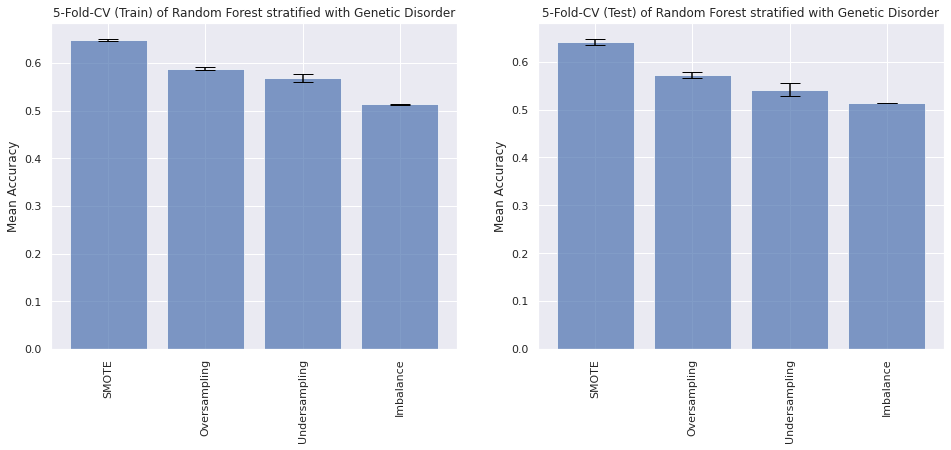
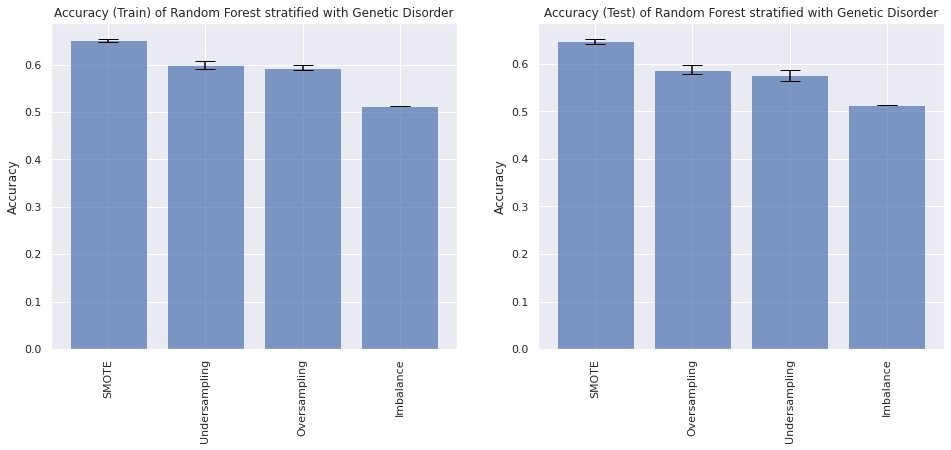
**Oversampling**

* Minority classes, from Genetic Disorder or Genetic Subclass, were oversampled by picking samples at random with replacement
* The sampling strategy was defaulted to use “not majority” which resample all classes but the majority class
* Avoid the removal of important data points as done in undersampling
* Computational runtime was significantly increased

**SMOTE**

* Instead of oversampling minority class by duplication, SMOTE synthesize new examples from the minority class which are close in the feature space.
* Provide additional information to the model
* Sampling strategy was defaulted to use “not majority” which resample all classes but the majority class.
* Best imbalance treatment method amongst the ones tried.

**Comparison of imbalance treatment methods**



* Class imbalance was noticeable in both Genetic Disorder and Disorder Subclass
* The performance of imbalance treatment decreases in the following order: SMOTE > Oversampling > Undersampling > Imbalance
* Error bars are standard deviation.

**Models Tested**

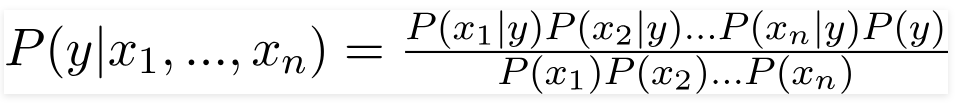
**Supervised Learning**

Support Vector Machine

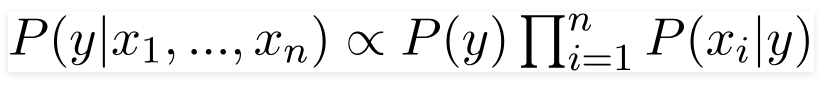
* SVM breaks the multiclass problem into multiple binary classification problems through the one-to-one approach
* Finds a hyperplane that best separates between every two classes, and is optimised by maximising the margin
* Large genetic disorder dataset with about 20000 rows meant that training time using SVM was high so it was not used as the final model.
* SVM Training time complexity = O(n3) VS Random forest Training time complexity = O(n\*log(n)\*d\*k)
  + where n = number of training samples, k = number of decision trees, d = dimensionality of the data.
* SVM was not carried out for SMOTE and Oversampling treatment on dataset with train test split stratified against Disorder Subclass due to immense runtime

Naive Bayes

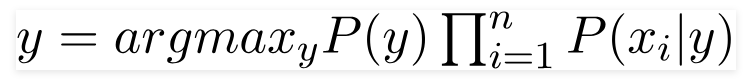
* Naive Bayes is a probabilistic model which can be used in Multiclass Classification, suitable for prediction of Genetic Disorder and Disorder Subclass.



* Bayes Theorem allows the finding of probability of a Genetic Disorder class (hypothesis) happening given a certain set of predictor values (evidence)
* Assumption of Bayes Theorem
  + Predictors are independent and that all predictors have equal effect on the outcome
* Due to the denominator being a constant, it can be removed in the above equation to give the equation below.



* Class of Genetic Disorder assigned by finding the class with the highest probability with the given predictors



* Naive Bayes did not work as well as other supervised learning models in this project and thus was not chosen to be used in the final model

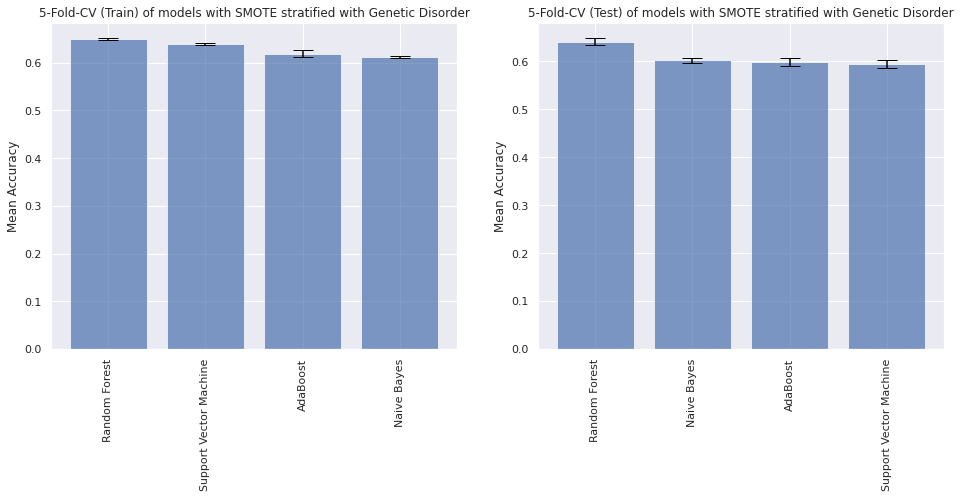
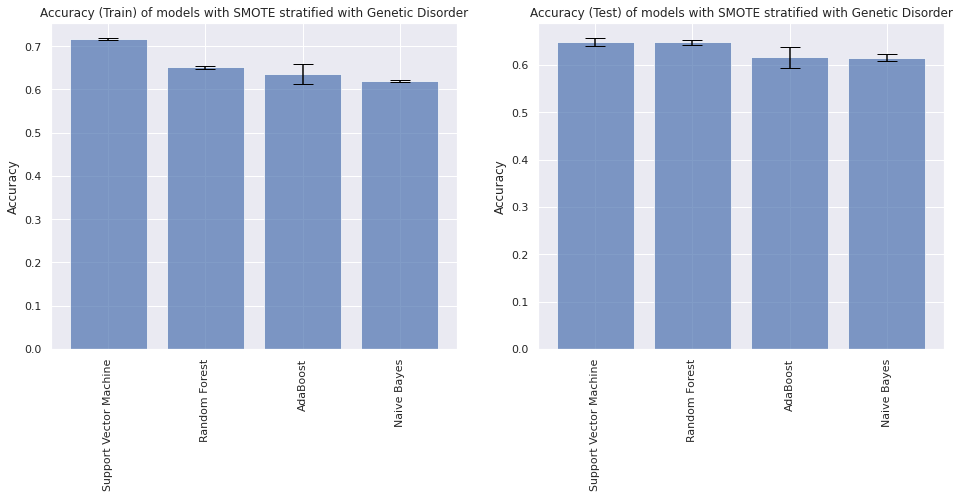
Random Forest

* Random Forest utilises the ensemble technique and is suitable for a Multiclass problem.
* By creating a bunch of decision trees that use different variables and data points for training, collaborative learning can be achieved and the class assigned ultimately will be via a “vote” whereby the class will be the one that is predicted by most decision trees.
* Best performing model out of all supervised learning models and was used for the final models.

AdaBoost

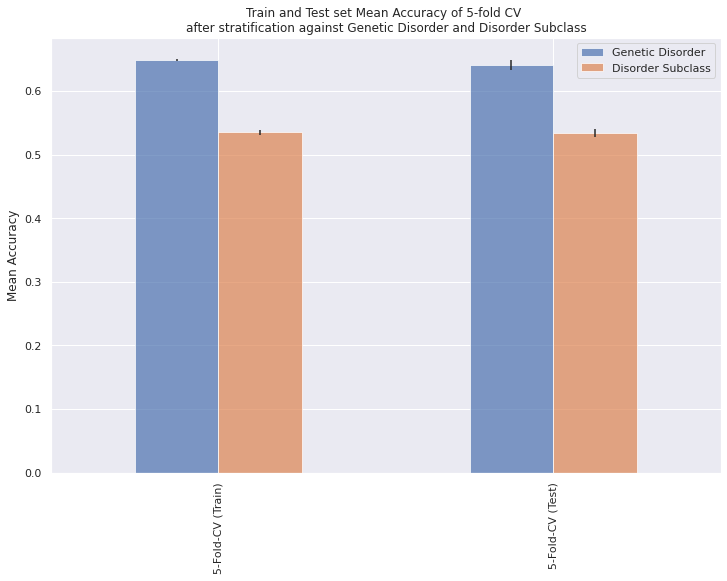
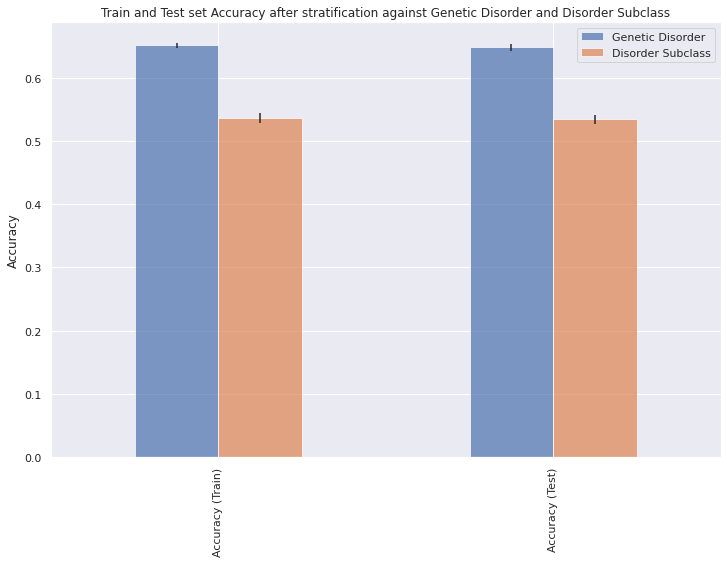
* Boosting algorithm aims to improve the prediction power by converting a number of weak learners to strong learners
* Decision trees with 1 levels were used as the weak models - Decision stumps
* Model mechanics
  + A model was first built and equal weights were given to all the data points
  + Higher weights were then assigned to points that were wrongly classified
  + Points with higher weight were then given more importance in the subsequent model as they will be oversampled within the new dataset picked
  + The process will continue until the error is minimised.
* Perform better than Naive Bayes in general but underperformed in comparison to Random Forest and Support Vector Machine.

**Supervised Learning Model Performance Comparison - SMOTE and Stratified against Genetic Disorder**

****

* Support Vector Machine had the highest accuracy for train and test set but was overfitted
* Random Forest performed relatively consistently throughout for accuracy and cross validation results
* Random Forest was decided to be the final model for Genetic Disorder model and Disorder Subclass models
* AdaBoost was performing better than Naive Bayes most of the time but was not chosen due to poorer results when compared to Random Forest
* Error bars are standard deviation

**Random Forest Model Performance Comparison after SMOTE - Stratified against Genetic Disorder and Disorder Subclass**

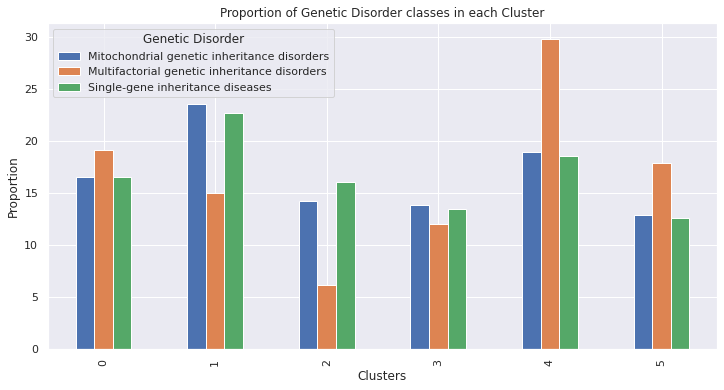
****

* Genetic Disorder model
* Stratification against Genetic Disorder during train test split showed better performance than stratification against Disorder Subclass
* Numbers of Alzheimer’s and Cancer cases are extremely small compared to total number of observations, stratification against Disorder Subclass was done to avoid excluding these data points from the train or test sets for both main and submodels (to keep same train and test set for all models)
* If prediction of Genetic Disorder classes was the only problem, then stratification against Genetic Disorder would give much better results.
* Error bars are standard deviation

**Unsupervised Learning**

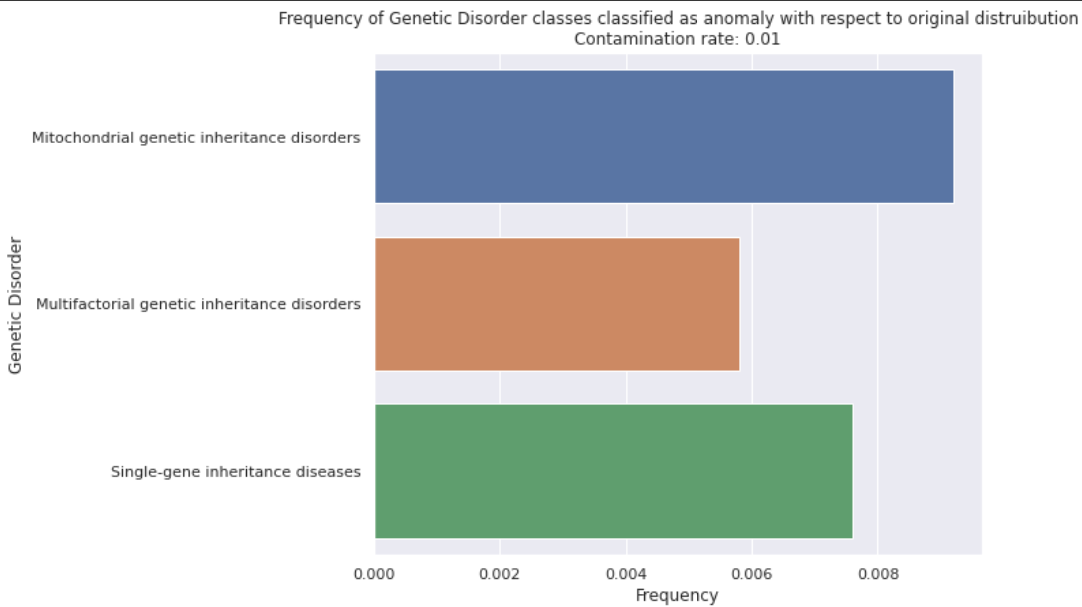
KModes

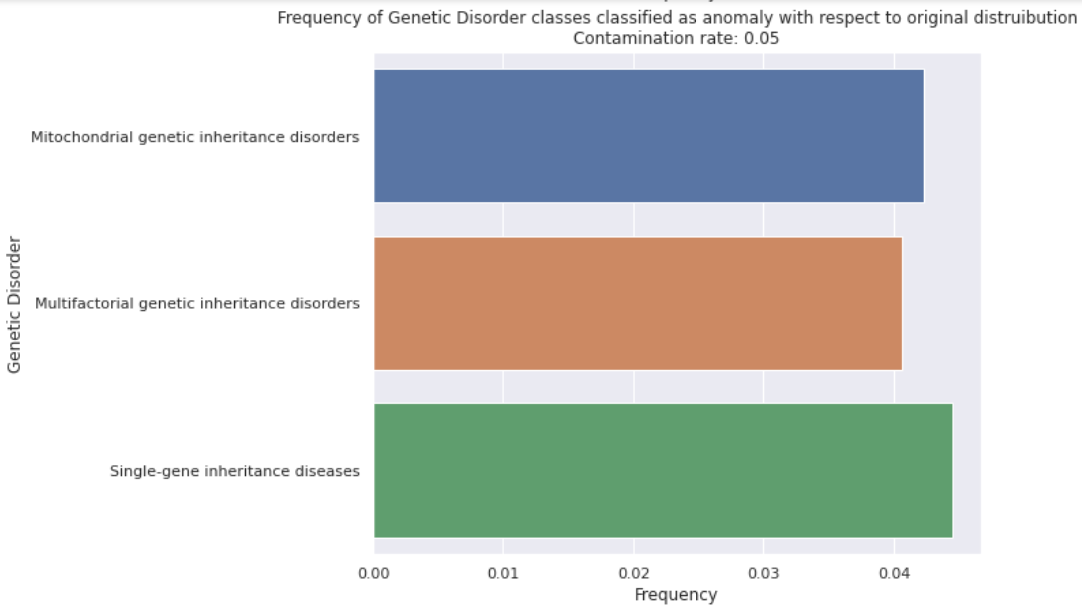
* As the predictors were mainly categorical variables to begin with and the few continuous variables were all converted to categorical variables, KMeans would no longer be an option to be used for clustering. KModes was used instead.
* Works by calculating dissimilarity between the data points.
* The optimal number of clusters, also the hyperparameter K, was determined by plotting an Elbow curve and the optimal number was chosen at the area where the curve bends.
* 6 was chosen for all KModes performed after different imbalance treatment methods. Therefore, 6 points were picked at random to label as clusters.
* Dissimilarities were then calculated and each observation was assigned to the closest cluster. New modes of clusters were redefined and the process repeats until there is no more re-assignment of points.
* After clustering with all types of imbalance treatments, all were found to be ineffective in grouping the different Genetic Disorder classes, particularly between single-gene and mitochondrial disorders.



Anomaly Detection - KNN

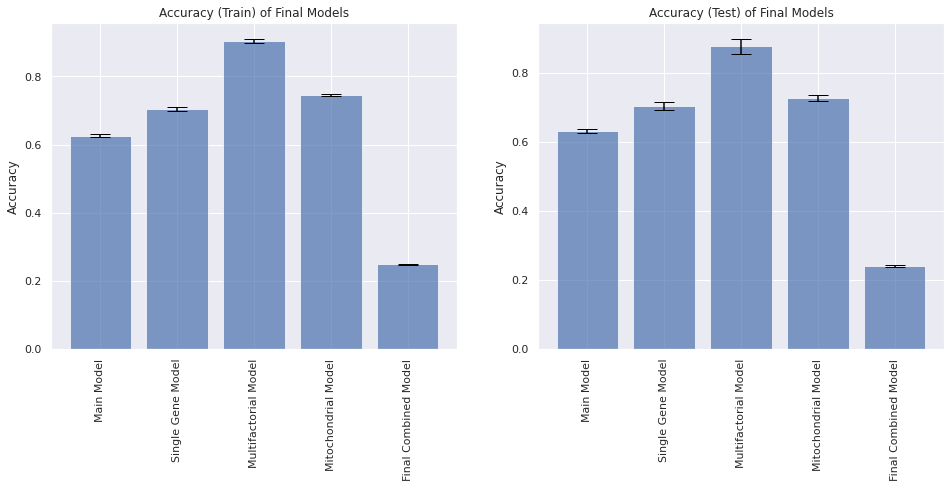
* Used the K-Nearest Neighbour model
* Anomalies of the dataset were extracted out after one-hot encoding
* Hyperparameter was set such that a data point distance to its kth nearest neighbour is viewed as the outlying score and it can be interpreted as a measure of density
* The contamination was set to change from 0.01 to 0.1
* Multifactorial genetic inheritance disorders were less likely to be flagged as outliers at low contamination rate



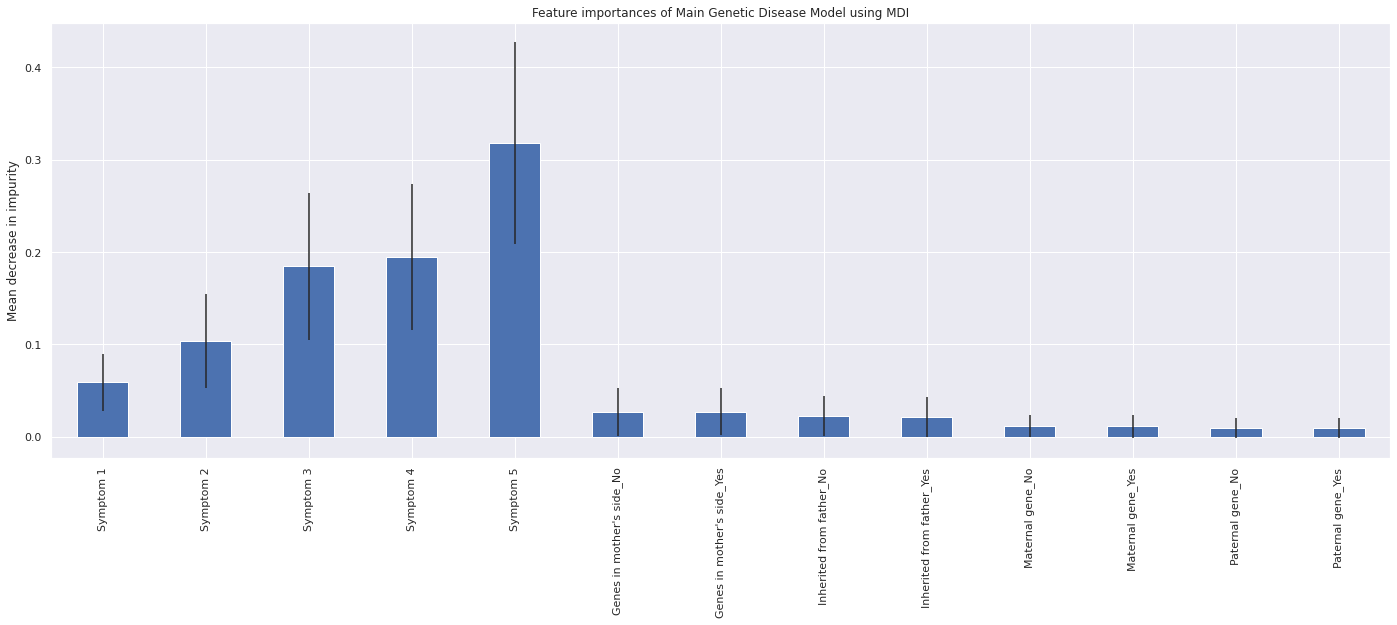


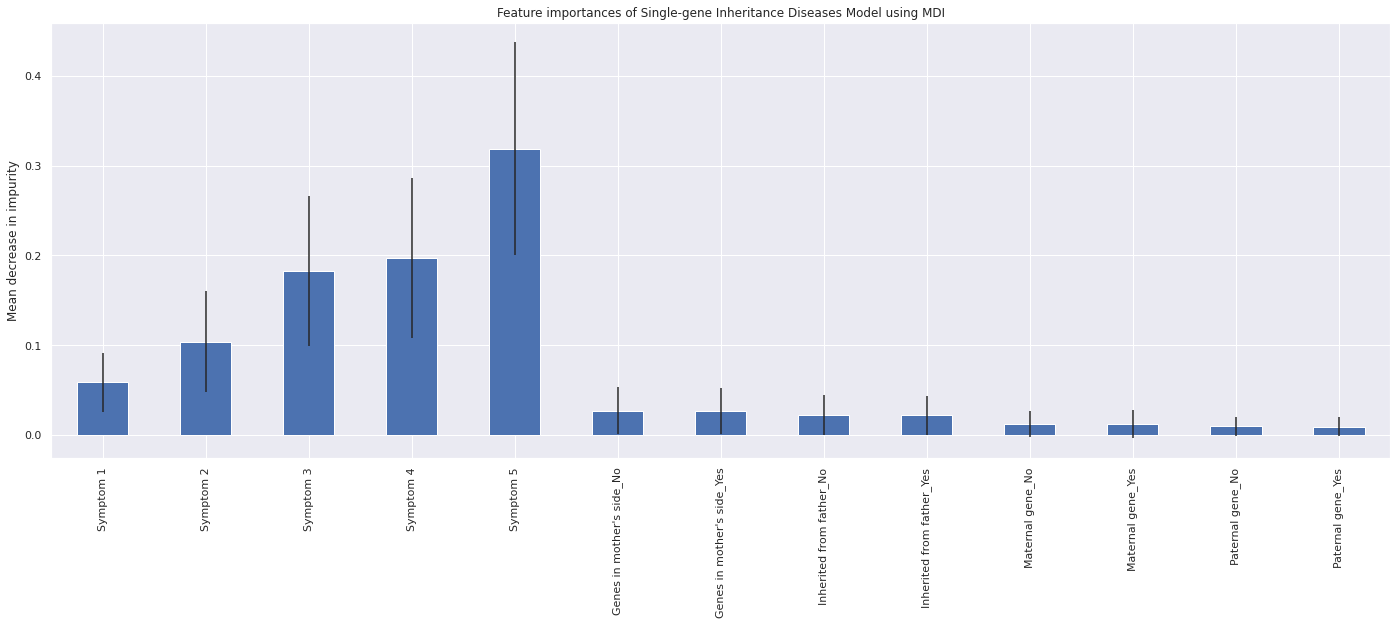
**Final Models**

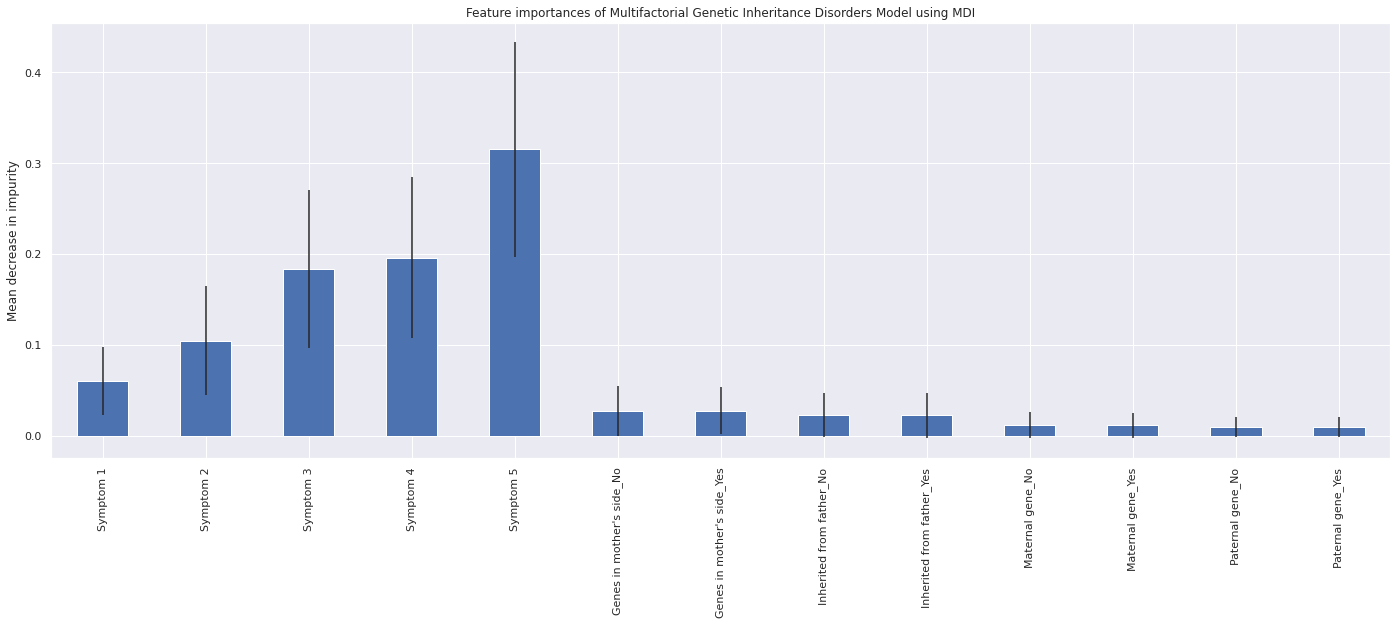
* Main model was chained to 3 sub models
* Prediction of Genetic Disorder value by the main model decides which submodel to use to predict the Disorder Subclass value
* Model to use was decided based on comparison of performance between different models

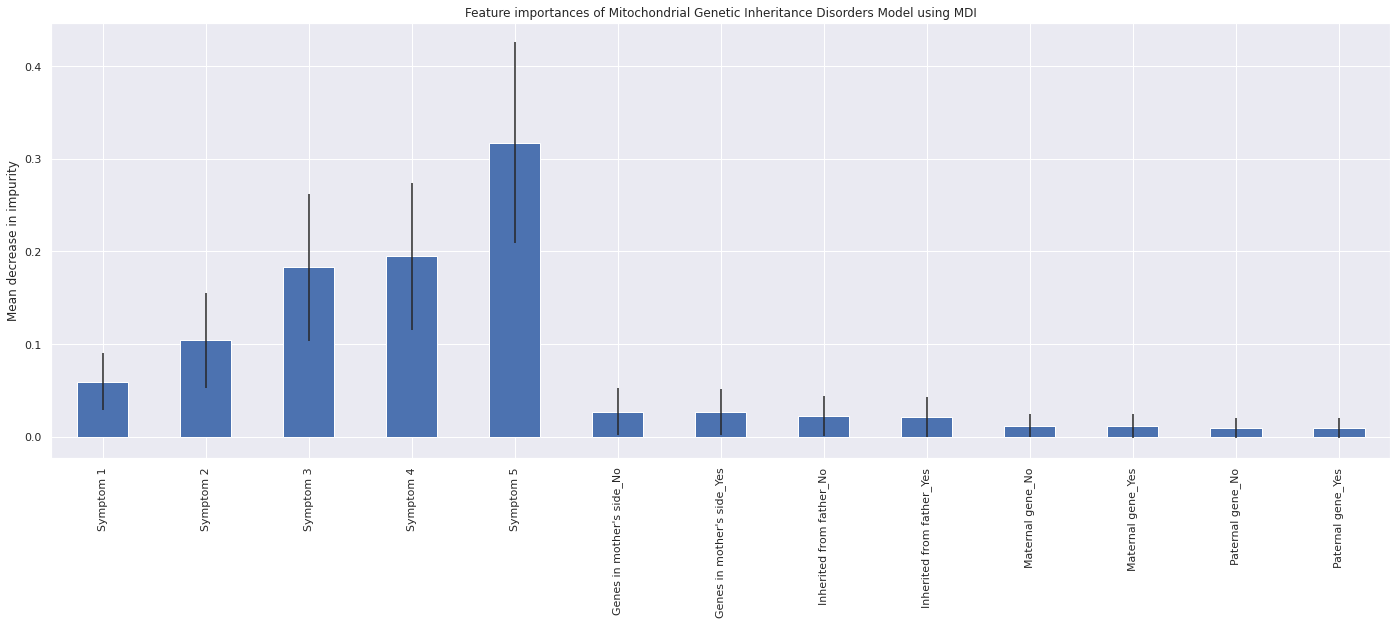


* Built the final model using SMOTE imbalance treatment and stratifying against Disorder Subclass
  + Main model had accuracy of 62.7 and 63.3 percent for the train and test set respectively, above the baseline of 33.3 percent
  + Combined model had accuracy of 24.9 and 24.3 percent for the train and test set respectively, above the baseline of 11.1 percent
* Error bars are standard deviation.









* Variable importance was similar for Genetic Disorder model and Disorder Subclass models.
* Only the genetic inheritance factors and symptoms were found to be important predictors.
  + Due to the important role of genes in the manifestation of diseases. Given that different diseases can have different symptoms, it is also essential in the identification of disease.
* Most of the variables turned out to be useless as predicted from the exploratory data analysis.
* Therefore, only useful predictors were kept to create models that are lower in complexity.
* The mean and standard deviation of accumulation of the impurity decrease for each variable within each tree of the Random Forest models was compared visually and plotted as illustrated above.
* Error bars are standard deviation.

**Conclusion**

* Symptoms and genes are not named in the original dataset description
  + Difficult to associate the exact meaning of each symptoms and genes with any of the disorders, or explain why the frequency of the symptom and gene variables deemed important for prediction was similar across all 3 disorders
* Genetic Disorder and the Disease Subclass were heavily impacted by the patients’ genes
  + Expected from a biological standpoint
* Disease may manifest different symptoms
  + Make sense that symptoms will shed light on both Genetic Disorder and the Disease Subclass.

**References**

1. https://machinelearningmastery.com/iterative-imputation-for-missing-values-in-machine-learning/
2. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
3. https://www.youtube.com/watch?v=peh2l4dePBc
4. https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/
5. https://towardsdatascience.com/introducing-anomaly-outlier-detection-in-python-with-pyod-40afcccee9ff
6. https://www.analyticsvidhya.com/blog/2021/06/kmodes-clustering-algorithm-for-categorical-data/
7. https://imbalanced-learn.org/stable/references/generated/imblearn.under\_sampling.RandomUnderSampler.html?highlight=undersample#imblearn.under\_sampling.RandomUnderSampler
8. https://imbalanced-learn.org/stable/references/generated/imblearn.over\_sampling.RandomOverSampler.html?highlight=oversample#imblearn.over\_sampling.RandomOverSampler
9. https://imbalanced-learn.org/stable/references/generated/imblearn.over\_sampling.SMOTE.html?highlight=smote#imblearn.over\_sampling.SMOTE
10. https://7-hiddenlayers.com/time-complexities-of-ml-algorithms/

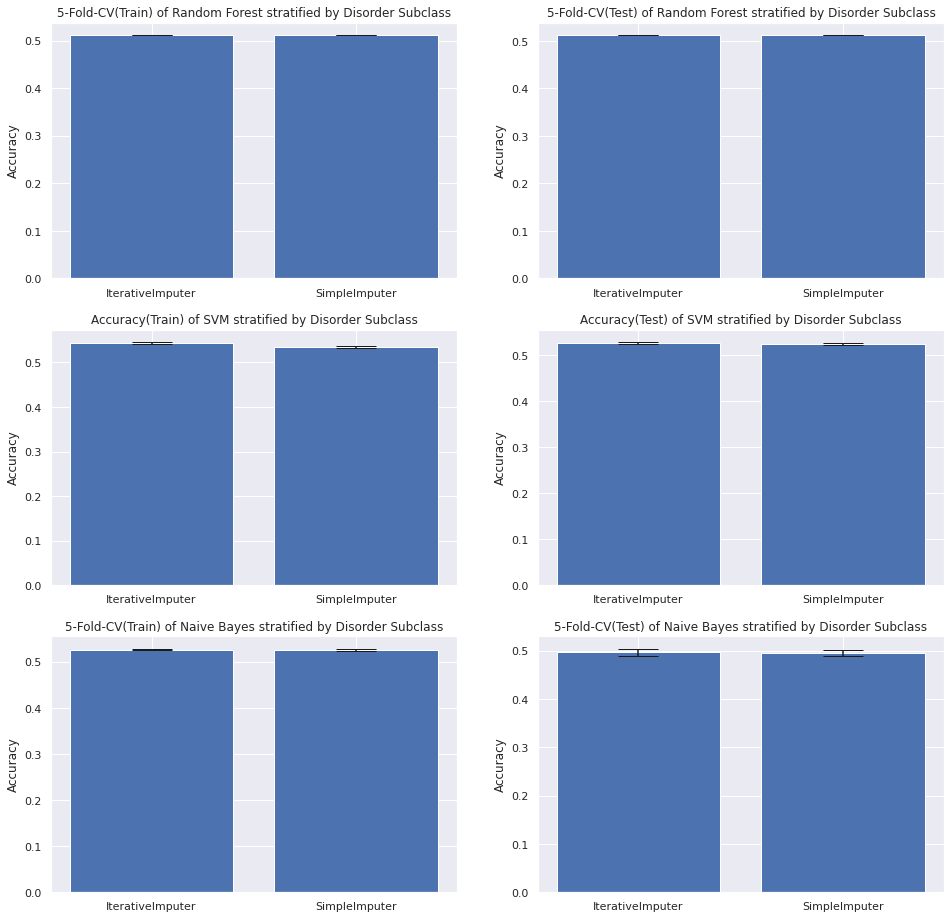
**BACKUP**

**Missing Values Treatment**

Genetic Disorder column is directly related to the Disorder Subclass, the rows with missing values in Genetic Disorder were directly filled with values based on the Disorder Subclass according to the table below.

| **Genetic Disorder** | **Disorder Subclass** |
| --- | --- |
| Single-gene | Cystic fibrosis |
| Hemochromatosis |
| Tay-Sachs |
| Multifactorial | Alzheimer’s |
| Cancer |
| Diabetes |
| Mitochondrial | Leigh syndrome |
| Mitochondrial myopathy |
| Leber’s hereditary optic neuropathy |

To fill in missing values in other predictors, both simply filling in with the mode of the predictor column using SimpleImputer and iterative imputation of missing values using IterativeImputer were tested. IterativeImputer uses the multivariate imputation by chained equations approach whereby each predictor is modelled as a function of other predictors and the missing values were predicted imputed. Naïve Bayes model was used in iterative imputation as label encoding was done initially just for imputation purpose and Naïve Bayes can ignore ordinality of data. The process repeats until the maximum number of iterations decided is reached or convergence occurs whereby there are hardly any more changes in values as compared to the previous iteration. IterativeImputer and SimpleImputer yielded similarity in this dataset when a Random Forest model is built and hence either one can be used in this dataset.

****

**Imbalance Treatment**

**Imbalance**

Performance of models created were evaluated using original imbalance data. The results serve as a control to compare with other imbalance treatment methods. The imbalance dataset was hypothesised to produce the worse performing models.

**Undersampling**

The majority classes, from Genetic Disorder or Genetic Subclass, were undersampled randomly without replacement. The sampling strategy was set to use “not minority” which resamples all classes except the minority class. One benefit of using this imbalance treatment would be the shorter computational time for model training as dataset size was reduced. However, it may risk removing important data points that should have been used for model training.

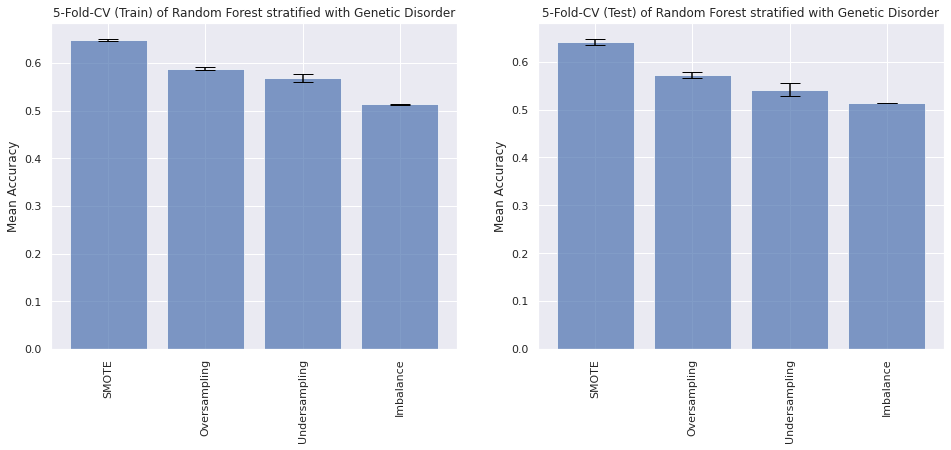
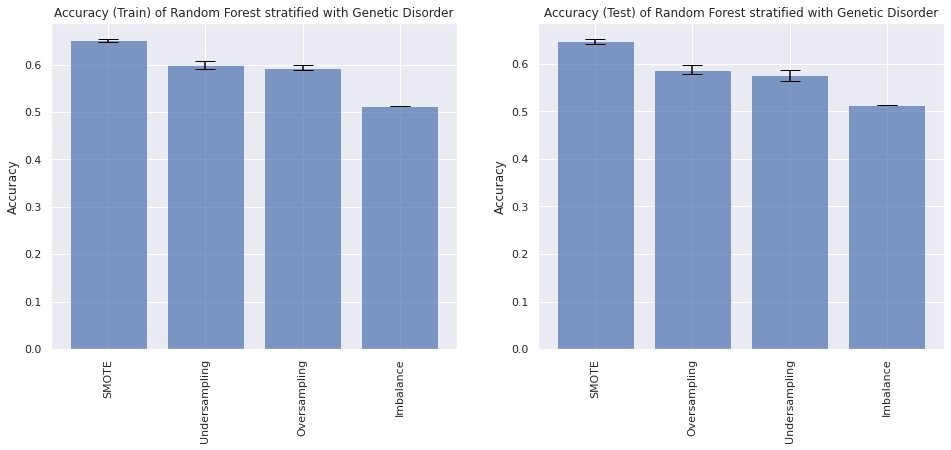
**Oversampling**

The minority classes, from Genetic Disorder or Genetic Subclass, were oversampled by picking samples at random with replacement. The sampling strategy was defaulted to use “not majority” which resample all classes but the majority class. One benefit of using this imbalance treatment would be avoiding the removal of important data points as done in undersampling. However, computational runtime was significantly increased.

**SMOTE**

Instead of oversampling minority class by duplication, SMOTE synthesize new examples from the minority class which are close in the feature space. This helps to provide additional information to the model. The sampling strategy was defaulted to use “not majority” which resample all classes but the majority class. This method was found to be the best imbalance treatment method amongst the ones tried.

**Comparison of imbalance treatment methods**



Class imbalance was noticeable in both Genetic Disorder and Disorder Subclass. Various methods of imbalance treatment were used and the results were compared to identify the best imbalance treatment for this dataset. The performance of imbalance treatment decreases in the following order: SMOTE > Oversampling > Undersampling > Imbalance. Error bars are standard deviation.

**Models Tested**

**Supervised Learning**

Support Vector Machine

The Support Vector Machine breaks the multiclass problem into multiple binary classification problems through the one-to-one approach. The model finds a hyperplane that best separates between every two classes, and is optimised by maximising the margin. However, the large genetic disorder dataset with about 20000 rows meant that training time using SVM was high so it was not used as the final model.

SVM Training time complexity = O(n3)

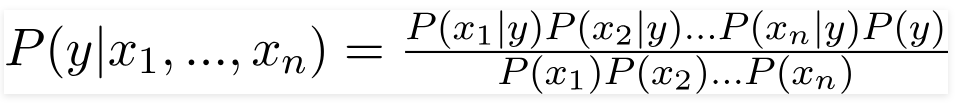
Random forest Training time complexity = O(n\*log(n)\*d\*k)

Where n = number of training sample, k = number of decision trees, d = dimensionality of the data.

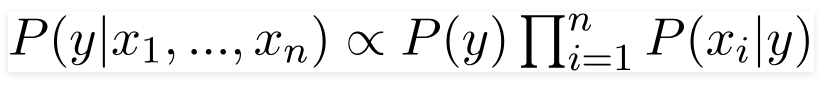
SVM was not carried out for SMOTE and Oversampling treatment on dataset with train test split stratified against Disorder Subclass due to immense runtime required.

Naive Bayes

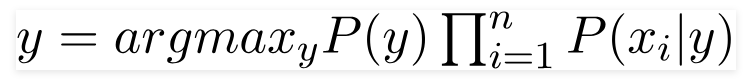
Naive Bayes is a probabilistic model which can be used in Multiclass Classification, suitable for prediction of Genetic Disorder and Disorder Subclass.



The Bayes Theorem allows the finding of probability of a Genetic Disorder class (hypothesis) happening given a certain set of predictor values (evidence). The assumption of Bayes Theorem is that the predictors are independent and that all predictors have equal effect on the outcome. Due to the denominator being a constant, it can be removed in the above equation to give the equation below.



The class of Genetic Disorder can then be assigned by finding the class with the highest probability with the given predictors.



In general, Naive Bayes did not work as well as other supervised learning models in this project and thus was not chosen to be used in the final model.

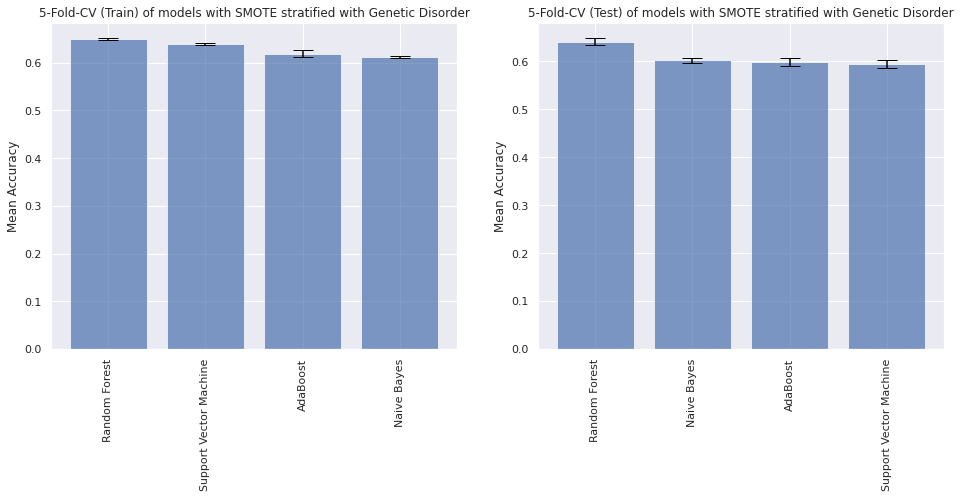
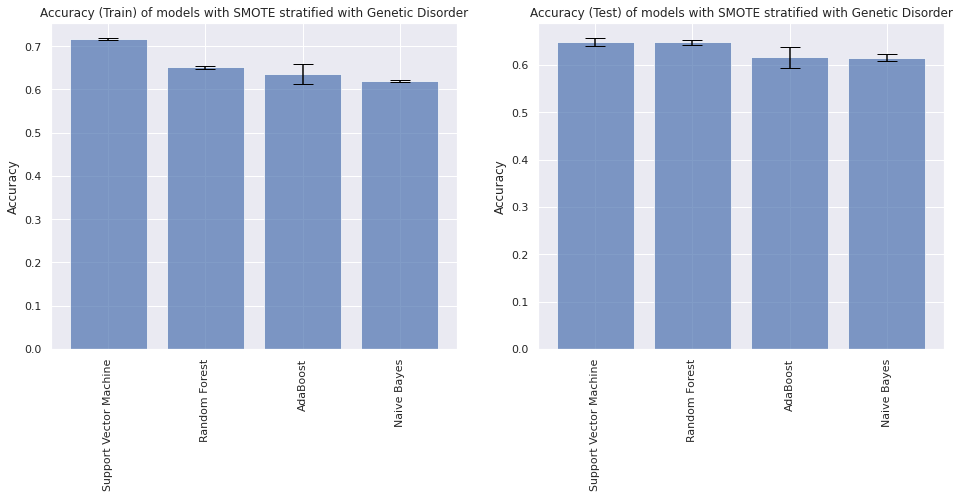
Random Forest

Random Forest utilises the ensemble technique and is suitable for a Multiclass problem. By creating a bunch of decision trees that use different variables and data points for training, collaborative learning can be achieved and the class assigned ultimately will be via a “vote” whereby the class will be the one that is predicted by most decision trees. Random Forest was found to be the best performing model out of all supervised learning models and was used for the final models.

AdaBoost

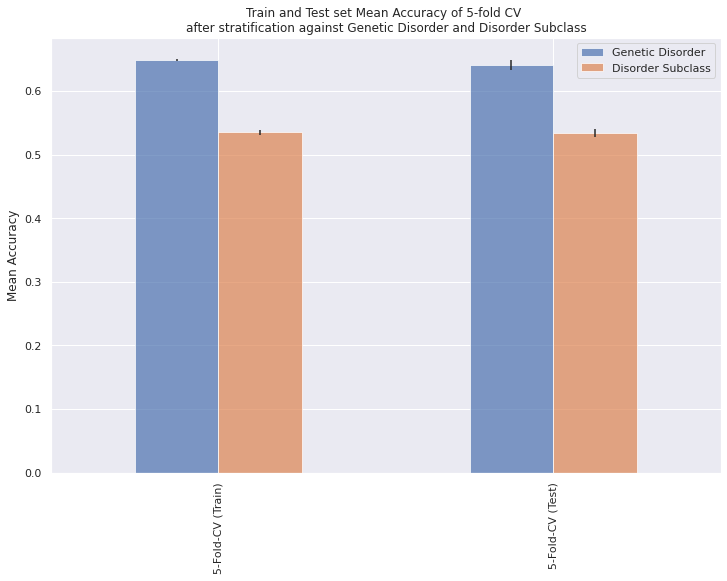
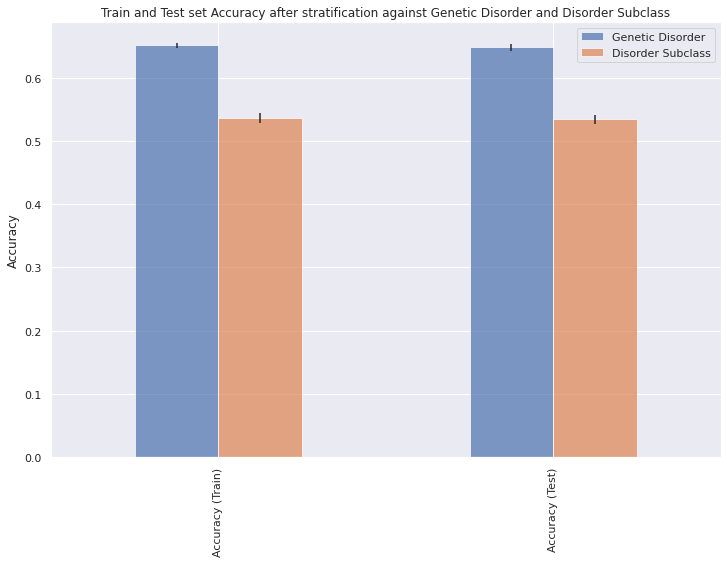
Boosting algorithm aims to improve the prediction power by converting a number of weak learners to strong learners. For AdaBoost, decision trees with 1 levels were used as the weak models known as decision stumps. A model is first built and equal weights were given to all the data points. Higher weights were then assigned to points that were wrongly classified. Points with higher weight were then given more importance in the subsequent model as they will be oversampled within the new dataset picked. The process will continue until the error is minimised. AdaBoost as a model to predict Genetic Disorder classes was found to perform better than Naive Bayes in general but underperformed in comparison to Random Forest and Support Vector Machine.

**Supervised Learning Model Performance Comparison - SMOTE and Stratified against Genetic Disorder**

****

Although the Support Vector Machine had the highest accuracy for train and test set, it was overfitted. Random Forest on the other hand, performed relatively consistently throughout for accuracy and cross validation results. Thus, Random Forest was decided to be the final model for Genetic Disorder model and Disorder Subclass models. AdaBoost was performing better than Naive Bayes most of the time but was not chosen due to poorer results when compared to Random Forest. Error bars are standard deviation.

**Random Forest Model Performance Comparison after SMOTE - Stratified against Genetic Disorder and Disorder Subclass**

****

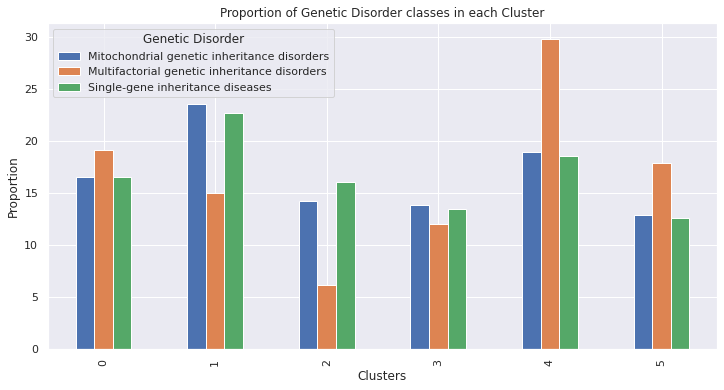
In the creation of the Genetic Disorder model, stratification against Genetic Disorder during train test split showed better performance than stratification against Disorder Subclass. As the numbers of Alzheimer’s and Cancer cases are extremely small compared to total number of observations, stratification against Disorder Subclass was done to avoid excluding these data points from the train or test sets for both main and submodels (to keep same train and test set for all models). If prediction of Genetic Disorder classes was the only problem, then stratification against Genetic Disorder would give much better results. Error bars are standard deviation.

**Unsupervised Learning**

KModes

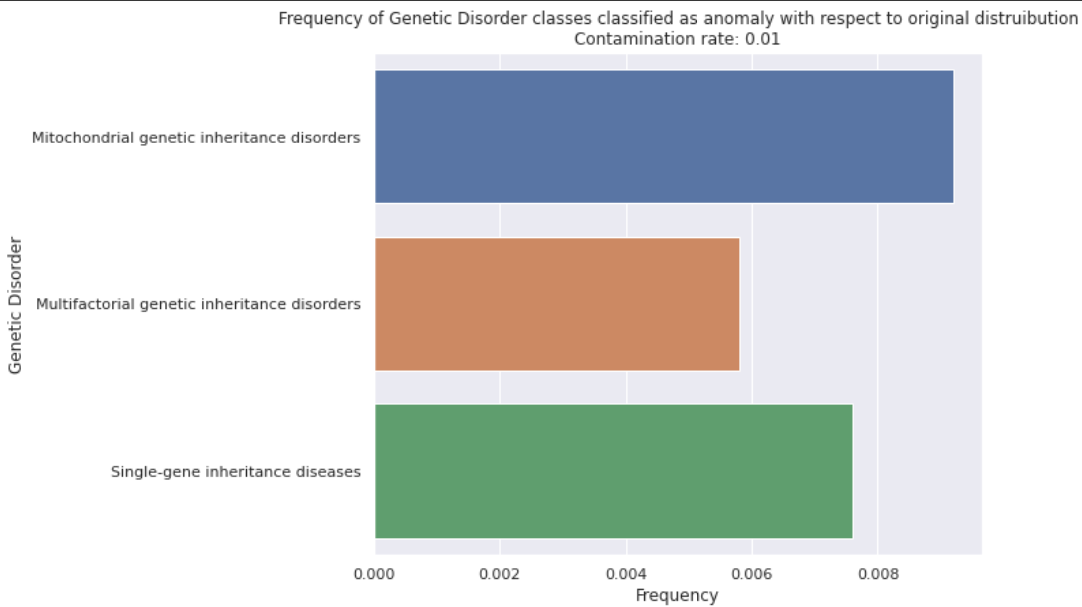
As the predictors were mainly categorical variables to begin with and the few continuous variables were all converted to categorical variables, KMeans would no longer be an option to be used for clustering. KModes was used instead. It works by calculating dissimilarity between the data points. In the project, the optimal number of clusters, also the hyperparameter K, was determined by plotting an Elbow curve and the optimal number was chosen at the area where the curve bends. 6 was chosen for all KModes performed after different imbalance treatment methods. Therefore, 6 points were picked at random to label as clusters. Dissimilarities were then calculated and each observation was assigned to the closest cluster. New modes of clusters were redefined and the process repeats until there is no more re-assignment of points.

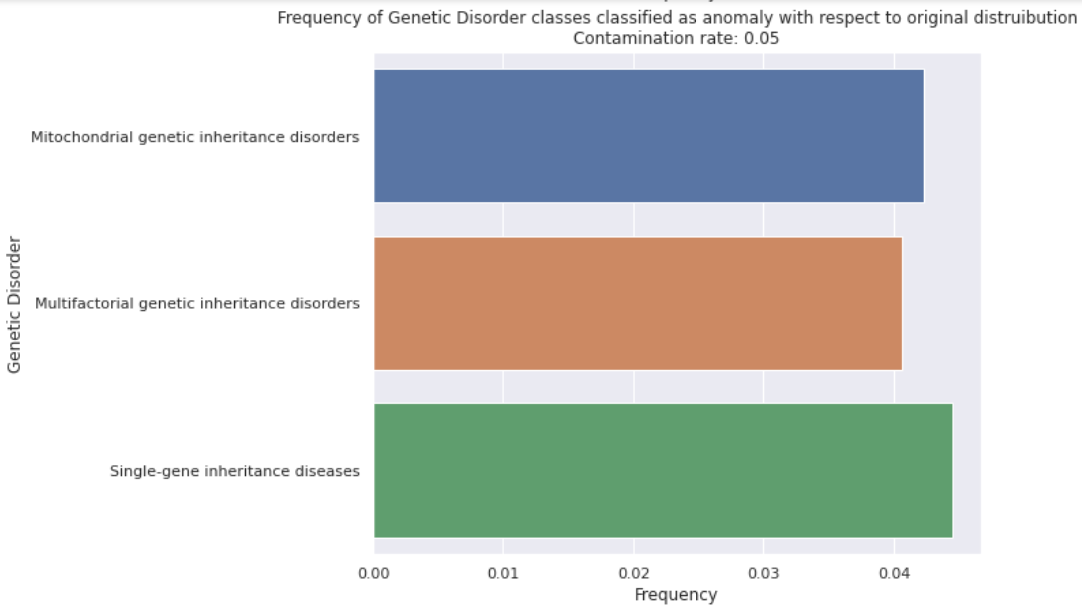
After clustering with all types of imbalance treatments, all were found to be ineffective in grouping the different Genetic Disorder classes, particularly between single-gene and mitochondrial disorders.



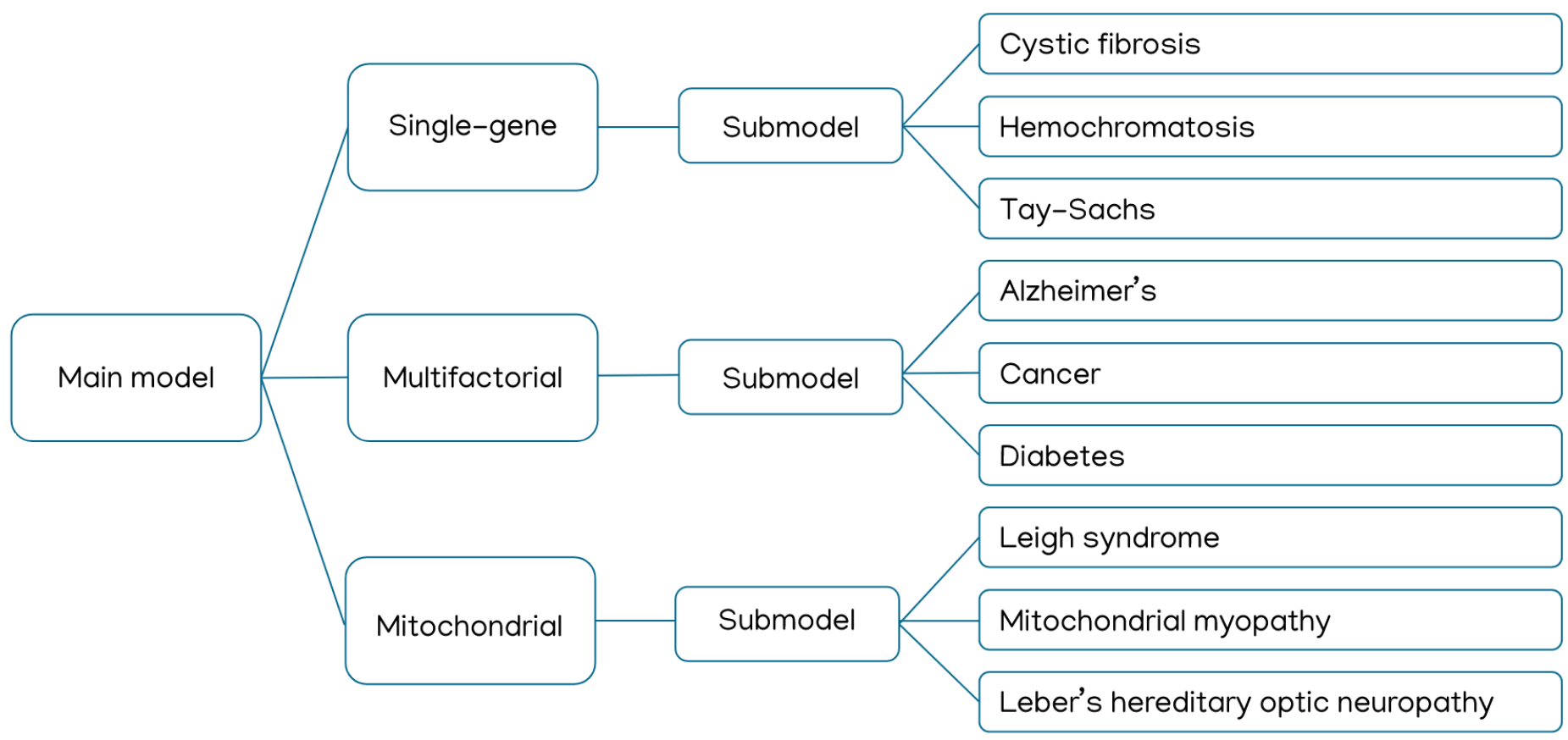
Anomaly Detection - KNN

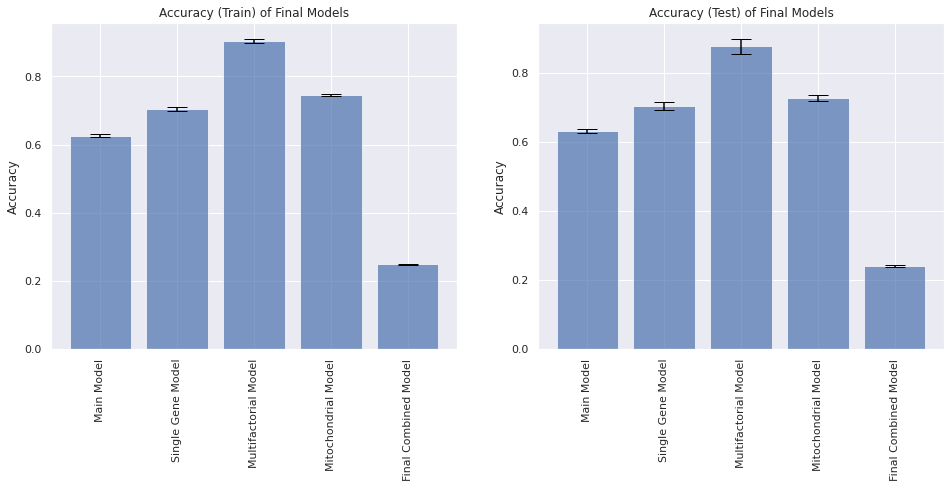
Using the K-Nearest Neighbour model, the anomalies of the dataset were extracted out after one-hot encoding. The model hyperparameter was set such that a data point distance to its kth nearest neighbour is viewed as the outlying score and it can be interpreted as a measure of density. The contamination was set to change from 0.01 to 0.1. Multifactorial genetic inheritance disorders were less likely to be flagged as outliers at low contamination rate.



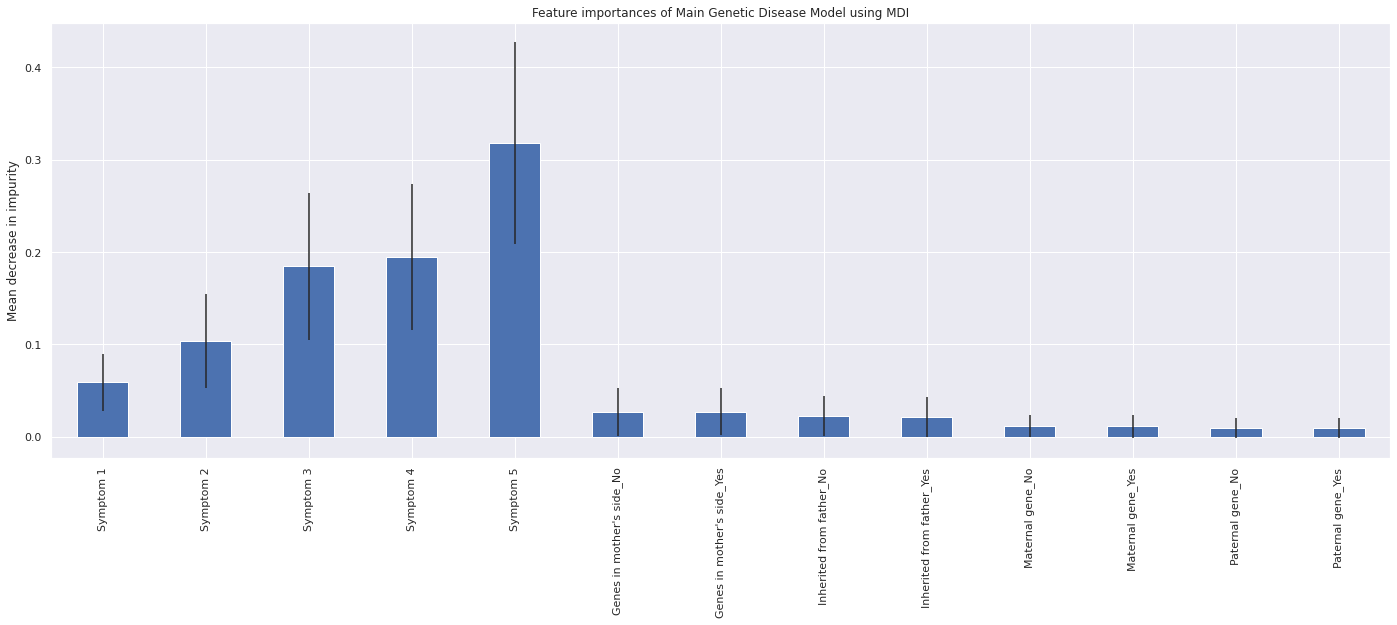


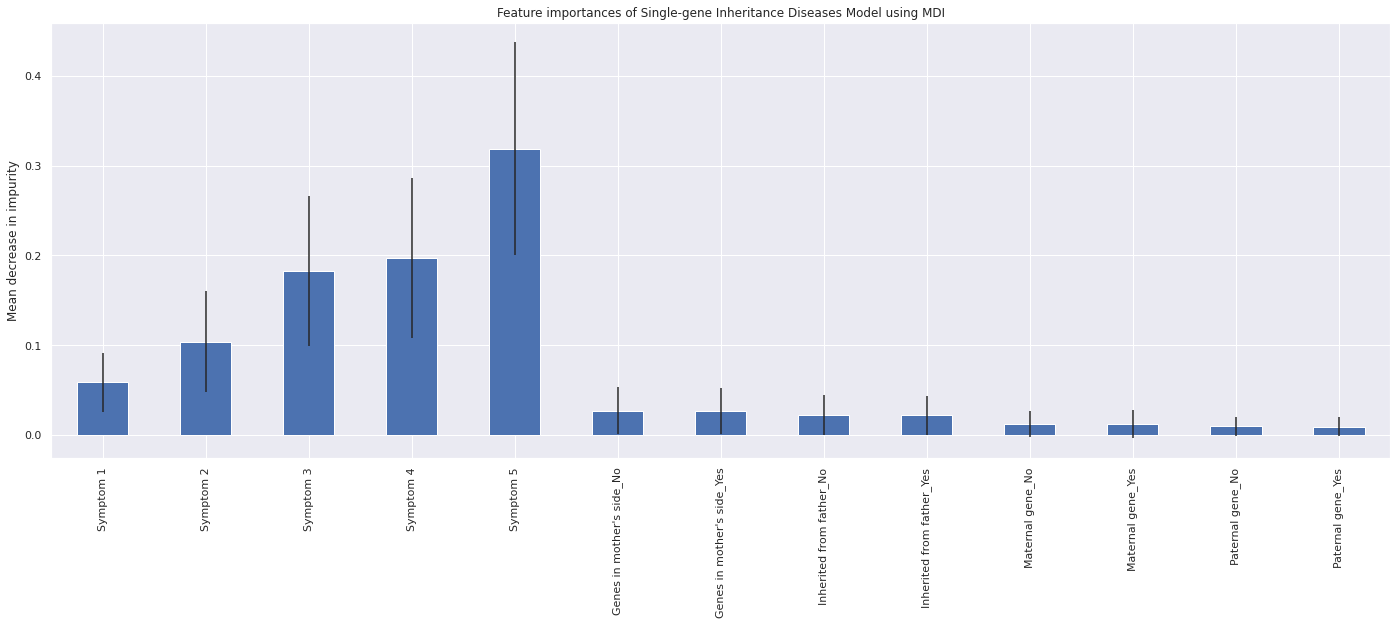
**Final Models**

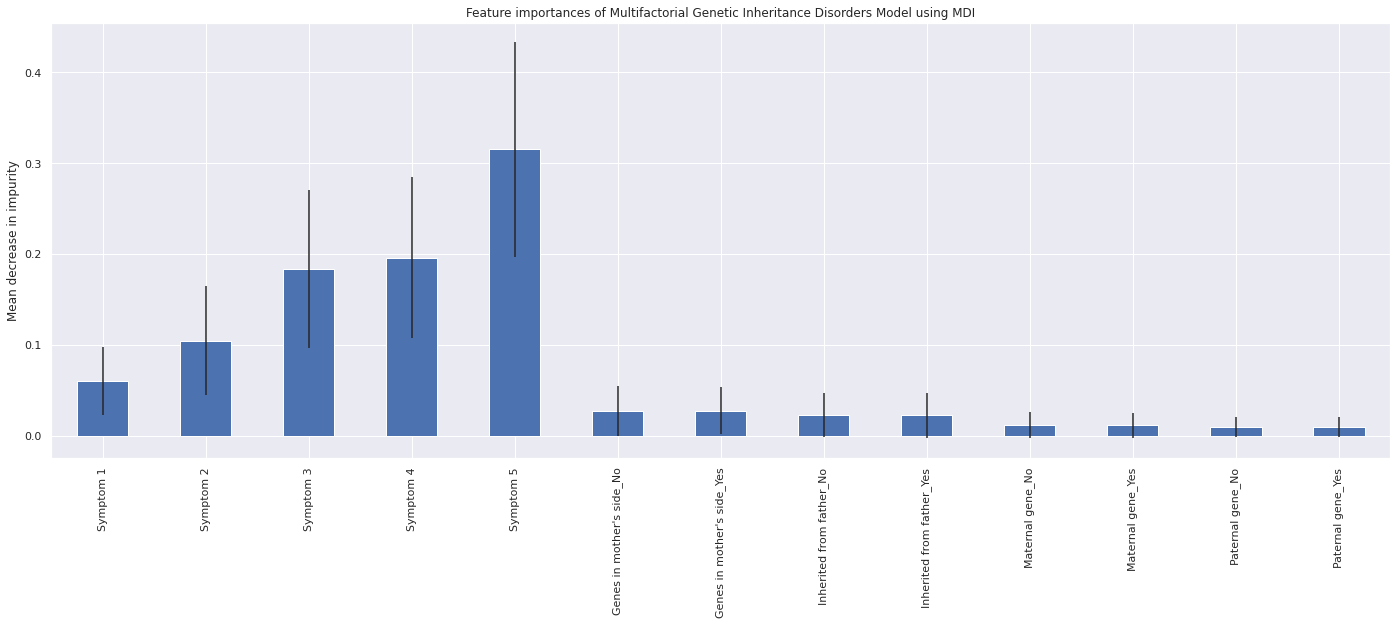
****To solve the problem, a main model was chained to 3 sub models. The prediction of Genetic Disorder value by the main model will decide which submodel to use to predict the Disorder Subclass value. The model to use was decided based on comparison of performance between different models.

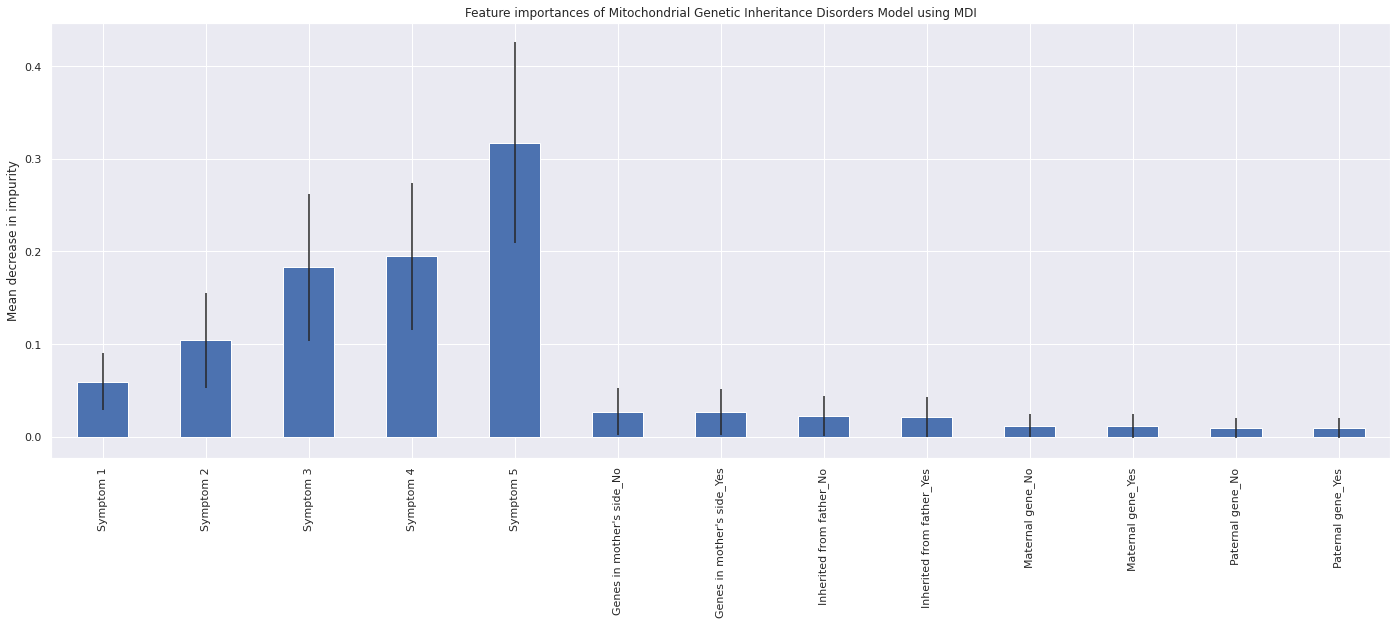


After building the final model using SMOTE imbalance treatment and stratifying against Disorder Subclass, the main model had accuracy of 62.7 and 63.3 percent for train and test set respectively, above the baseline of 33.3 percent. For the combined model, it had accuracy of 24.9 and 24.3 percent for train and test set respectively, above the baseline of 11.1 percent. Error bars are standard deviation.









Variable importance was found to be similar for Genetic Disorder model and Disorder Subclass models. Most of the variables turned out to be useless as predicted from the exploratory data analysis. Therefore, only useful predictors were kept to create models that are lower in complexity. The mean and standard deviation of accumulation of the impurity decrease for each variable within each tree of the Random Forest models was compared visually and plotted as illustrated above. Error bars are standard deviation.

Only the genetic inheritance factors and symptoms were found to be important predictors for the Genetic and Disorder Subclass. This is due to the important role of genes in the manifestation of diseases. Given that different diseases can have different symptoms, it is also essential in the identification of disease.

**Conclusion**

Given that the symptoms and genes are not named in the original dataset description, it is difficult to associate the exact meaning of each symptoms and genes with any of the disorders, or explain why the frequency of the symptom and gene variables deemed important for prediction was similar across all 3 disorders. However, the Genetic Disorder and the Disease Subclass were indeed heavily impacted by the patients’ genes which was also expected from a biological standpoint. As disease may manifest different symptoms, it also make sense that symptoms will shed light on both Genetic Disorder and the Disease Subclass.