Disease Recurrence Analysis

Abdelrehim Sabri, Abdelrehim.sabri@ryerson.ca (500998232)

Summary

The project analyzed a dataset for patients' demographic contain standard information regarding individuals from a variety of ancestral lines. The study aims at understating the patient profiles, from age, work style, and ancestry, to predict the type of patient and disease the hospital is getting to be able to hire the proper staff, and account for the medications the hospital should be equipped with for diagnosis and treatment.

The data was preprocessed primarily and balanced using Under and Over Sampling, and SMOTE and further discretized to enable the Machine Learning algorithms to perform better. Numerous ML algorithms were used, and results of Random Forest, Naive Bayes and Logistic Regression were tabulated and compared based on parameters such as Accuracy, Error Rate and Positive rate. Random Forest was identified as the best ML algorithm.

Introduction

Every ethnic group has a different diet and lifestyle. When people moved to North America, they brought their eating habits, level of education, and lifestyle with them. The large size and highly diverse population resulted in different types of diseases that effected different age and ethnic groups.

“Diagnosis has important implications for patient care. When a diagnosis is accurate and made in a timely manner, a patient has the best opportunity for positive health outcome because clinical decision making well be tailored to a correct understanding of the patient's health problem.” [1]

The research question is to define what factors are relevant to predicting the disease the patient might have. Also, who might be susceptible to which type of disease. The dataset is based on people living in United States.

After exploring the data, there is a possibility that a patient might have multiple diseases. Therefore, each instance can be assigned with multiple categories, as such, this type of problem needs to use multi-label classification.

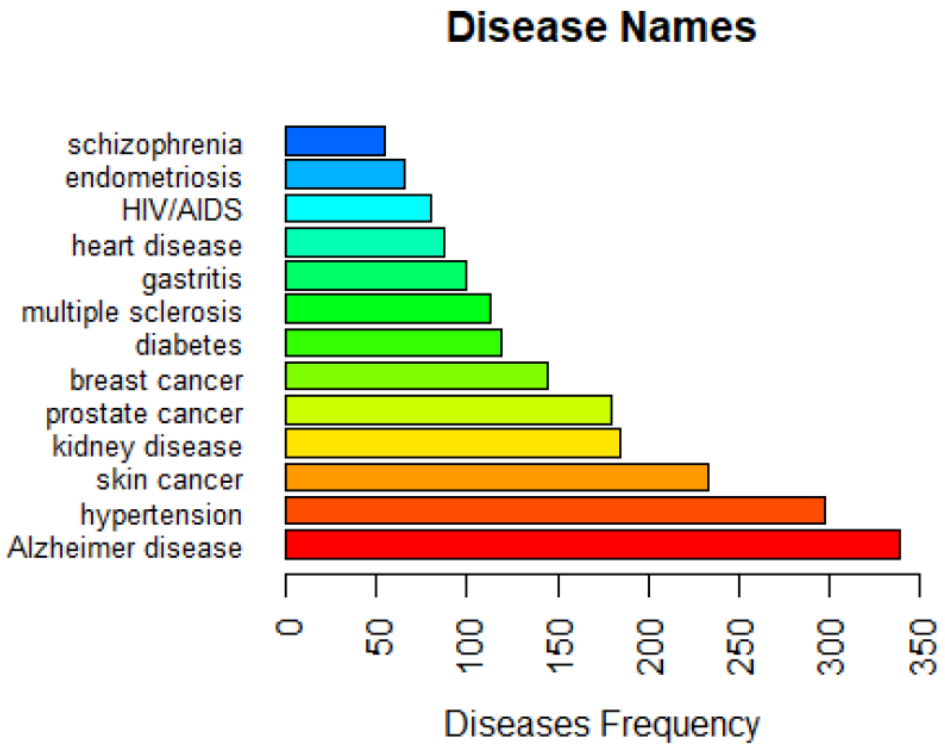
1. Data Preparation

In this section the patients' data set is explained. Some anomalies like an imbalanced data problem are described and solved.

The data used for this project can be found at
https://www.kaggle.com/karimnahas/medicaldata.

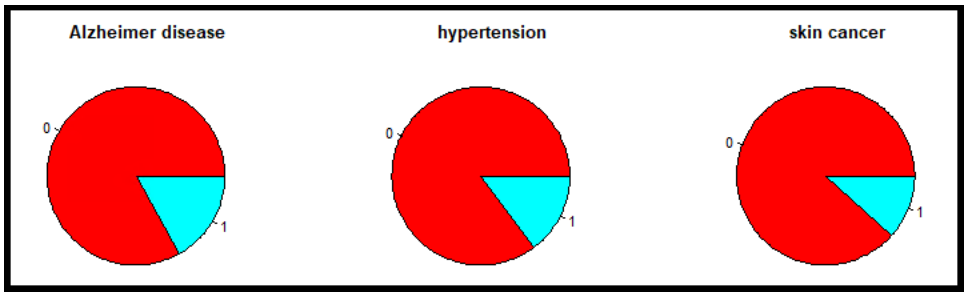
1.1 Description of the Patients Data

The project uses the patients' data. Each patient is diagnosed with a disease. There are 13 diseases listed in the dataset. For the simplification of the project, 13 new attributes were created for the 13 diseases in the dataset, to make data a binary classification problem, where individuals are classified as has a disease or does not have that disease. The dataset contains 14 categorical/nominal variables (not including the class variable). The number of instances is 2000 and variables are coming from different hospitals. In this study, will only study the top three diseases (Alzheimer, hypertension, and skin cancer).



A problematic characteristic of this data is the presence of a class imbalanced problem as shown in the graph below.

Figure 1: Piechart of the class (each disease)



0: does not have the disease, 1: has the disease

Imbalanced datasets are one in which the majority case greatly outweighs the minority case. In this case, the Figure 1 above, shows the number of instances to be not diagnosed is around 85% and it is around 15% for diagnosed. The Figures 2, 3, and 4 below show the partition of all the variables of the patient data depending on the class variable.

Figure 2: Classification of the class (Alzheimer Disease) among the attributes of the original patient data

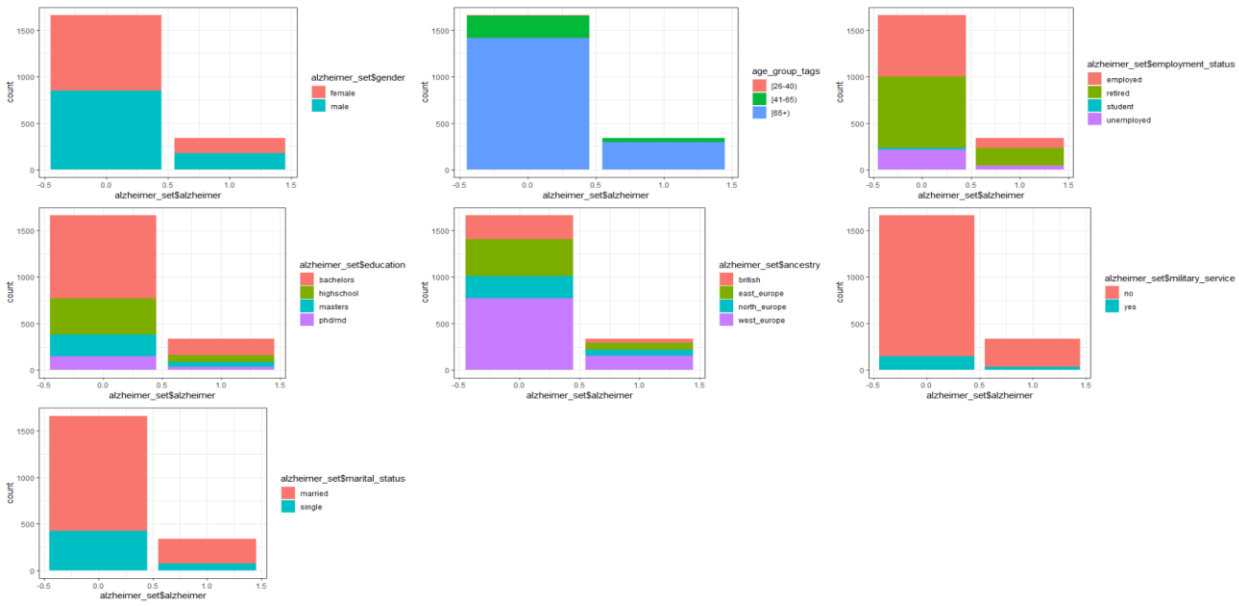


Figure 3: Classification of the class (Hypertension Disease) among the attributes of the original patient data

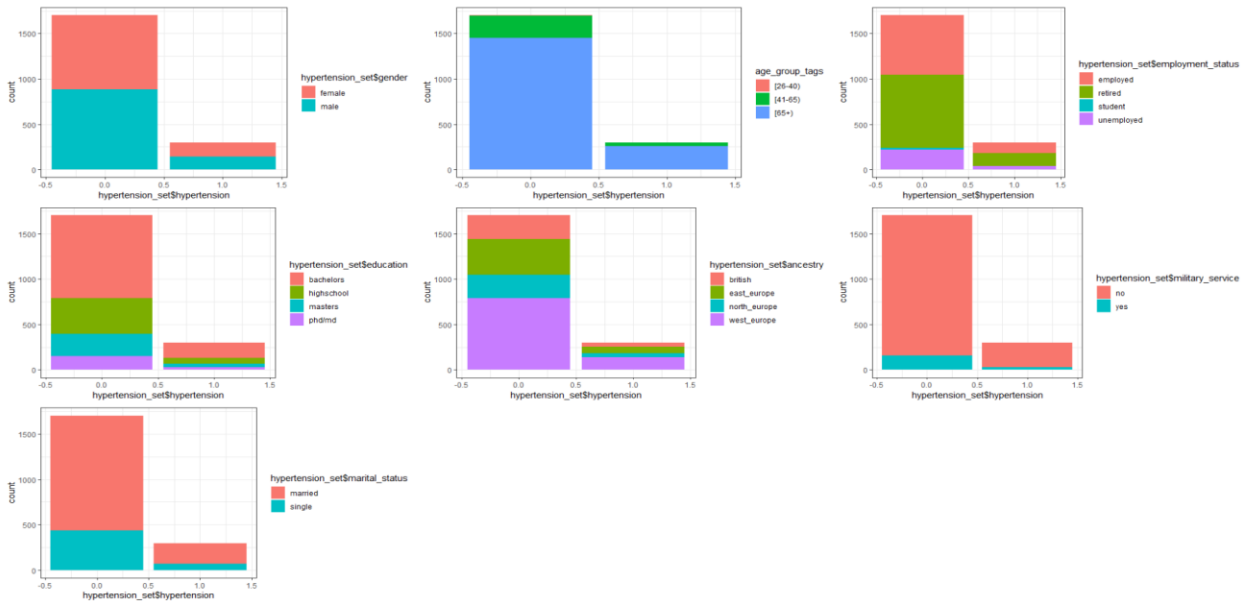
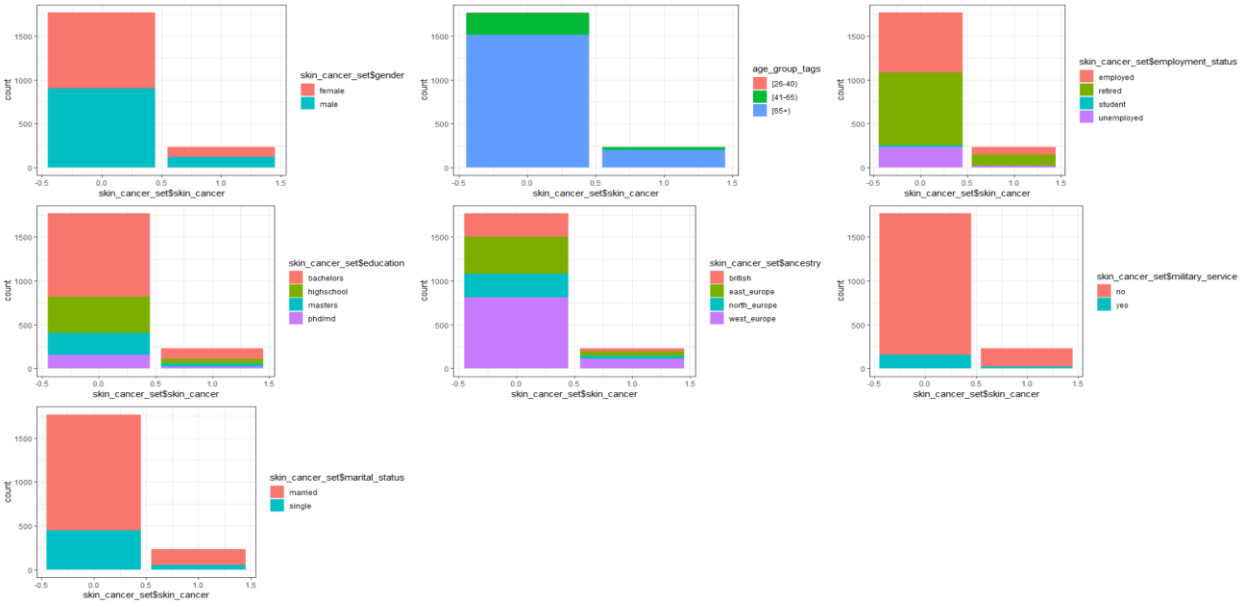


Figure 4: Classification of the class (Skin Cancer Disease) among the attributes of the original patient data



Work with this kind of data without solving the problem of the imbalanced data lead to bias and the results become inclined towards the majority class and impacts the decisions taken by data scientists.

In the section “Imbalanced data problem and its resolution”, will describe some techniques and algorithms used to resolve the problem of the imbalanced data.

The table 1 below shows the description of the attributes:

Table 1: Description of the attributes of patients' data

#	Name of the Attribute	Role	Datatype	Values of the attribute
1	Disease	Target	categorical	prostate cancer
				skin cancer
				breast cancer
				HIV/AIDS
				diabetes
				heart disease
				hypertension
				endometriosis
				multiple sclerosis
				schizophrenia
				kidney disease
				gastritis
				Alzheimer disease
2	Gender	Input	categorical	Male
				Female
3	DOB	Input	quantitative	
4	Zip Code	Input	categorical	quantitative
5	Employment_statuts	Input	categorical	Employed
				Retired
				Student
				Unemployed
6	Education	Input	categorical	bachelors
				highschool
				masters
				phd/md
7	Marital_status	Input	categorical	married
				Single
8	children	Input	quantitative	
9	Ancestry	Input	categorical	british
				east_europe
				north_europe
				west_europe
10	Avg_commute	Input	quantitative	
11	Daily_internet_use	Input	quantitative	
12	Available_vehicles	Input	quantitative	
13	military_service	Input	categorical	No
				Yes

For the numerical variables, will do their descriptive statistics and their correlation matrix. For the categorical/nominal variables, will compute the correlation between those variables by

computing their Cramer's V values. The Cramer's V measures the strength of the association between one ordinal/nominal /categorical variable with either another ordinal/nominal/categorical variable.

1.2 Descriptive statistics and correlation the variables of the patient data

For the numerical variables, will do their descriptive statistics and their correlation matrix. For the categorical/nominal variables, will compute the correlation between those variables by computing their Cramer's V values.

1.2.1 Descriptive statistics and correlation matrix of the quantitative attributes of the patient data

The table 2 describes the descriptive statistics of the quantitative variables of the patient data.

Table 2: Descriptive statistics of numerical variables of patients' data

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Children	2000	0	7	2.27	1.616	2.611
Avg_commute	2000	-2.47	63.73	30.38	10.027	10.027
Daily_internet_use	2000	1.01	8.82	4.99	1.388	1.926
Available_vehicles	2000	0	4	1.75	1.119	1.252

There are no missing values for the quantitative variables of the patient data. So, there is no problem of missing values with the patient data apart its issue from the imbalanced data problem.

1.2.3 Discretization of the continuous quantitative attributes: Age and Ancestry

Given those variables are not divided into classes, they will be discretizing, and enabling it to be used together with the others, with the same methods. This will provide greater simplicity and readability.
There is a threshold at 65 years. No other threshold is clearly evident. It is decided to have a first band for people where Age is lower than 25 and one band for the senior people (Age

greater than 65) but, made sure to have enough data in these bands. After, divided in 3 bands with at equal size the 'Age' between 25 and 65 years old. Finally, got five bands:

Bin	Class 'age'
0-25	0
26-40	25
41-50	40
50-65	50
Age > 65	65

Table 3: Age categories

For the ancestry, the number of countries is large, since all the countries are from Europe, the countries were group based on European ethnicities.
Finally, made the ancestry group into four have:

Bin	Class 'ancestry'
Ukraine, Russia, Poland, Czech Republic, Hungary	east_europe
Austria, Belgium, France, Germany, Italy, Netherlands, Portugal, Spain, Switzerland	west_europe
Sweden, Finland, Denmark	north_europe
England, Scotland, Ireland	British

Table 4: Ancestry categories

ss

1.2.4 Cramer's V correlation of the nominal attributes of the patients' data

The Cramer's V measures the strength of the association between two nominal values, giving a value between 0 and +1. It based on Pearson's chi-squared statistic. Cramer's V is computed by taking the squared root of the chi-squared statistic divided by the sample size and the minimum dimension minus 1: $V = \sqrt{\frac{\chi^2/_n}{min(k-1,r-1)}}$ where: χ^2 is derived from Pearson's chi-squared test (It is the chi square statistic), *n* is the grand total of observations, *k* being the number of columns and *r* being the number of rows.

The p-value for the significance of V is the same one that is calculated using the Pearson's chi-squared test.

The chi square statistic (χ^2) is defined as $\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$ where O_i is the observed number of cases in category i, and E_i is the expected number of cases in category i .

In mathematical terms, the χ^2 distribution with d degrees of freedom, continuous variable, is the sum of the squares of d normally distributed variables.

1.2.5 User's guide to Association and correlation coefficients

Table 5: User's guide for Pearson correlation r

User's guide for Pearson correlation r	
Size of Correlation (r)	**Interpretation**
.90 to 1.00 (−.90 to −1.00)	Very high positive (negative) correlation
.70 to .90 (−.70 to −.90)	High positive (negative) correlation
.50 to .70 (−.50 to −.70)	Moderate positive (negative) correlation
.30 to .50 (−.30 to −.50)	Low positive (negative) correlation
.00 to .30 (.00 to −.30)	negligible correlation

Table 6: User's guide for Cramer's V association v

User's guide for Cramer's V association v		
LEVEL OF ASSOCIATION (v)	**Verbal Description**	**COMMENTS**
0.00	No Relationship	Knowing the independent variable does not reduce the number of errors in predicting the dependent variable at all.
.00 to .15	Not generally useful	Not acceptable
.10 to .20	**Weak**	**Minimally acceptable**
.20 to .25	**Moderate**	**Acceptable**
.25 to .30	**Moderately Strong**	
.30 to .35	**Strong**	
.35 to .40	**Very Strong**	
.40 to .45	**Worrisomely Strong**	**Either an extremely good relationship or the two variables are measuring the same concept**
.45 to .99	Redundant	The two variables are probably measuring the same concept.
1.00	Perfect Relationship.	If the independent variable is known, the dependent variable can be perfectly predicted.

1.2.4 Results and Interpretations

The tables 7, 8, 9 shows the results between the variables and the target of patients' data.

It seems that the association between the most qualitative variables are either not generally useful (the Cramer's values are between 0.0 and 0.15) or redundant (the Cramer's values are between 0.45 and 0.99). So, no use to do filter selection when running the models.

Table 7: Calculating the Cramer V attributes against Alzheimer disease

Variables	Cramer's V Measure
gender	0.0030
education	0.0191
military_service	0.0254
marital_status	0.0510
available_vehicles	0.0520
age	0.0610
ancestry	0.0720
children	0.0733
employmnet_status	0.1061
zipcode	0.1204
daily_internet_use	0.6967
avg_commute	0.9268

Table 8: Calculating the Cramer V attributes against Hypertension disease

Variables	Cramer's V Measure
military_service	0.0031
employmnet_status	0.0155
marital_status	0.0191

education	0.0207
available_vehicles	0.0262
ancestry	0.0294
gender	0.0429
age	0.1028
children	0.1042
zipcode	0.1151
daily_internet_use	0.6867
avg_commute	0.9367

Table 9: Calculating the Cramer V attributes against Skin Cancer disease

Variables	Cramer's V Measure
gender	0.0011
military_service	0.0126
education	0.0175
marital_status	0.0217
ancestry	0.0312
age	0.0777
zipcode	0.0849
available_vehicles	0.0949
employmnet_status	0.1082
children	0.1133
daily_internet_use	0.7354

avg_commute	0.9498

1.3 Imbalanced data problem of patients' data and its resolution

It already mentioned the problem of imbalanced patients' data in the section 1.1 (description of patients' data). The present section describes the imbalanced problem of data and defines some technics and algorithms to solve the imbalanced problem of data. Particularly, will present three algorithms that will use to resolve the imbalanced problem of patients' data.

1.3.1 Imbalanced data problem

Classical machine learning algorithms assume that the number of objects in considered classes is roughly similar. However, in many real-life situations the distribution of examples is skewed since representatives of some of classes appear much more frequently. The classes with the larger number of instances are called majority classes and the classes with the smaller number of instances are referred to as the minority classes. That situation is the imbalanced data problem. This poses a difficulty for learning algorithms, as they will be biased towards the majority group. Intuitively, since there are large number of majority class examples, a classification model tends to favor majority classes while incorrectly classifying the examples from the minority classes. In, real-world, will face the imbalanced data problem in many data independently the fields of research. Canonically, the data imbalance problem is classified in two categories: a binary imbalanced data problem (binary class) and a multi-class imbalanced data problem (multi-class).

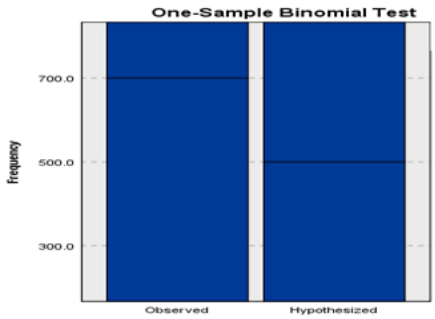
Will use the One-Sample Binomial Test to confirm our observation concerning the presence of imbalanced class. A One-Sample Binomial Test is used for testing whether a proportion from a single dichotomous variable is equal to a presumed population value.

The table 10 below summarizes our imbalanced patients' dataset test. Below will focus on the Alzheimer disease example. But all other diseases will follow the same solution scenarios.

Table 10: A One -Sample Binomial Test

Imbalanced data Test: A One-Sample Binomial Test				
Hypothesis Test Summary				
	Null Hypothesis	Test	Sig.	Decision
1	The categories defined by have Alzheimer = 1 and 0 occur with probabilities .500 and .500.	One-Sample Binomial Test	.000	Reject the null hypothesis.
Asymptotic significances are displayed. The significance level is .050.				

Figure 5: One-Sample Binomial Test



1.3.2 Approach to handling Imbalanced Datasets: Resampling techniques

Learning from imbalanced data has been studied actively for about two decades in machine learning. A Data scientist facing this problem for the first time often ask, 'What should I do when my data is imbalanced?'. According to Fawcett, this has no definite answer for the same reason that the general question Which learning algorithm is best? has no definite answer: it depends on the data.

Will focus this part on resampling techniques because a resampling technique is a part of a data preprocessing to deal with the imbalanced data problem. Indeed, the data preprocessing plays a big role in the identification of class overlapping and label noise. Therefore, proper data cleaning and sampling procedures that consider the varying characteristics of classes and balanced performance on all of them must be proposed Krawczyk (2016). Algorithmic ensemble techniques are more a part of a modeling algorithms.

A vast number of techniques has been tried, with varying results and few clear answers. The main objective of balancing classes is to either increasing the frequency of the minority class or decreasing the frequency of the majority class. This is done in order to obtain approximately the same number of instances for both the classes. Here, are listed some techniques.

1.3.2.1 Naïve/Easy Approach: Do Nothing

The naïve approach consists to do nothing. Sometimes you get lucky and nothing needs to be done and sometimes no. You can train on the so-called natural (or stratified) data and sometimes it works without need for modification.

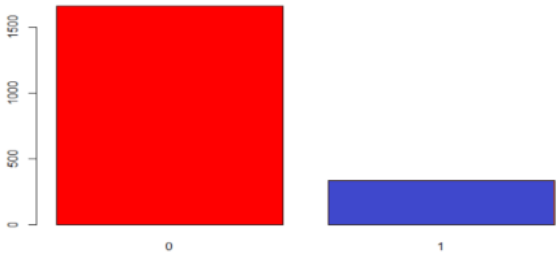
1.3.2.2 Under-sampling

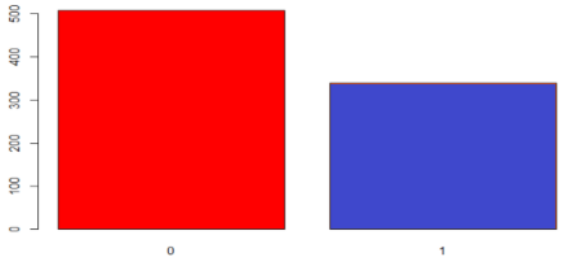
Random Under-sampling aims to balance class distribution by randomly eliminating majority class examples. This is done until the majority and minority class instances are balanced out.

- **Advantages**
  - It can help improve run time and storage problems by reducing the number of training data samples when the training data set is huge

- **Disadvantages**
  - It can discard potentially useful information which could be important for building rule classifiers
  - The sample chosen by random under sampling may be a biased sample. And it will not be an accurate representative of the population. Thereby, resulting in inaccurate results with the actual test data set

See below the result after using under-sampling on the patients' data.





BEFORE USING UNDER-SAMPLING	AFTER USING UNDER-SAMPLING

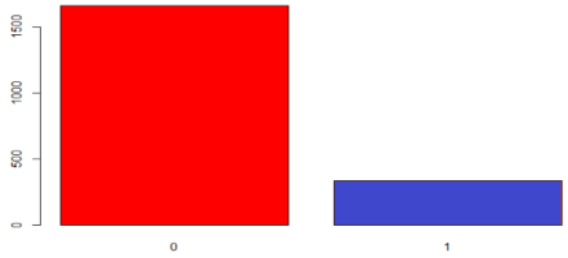
1.3.2.3 Over-sampling

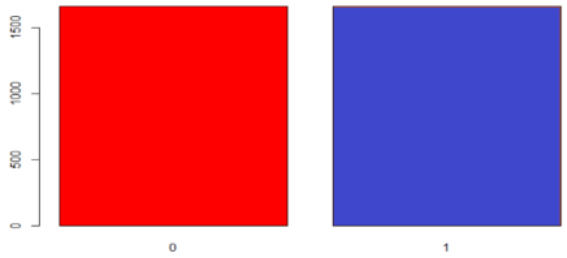
Over-Sampling increases the number of instances in the minority class by randomly replicating them in order to present a higher representation of the minority class in the sample.

- **Advantages**
  - Unlike under sampling this method leads to no information loss.
  - Outperforms under sampling

- **Disadvantages**
  - It increases the likelihood of overfitting since it replicates the minority class events

- See below the result after using over-sampling on the patients' data.





BEFORE USING OVER-SAMPLING	AFTER USING OVER-SAMPLING

1.3.2.4 Neighbor-based approaches: example of the Cluster-based over sampling

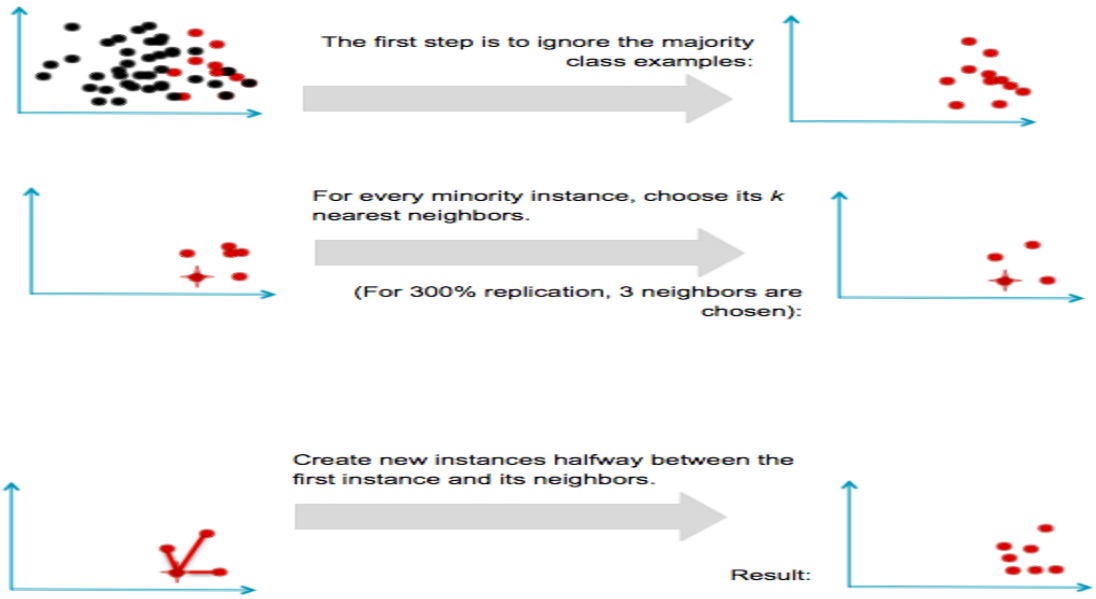
In this case, the K-means clustering algorithm is independently applied to minority and majority class instances. This is to identify clusters in the dataset. Subsequently, each cluster is oversampled such that all clusters of the same class have an equal number of instances and all classes have the same size.

- **Advantages**
  - This clustering technique helps overcome the challenge between class imbalance. Where the number of examples representing positive class differs from the number of examples representing a negative class.
  - Also, overcome challenges within class imbalance, where a class is composed of different sub clusters. And each sub cluster does not contain the same number of examples.

- **Disadvantages**
  - The main drawback of this algorithm, like most oversampling techniques is the possibility of over-fitting the training data.

1.3.2.5 Synthesizing new examples: SMOTE and descendants

This direction of research has involved not resampling of examples, but synthesis of new ones. The best known of this approach is the Chawla's SMOTE system or SMOTE (Synthetic Minority Over-Sampling Technique) simply. The idea is to create new minority examples by interpolating existing ones. The process is basically as follows given by Fawcett (2015):

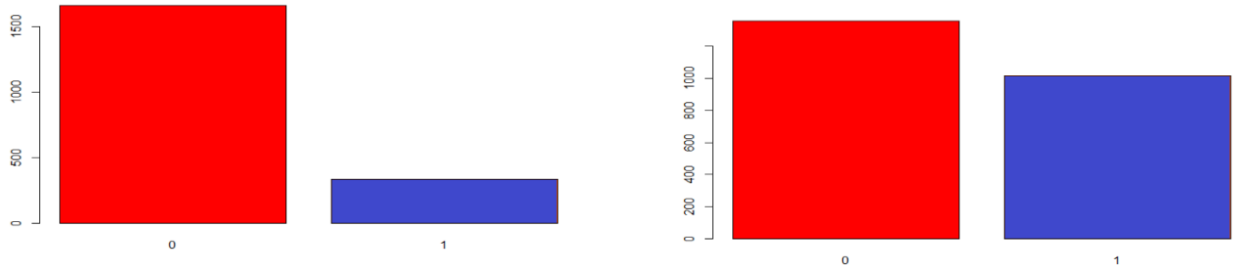


SMOTE was generally successful and led to many variants, extensions, and adaptations to different concept learning algorithms. SMOTE has some advantages and disadvantages.

- **Advantages**
  - Mitigates the problem of overfitting caused by random oversampling as synthetic examples are generated rather than replication of instances
  - No loss of useful information

- **Disadvantages**
  - While generating synthetic examples SMOTE does not take into consideration neighboring examples from other classes. This can result in increase in overlapping of classes and can introduce additional noise
  - SMOTE is not very effective for high dimensional data

See below the result after using SMOTE on the patients' data.



BEFORE USING SMOTE	AFTER USING SMOTE

1.4 Selection of Attributes

In many practical situations there are far too many attributes for learning schemes to handle, and some of them—perhaps the overwhelming majority—are clearly irrelevant or redundant. Consequently, the data must be preprocessed to select a subset of the attributes to use in learning. Most machine learning algorithms are designed to learn which are the most appropriate attributes to use for making their decisions. For example, decision tree methods

choose the most promising attribute to split on at each point and should—in theory—never select irrelevant or unhelpful attributes.

Our selection criteria are based on the values of Pearson Statistics and the value of P-value. The values of Pearson Statistics need to be significant at 5%.

Table 11: Selection criteria for Alzheimer Disease

Selected Attributes	*Ranks*	*p-value*	*Pearson Chi-square Value*	*df*
avg_commute	1	0.0000	2853.5000	1520
daily_internet_use	2	0.0000	1612.5000	573
employmnet_status	3	0.0000	37.4110	3
Zipcode	4	0.0000	48.1410	12
Ancestry	5	0.0006	17.2010	3
marital_status	6	0.0033	8.6472	1
Age	7	0.0062	12.3620	3
Children	8	0.0126	17.8620	7
available_vehicles	9	0.0620	8.9663	4
military_service	10	0.1426	2.1493	1
Education	11	0.7514	1.2066	1
Gender	12	0.8622	0.0301	1
** Pearson Chi-Square Value is significant at the 1% level				

* Pearson Chi-Square Value is significant at the 5% level

Table 12: Selection criteria for Hypertension Disease

Selected Attributes	*Ranks*	*p-value*	*Pearson Chi-square Value*	*df*
avg_commute	1	0.0000	2986.7000	1520
daily_internet_use	2	0.0000	1605.3000	573
age	3	0.0000	35.9430	3
children	4	0.0000	36.9270	7
zipcode	5	0.0000	45.1350	12
gender	6	0.0123	6.2623	1
marital_status	7	0.2662	0.2361	1
ancestry	8	0.3994	2.9499	3
available_vehicles	9	0.6726	2.3449	4
education	10	0.6908	1.4630	1
employmnet_status	11	0.8447	0.8197	3
military_service	12	0.8582	0.0319	1

** Pearson Chi-Square Value is significant at the 1% level

* Pearson Chi-Square Value is significant at the 5% level

Table 13: Selection criteria for Skin Cancer Disease

Selected Attributes	*Ranks*	*p-value*	*Pearson Chi-square Value*	*df*
avg_commute	1	0.0000	3188.0	1520
daily_internet_use	2	0.0000	1911.4000	573
employmnet_status	3	0.0000	41.3430	3
Children	4	0.0000	45.3300	7
available_vehicles	5	0.0000	31.8320	4
Age	6	0.0001	21.3090	3
Zipcode	7	0.0129	25.4450	12
marital_status	8	0.1968	1.6657	1
Ancestry	9	0.3297	3.4317	3
military_service	10	0.4547	0.5588	1
Education	11	0.7814	1.0819	1
Gender	12	0.9463	0.0045	1
** Pearson Chi-Square Value is significant at the 1% level * Pearson Chi-Square Value is significant at the 5% level				

2. Predictive Modeling/Classification

In the analysis, three different classification algorithms were used to build three different models. The algorithms are Naïve Bayes, Random Forest, and Logistic Regression.

The evaluation of the datasets was broken up into three datasets. The first dataset is using Under-sampling data, the second dataset is using over-sampling data, and the third dataset is using SMOTE algorithm. Each evaluation were ran ten different iterations.

Using R, the data was split into Train data (70% data) and Validation data (30% data). The 10-fold cross-validation is applied to evaluate model performance.

Using the 10-fold cross validation (10-FCV):

Cross-validation, a standard evaluation technique, is a systematic way of running repeated percentage splits. Divide a dataset into 10 pieces ("folds"), then hold out each piece in turn for testing and train on the remaining 9 together. This gives 10 evaluation results, which are averaged.

2.1 Naive Bayes Classifier

Advantages:

- Simple but powerful algorithm for predictive modelling
- Based on Bayes theorem of conditional probability that event A will happen only when event B has already happened
- Bayes' Theorem in terms of probability P(A|B) = P(B|A) P(A) / P(B)
  P denotes probability

  P(A|B) Probability of event A occurring given that event B has occurred

  P(B|A) Probability of event B occurring given that event A has occurred

  P(A) Probability of event A occurring

  P(B) Probability of event B occurring

- Fast and scalable
- Performs well in categorical input variables

Disadvantages

- Assumes that all variables in the dataset are independent i.e. are not correlated to each other which is almost impossible in real life
- Must choose the most probable function
- Assumes Gaussian distribution in case of numerical data which is a strong assumption

- Will assign a zero probability when encountered with a categorical variable in test data which was not present in the training data and will be unable to make a prediction. This is known as zero frequency. Need to use smoothing technique in such a case

2.2 Random Forest Classifier

Random Forest consists of large number of individual decision trees that operate as an ensemble, the use of multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Each individual tree in the random forest spits out class prediction and the class with the most votes become the modle's prediction.

Random Forest algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, random forest algorithm can be used for solving regression and classification problems too.

Advantages:

- It can handle binary features, categorical features, and numerical features
- There is very little pre-processing that needs to be done
- The data does not need to be rescaled or transformed
- Random forest handles outliers by essentially binning them. It is also indifferent to non-linear features
- It has method for balancing error in class population unbalance data sets, it tries to minimize the overall error rate
- Each decision tree has a high variance, but low bias and helps reduce the variance

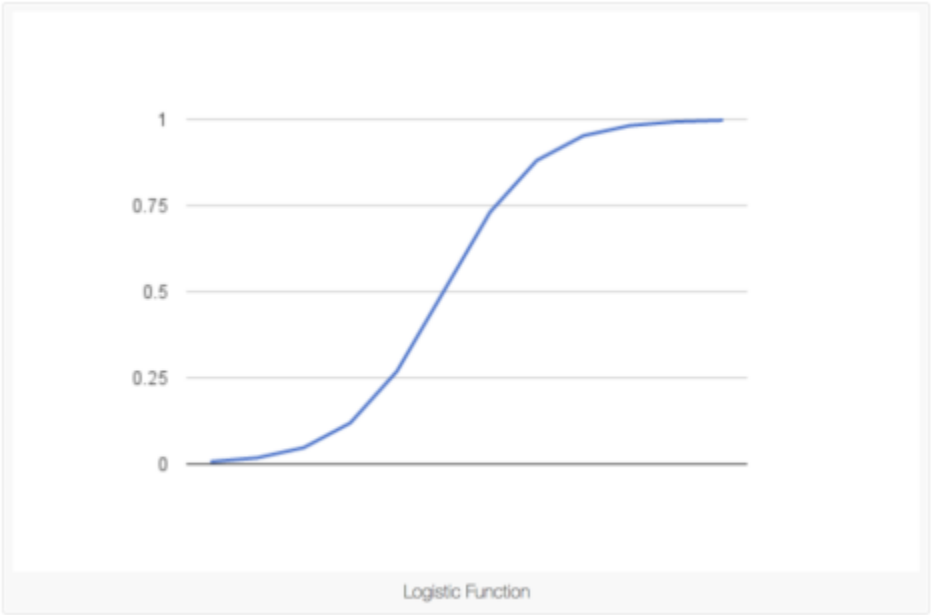
Disadvantages

- It can tend to overfit, so tune the hyperparameters is required
- For very large data sets, the size of the trees can take up a lot of memory

2.3 Logistic Regression Classifiers

- A machine learning algorithm which is based on the concept of probability. It is used for classification and used to do predictive analysis. It makes linear regression do probabilities.
- Measures relationship between dependent variable and 1 or more independent variables
- Never gets below 0 and never gets above 1. Has a smooth transition in between 0 and 1.

- Differs from linear regression in using a more complex cost function known as sigmoid function or the 'logistic function'. Linear regression minimizes a squared error while logistic regression maximizes probabilistic function known as 'log likelihood' function.
- Limits the cost function between 0 and 1
- Sigmoid maps predictions to probabilities
- Gives set of classes based on probability when the input is passed through the prediction function which returns a probability score between 0 and 1.
- There is a threshold value above which classify values into Class 1 and below which classify values into the other Class2.
- Cost function of linear regression can be used to minimize the cost function in logistic regression and would give many local minimums and would be difficult to find the global minimum.
- Gradient Descent is used to minimize the cost value function. Gradient descent function is applied on each parameter.
- Works better when attributes which are unrelated to output variable and also attributes which are correlated to each other
- Feature Engineering plays an important role in logistic regression
- Logistic function looks like the following:



Logistic function (Referenced from "The Logistic Regression Algorithm", in machine-learning blog.com, 2018)

Advantages:

- Used to solve a classification problem i.e. when the variable takes 2 values
- Very efficient and simple, doesn't require too many computational resources
- Highly interpretable in nature
- Doesn't require input variables to be scaled or tuned
- Serves as a baseline for complex algorithms

Disadvantages:

- It can't be used to solve non-linear problems as its decision surface is linear
- Can only predict a categorical outcome as its outcome is discrete
- Depends highly on the proper presentation of data
- Can be easily outperformed by more complex algorithms

2.4 Metrics for Classification Tasks

Confusion matrix

The confusion matrix for a two-class problem presents the results obtained by a given classifier. This table provides for each class the instances that were correctly classified i.e. the number of True Positives (TP) and True Negatives (TN), and the instances that were wrongly classified, i.e. the number of False Positives (FP) and False Negatives (FN).

		Predicted	
		Positive	Negative
Observed	Positive	*TP*	*FN*
	Negative	*FP*	*TN*

The metrics used in imbalanced domains must consider the user preferences and, should consider the data distribution. To fulfill this goal several performance measures were proposed.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$true\ positive\ rate\ (recall\ or\ sensitivity): TP\ rate = \frac{TP}{TP + FN}$$

$$true\ negative\ rate\ (specificity):\ TN\ rate = \frac{TN}{TN + FP}$$

$$false\ positive\ rate:\ FP\ rate = \frac{FP}{TN + FP}$$

$$false\ negative\ rate:\ FN\ rate = \frac{FN}{TP + FN}$$

$$positive\ preditive\ value\ (precision):\ PP\ value = \frac{TP}{TP + FP}$$

$$negative\ preditive\ value:\ NP\ value = \frac{TN}{TN + FN}$$

The *F-measure* (F_β), a combination of both precision and recall, is defined as follows:

$$F_\beta = \frac{(1+\beta)^2.recall.precision}{\beta^2.recall + precision}$$

Where β is a coefficient to adjust the relative importance of recall with respect to precision (if $\beta = 1$ precision and recall have the same weight, large values of β will increase the weight of recall whilst less than 1 will give more importance to precision).

F_β is commonly used and is more informative about the effectiveness of a classifier on predicting correctly the cases that matter to the user. This metric value is high when both recall (a measure of completeness) and precision (a measure of exactness) are high.

An also frequently used metric when with imbalanced data sets is the geometric mean (G-mean) which is defined as:

$$G - Mean = \sqrt{\frac{TP}{TP + FN}} * \sqrt{\frac{TN}{TN + FP}} = \sqrt{\text{sensitivity} * \text{specificity}}$$

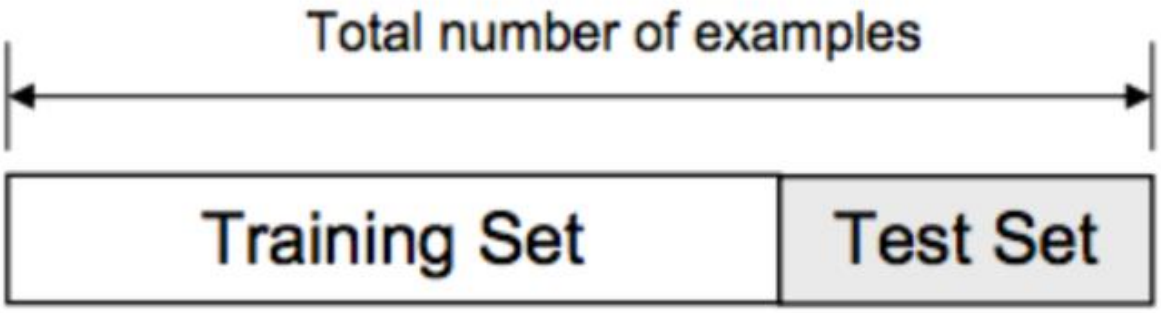
G-Mean is an interesting measure because it computes the geometric mean of the accuracies of the two classes, attempting to maximize them while obtaining good balance.

The receiver operating characteristics (ROC) curve and the corresponding area under the ROC curve (AUC) are also two popular tools used in imbalanced domains. The AUC allows the evaluation of the best model on average.

$$AUC = \frac{1 + TP\ rate - FP\ rate}{2} = \frac{TP\ rate + TN\ rate}{2}$$

2.5 Running the datasets against the algorithms

The data used is split into training (70% data) and test (30% data) data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. The test dataset is there in order to test the model's prediction on this subset.



Train/Test Split

2.5.1.1 Alzheimer disease: Logistic Regression balanced dataset

Below is the result of running the Logistic Regression algorithm using under-sample, over-sample and SMOTE. The result shows the 10 iteration results.

	Logistic Regression											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.7900	0.8443	0.9183	0.8798	0.5550	0.8827	0.5398	0.6700	0.6983	0.8351	0.7968	0.8155
2	0.8117	0.8406	0.9562	0.8947	0.5500	0.8625	0.5498	0.6715	0.7033	0.8333	0.8068	0.8198
3	0.7900	0.8406	0.9243	0.8805	0.5483	0.8714	0.5398	0.6667	0.7283	0.8410	0.8327	0.8368
4	0.7850	0.8410	0.9163	0.8770	0.5267	0.8785	0.5040	0.6405	0.7100	0.8374	0.8108	0.8239
5	0.7883	0.8415	0.9203	0.8792	0.5200	0.8690	0.5020	0.6364	0.7250	0.8363	0.8347	0.8355
6	0.7967	0.8430	0.9303	0.8845	0.5450	0.8567	0.5478	0.6683	0.7017	0.8414	0.7928	0.8164
7	0.7967	0.8442	0.9283	0.8843	0.5350	0.8656	0.5259	0.6543	0.7133	0.8354	0.8187	0.8270
8	0.7867	0.8489	0.9064	0.8767	0.5367	0.8758	0.5199	0.6525	0.7000	0.8340	0.8008	0.8171
9	0.7817	0.8391	0.9143	0.8751	0.5450	0.8459	0.5578	0.6723	0.7250	0.8404	0.8287	0.8345
10	0.7800	0.8464	0.9004	0.8726	0.5317	0.8647	0.5219	0.6509	0.7017	0.8358	0.8008	0.8179
Mean	**0.7907**	**0.8430**	**0.9215**	**0.8804**	**0.5393**	**0.8673**	**0.5309**	**0.6583**	**0.7107**	**0.8370**	**0.8124**	**0.8244**
Min	**0.7800**	**0.8391**	**0.9004**	**0.8726**	**0.5200**	**0.8459**	**0.5020**	**0.6364**	**0.6983**	**0.8333**	**0.7928**	**0.8155**
Max	**0.8117**	**0.8489**	**0.9562**	**0.8947**	**0.5550**	**0.8827**	**0.5578**	**0.6723**	**0.7283**	**0.8414**	**0.8347**	**0.8368**

2.5.1.2 Alzheimer disease: Random Forest balanced dataset

Below is the result of running the Random Forest algorithm using under-sample, over-sample and SMOTE. The result shows the 10 iteration results.

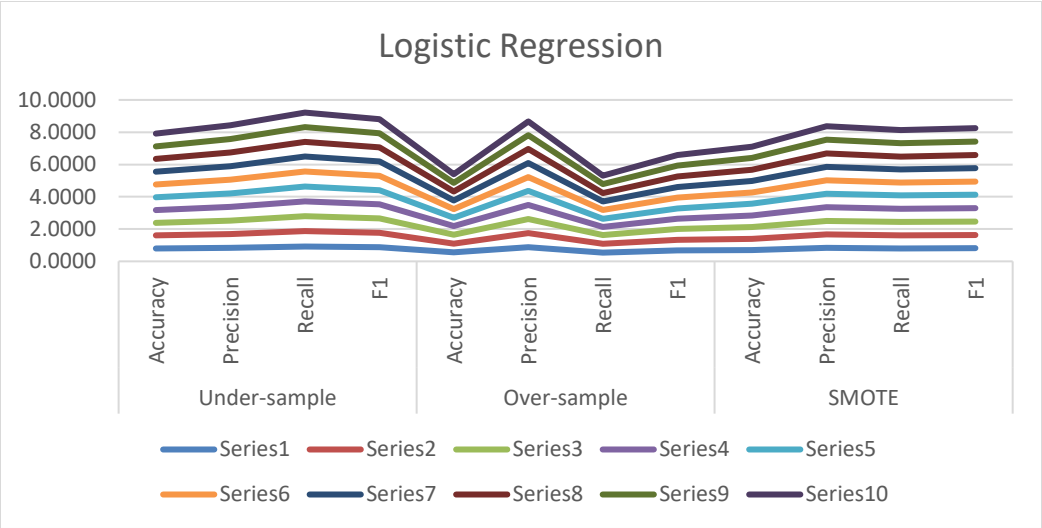
	Random Forest											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.8117	0.8394	0.9582	0.8949	0.5967	0.8939	0.5876	0.7091	0.7650	0.8412	0.8865	0.8632
2	0.8183	0.8441	0.9602	0.8984	0.5783	0.8952	0.5618	0.6903	0.7683	0.8405	0.8924	0.8657
3	0.8150	0.8448	0.9542	0.8962	0.5567	0.8907	0.5359	0.6692	0.7833	0.8444	0.9084	0.8752
4	0.7967	0.8467	0.9243	0.8838	0.5917	0.9028	0.5737	0.7016	0.7483	0.8369	0.8685	0.8524
5	0.8300	0.8436	0.9781	0.9059	0.5850	0.8941	0.5717	0.6974	0.7767	0.8446	0.8984	0.8707
6	0.8267	0.8443	0.9721	0.9037	0.5800	0.9058	0.5558	0.6889	0.7433	0.8412	0.8546	0.8478
7	0.8033	0.8529	0.9243	0.8872	0.6083	0.8985	0.5996	0.7192	0.7750	0.8392	0.9044	0.8706
8	0.8017	0.8450	0.9343	0.8874	0.5800	0.8931	0.5657	0.6927	0.7500	0.8398	0.8665	0.8529
9	0.8150	0.8485	0.9482	0.8956	0.5700	0.8935	0.5518	0.6823	0.7683	0.8393	0.8944	0.8660
10	0.8117	0.8455	0.9482	0.8939	0.5967	0.9063	0.5777	0.7056	0.7667	0.8377	0.8944	0.8651
Mean	**0.8130**	**0.8455**	**0.9502**	**0.8947**	**0.5843**	**0.8974**	**0.5681**	**0.6956**	**0.7645**	**0.8405**	**0.8869**	**0.8630**
Min	**0.7967**	**0.8394**	**0.9243**	**0.8838**	**0.5567**	**0.8907**	**0.5359**	**0.6692**	**0.7433**	**0.8369**	**0.8546**	**0.8478**
Max	**0.8300**	**0.8529**	**0.9781**	**0.9059**	**0.6083**	**0.9063**	**0.5996**	**0.7192**	**0.7833**	**0.8446**	**0.9084**	**0.8752**

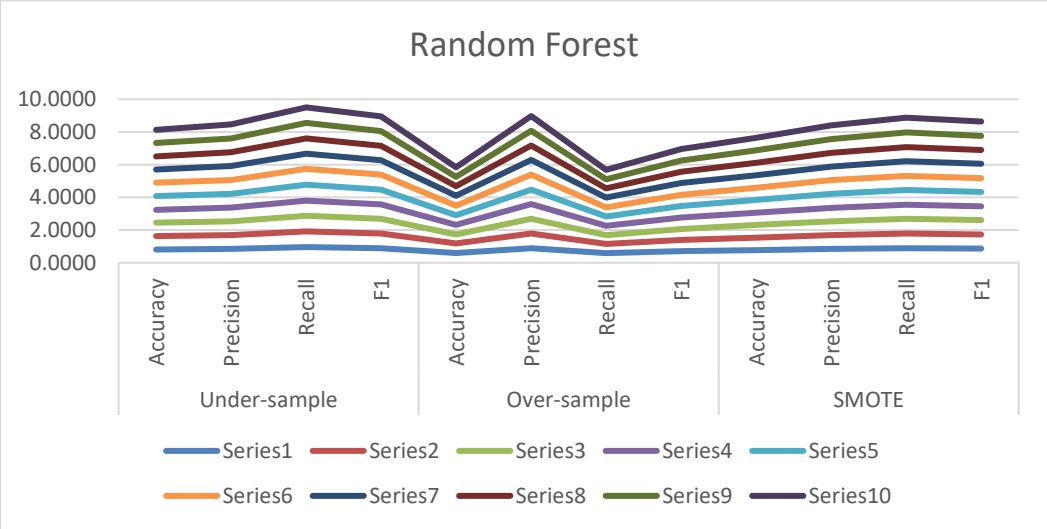
2.5.1.3 Alzheimer disease: Naïve Bayes balanced dataset

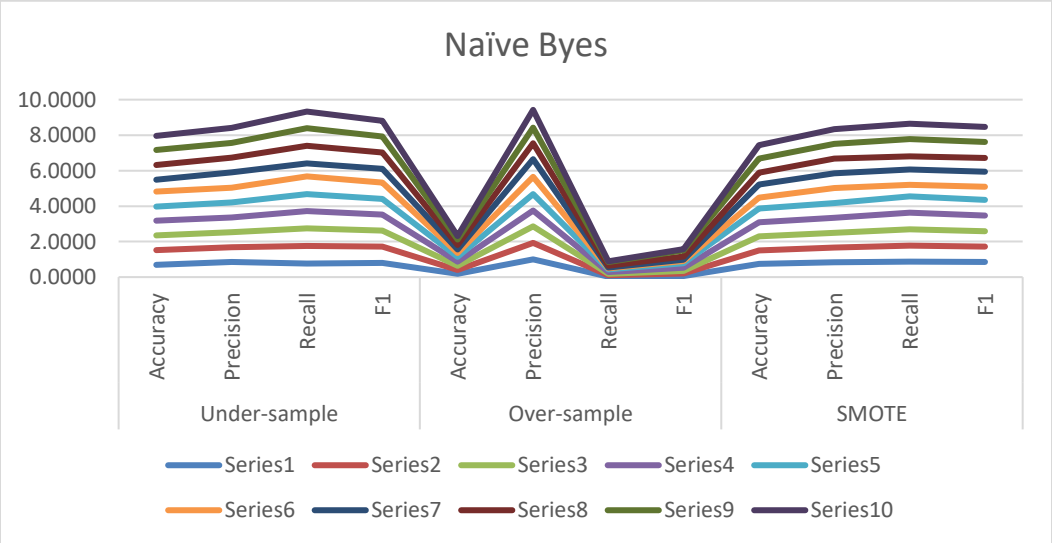
Below is the result of running the Naïve Bayes algorithm using under-sample, over-sample and SMOTE. The result shows the 10 iteration results.

	Naïve Byes											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.6867	0.8536	0.7550	0.8013	0.1867	1.0000	0.0279	0.0543	0.7500	0.8333	0.8765	0.8544
2	0.8367	0.8367	1.0000	0.9111	0.2233	0.9286	0.0777	0.1434	0.7600	0.8315	0.8944	0.8618
3	0.8283	0.8376	0.9861	0.9058	0.2267	0.9318	0.0817	0.1502	0.7783	0.8348	0.9163	0.8737
4	0.8300	0.8378	0.9880	0.9068	0.2350	0.8909	0.0976	0.1759	0.7950	0.8366	0.9382	0.8845
5	0.7967	0.8333	0.9462	0.8862	0.2783	0.9259	0.1494	0.2573	0.7850	0.8348	0.9263	0.8782
6	0.8367	0.8367	1.0000	0.9111	0.2150	0.9697	0.0637	0.1196	0.6033	0.8420	0.6474	0.7320
7	0.6717	0.8588	0.7271	0.7875	0.1983	1.0000	0.0418	0.0803	0.7483	0.8343	0.8725	0.8530
8	0.8350	0.8375	0.9960	0.9099	0.2383	0.8947	0.1016	0.1825	0.6567	0.8304	0.7410	0.7832
9	0.8367	0.8367	1.0000	0.9111	0.3133	0.8814	0.2072	0.3355	0.8067	0.8328	0.9622	0.8928
10	0.7967	0.8442	0.9283	0.8843	0.1983	1.0000	0.0418	0.0803	0.7433	0.8308	0.8705	0.8502
Mean	**0.7955**	**0.8413**	**0.9327**	**0.8815**	**0.2313**	**0.9423**	**0.0890**	**0.1579**	**0.7427**	**0.8341**	**0.8645**	**0.8464**
Min	**0.6717**	**0.8333**	**0.7271**	**0.7875**	**0.1867**	**0.8814**	**0.0279**	**0.0543**	**0.6033**	**0.8304**	**0.6474**	**0.7320**
Max	**0.8367**	**0.8588**	**1.0000**	**0.9111**	**0.3133**	**1.0000**	**0.2072**	**0.3355**	**0.8067**	**0.8420**	**0.9622**	**0.8928**

2.5.1.4 Alzheimer disease: Graph show the models differences







2.5.2.1 Hypertension disease: Logistic Regression balanced dataset

Below is the result of running the Logistic Regression algorithm using under-sample, over-sample and SMOTE. The result shows the 10 iteration results.

	Logistic Regression											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.8667	0.8735	0.9904	0.9283	0.4950	0.8957	0.4761	0.6217	0.7933	0.8859	0.8757	0.8808
2	0.8300	0.8779	0.9350	0.9056	0.5283	0.8974	0.5182	0.6570	0.7717	0.8907	0.8413	0.8653
3	0.8683	0.8712	0.9962	0.9295	0.5283	0.9054	0.5124	0.6545	0.7683	0.8855	0.8432	0.8639
4	0.8200	0.8780	0.9216	0.8993	0.5200	0.8930	0.5105	0.6496	0.7700	0.8858	0.8451	0.8650
5	0.8283	0.8763	0.9350	0.9047	0.5117	0.8859	0.5048	0.6431	0.7917	0.8872	0.8719	0.8795
6	0.8700	0.8739	0.9943	0.9302	0.5183	0.8774	0.5201	0.6531	0.8100	0.8866	0.8967	0.8916
7	0.8600	0.8885	0.9598	0.9228	0.5483	0.8962	0.5449	0.6778	0.7850	0.8878	0.8623	0.8749
8	0.8700	0.8739	0.9943	0.9302	0.5167	0.8949	0.5048	0.6455	0.7850	0.8863	0.8642	0.8751
9	0.8300	0.8752	0.9388	0.9059	0.5083	0.8958	0.4933	0.6363	0.7983	0.8911	0.8757	0.8833
10	0.8017	0.8755	0.9006	0.8878	0.4900	0.8945	0.4704	0.6165	0.8083	0.8893	0.8910	0.8902
Mean	**0.8445**	**0.8764**	**0.9566**	**0.9144**	**0.5165**	**0.8936**	**0.5055**	**0.6455**	**0.7882**	**0.8876**	**0.8667**	**0.8769**
Min	**0.8017**	**0.8712**	**0.9006**	**0.8878**	**0.4900**	**0.8774**	**0.4704**	**0.6165**	**0.7683**	**0.8855**	**0.8413**	**0.8639**
Max	**0.8700**	**0.8885**	**0.9962**	**0.9302**	**0.5483**	**0.9054**	**0.5449**	**0.6778**	**0.8100**	**0.8911**	**0.8967**	**0.8916**

2.5.2.2 Hypertension disease Random Forest balanced dataset

Below is the result of running the Random Forest algorithm using under-sample, over-sample and SMOTE. The result shows the 10 iteration results.

	Random Forest											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.8567	0.8895	0.9541	0.9207	0.5217	0.9436	0.4799	0.6362	0.8250	0.8899	0.9120	0.9008
2	0.8233	0.8897	0.9101	0.8998	0.4933	0.9294	0.4532	0.6093	0.8250	0.8943	0.9063	0.9003
3	0.8667	0.8893	0.9675	0.9267	0.5883	0.9340	0.5679	0.7063	0.7950	0.8891	0.8738	0.8814
4	0.8517	0.8931	0.9426	0.9172	0.5650	0.9367	0.5373	0.6829	0.8400	0.8917	0.9293	0.9101
5	0.8583	0.8996	0.9426	0.9206	0.5867	0.9421	0.5602	0.7026	0.8350	0.8926	0.9216	0.9069
6	0.8683	0.8868	0.9732	0.9280	0.5867	0.9283	0.5698	0.7062	0.8400	0.8903	0.9312	0.9103
7	0.8750	0.8986	0.9656	0.9309	0.6067	0.9335	0.5908	0.7237	0.8067	0.8921	0.8853	0.8887
8	0.8767	0.8825	0.9904	0.9333	0.5583	0.9272	0.5354	0.6788	0.8317	0.8893	0.9216	0.9052
9	0.8483	0.9045	0.9235	0.9139	0.5633	0.9365	0.5354	0.6813	0.8150	0.8916	0.8967	0.8942

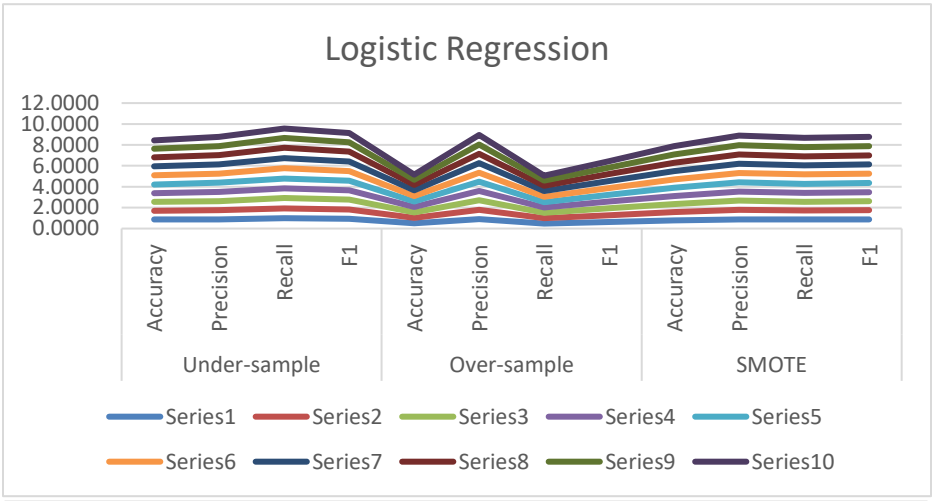
10	0.8600	0.8969	0.9484	0.9219	0.5367	0.9329	0.5048	0.6551	0.8383	0.8915	0.9273	0.9091
Mean	**0.8585**	**0.8930**	**0.9518**	**0.9213**	**0.5607**	**0.9344**	**0.5335**	**0.6782**	**0.8252**	**0.8913**	**0.9105**	**0.9007**
Min	**0.8233**	**0.8825**	**0.9101**	**0.8998**	**0.4933**	**0.9272**	**0.4532**	**0.6093**	**0.7950**	**0.8891**	**0.8738**	**0.8814**
Max	**0.8767**	**0.9045**	**0.9904**	**0.9333**	**0.6067**	**0.9436**	**0.5908**	**0.7237**	**0.8400**	**0.8943**	**0.9312**	**0.9103**

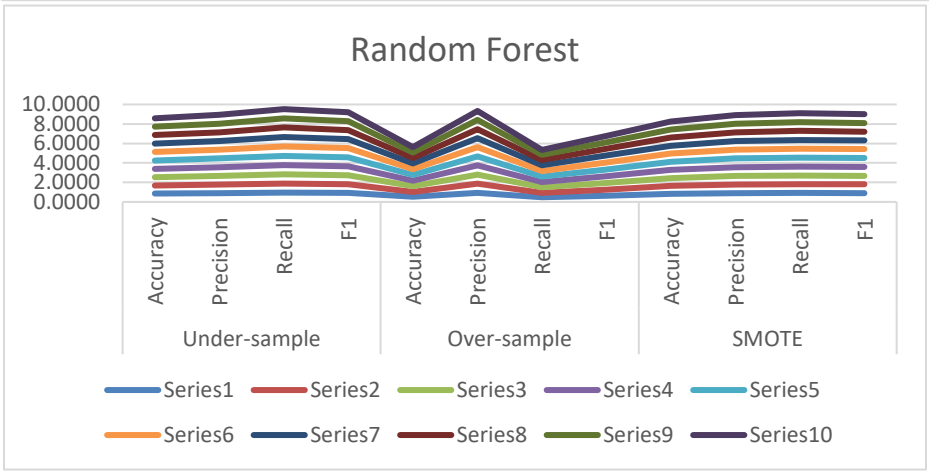
2.5.2.3 Hypertension disease: Naïve Bayes balanced dataset

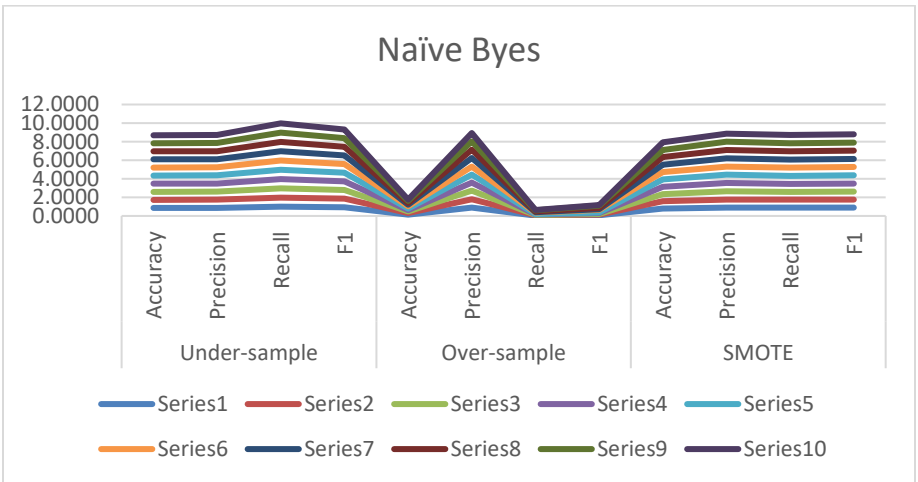
Below is the result of running the Naïve Bayes algorithm using under-sample, over-sample and SMOTE. The result shows the 10 iteration results.

	Naïve Byes											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.8717	0.8717	1.0000	0.9314	0.1567	0.9048	0.0363	0.0699	0.8117	0.8868	0.8987	0.8927
2	0.8550	0.8759	0.9713	0.9211	0.1567	0.9048	0.0363	0.0699	0.7833	0.8891	0.8585	0.8735
3	0.8717	0.8717	1.0000	0.9314	0.1567	0.9048	0.0363	0.0699	0.7567	0.8855	0.8279	0.8557
4	0.8717	0.8729	0.9981	0.9313	0.1683	0.8750	0.0535	0.1009	0.7733	0.8862	0.8489	0.8672
5	0.8717	0.8717	1.0000	0.9314	0.1583	0.8462	0.0421	0.0801	0.8017	0.8855	0.8872	0.8863
6	0.8717	0.8717	1.0000	0.9314	0.2483	0.8830	0.1587	0.2690	0.8017	0.8870	0.8853	0.8861
7	0.8717	0.8717	1.0000	0.9314	0.1583	0.9091	0.0382	0.0734	0.7967	0.8893	0.8757	0.8825
8	0.8717	0.8717	1.0000	0.9314	0.1917	0.9130	0.0803	0.1476	0.8100	0.8880	0.8948	0.8914
9	0.8717	0.8717	1.0000	0.9314	0.2250	0.8919	0.1262	0.2211	0.7750	0.8865	0.8509	0.8683
10	0.8717	0.8717	1.0000	0.9314	0.1567	0.9048	0.0363	0.0699	0.8117	0.8868	0.8987	0.8927
Mean	**0.8700**	**0.8722**	**0.9969**	**0.9304**	**0.1777**	**0.8937**	**0.0644**	**0.1172**	**0.7922**	**0.8871**	**0.8727**	**0.8796**
Min	**0.8550**	**0.8717**	**0.9713**	**0.9211**	**0.1567**	**0.8462**	**0.0363**	**0.0699**	**0.7567**	**0.8855**	**0.8279**	**0.8557**
Max	**0.8717**	**0.8759**	**1.0000**	**0.9314**	**0.2483**	**0.9130**	**0.1587**	**0.2690**	**0.8117**	**0.8893**	**0.8987**	**0.8927**

2.5.2.4 Hypertension disease: Graph show the models differences







2.5.3.1 Skin Cancer disease: Logistic Regression balanced dataset

Below is the result of running the Logistic Regression algorithm using under-sample, over-sample and SMOTE. The result shows the 10 iteration results.

	Logistic Regression											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.8583	0.8904	0.9588	0.9234	0.4583	0.9300	0.4232	0.5817	0.7600	0.8963	0.8258	0.8596
2	0.8633	0.8937	0.9607	0.9260	0.5217	0.9364	0.4963	0.6487	0.7483	0.8900	0.8184	0.8527
3	0.7967	0.8872	0.8839	0.8856	0.5067	0.9190	0.4888	0.6381	0.7267	0.8887	0.7921	0.8376
4	0.8567	0.8902	0.9569	0.9224	0.4950	0.9326	0.4663	0.6217	0.8017	0.8922	0.8839	0.8881
5	0.8417	0.8885	0.9401	0.9136	0.5167	0.9178	0.5019	0.6489	0.7000	0.8916	0.7547	0.8174
6	0.8550	0.8956	0.9476	0.9208	0.5700	0.9259	0.5618	0.6993	0.7567	0.8927	0.8258	0.8580
7	0.7817	0.8913	0.8596	0.8751	0.5550	0.9320	0.5393	0.6833	0.7317	0.8910	0.7959	0.8408
8	0.7867	0.8949	0.8614	0.8779	0.5300	0.9406	0.5037	0.6561	0.7267	0.8936	0.7865	0.8367
9	0.8000	0.8950	0.8783	0.8866	0.5383	0.9416	0.5131	0.6642	0.7750	0.8966	0.8446	0.8698
10	0.8583	0.8946	0.9532	0.9229	0.5150	0.9324	0.4906	0.6429	0.7833	0.8976	0.8539	0.8752
Mean	**0.8298**	**0.8921**	**0.9200**	**0.9054**	**0.5207**	**0.9308**	**0.4985**	**0.6485**	**0.7510**	**0.8930**	**0.8182**	**0.8536**
Min	**0.7817**	**0.8872**	**0.8596**	**0.8751**	**0.4583**	**0.9178**	**0.4232**	**0.5817**	**0.7000**	**0.8887**	**0.7547**	**0.8174**
Max	**0.8633**	**0.8956**	**0.9607**	**0.9260**	**0.5700**	**0.9416**	**0.5618**	**0.6993**	**0.8017**	**0.8976**	**0.8839**	**0.8881**

2.5.3.2 Skin Cancer disease: Random Forest balanced dataset

Below is the result of running the Random Forest algorithm using under-sample, over-sample and SMOTE. The result shows the 10 iteration results.

	Random Forest											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.7933	0.9133	0.8483	0.8796	0.6200	0.9527	0.6030	0.7091	0.7583	0.9027	0.8165	0.8574
2	0.7917	0.9082	0.8521	0.8792	0.6167	0.9419	0.6067	0.6903	0.7600	0.9079	0.8127	0.8577
3	0.8317	0.9047	0.9064	0.9055	0.5983	0.9536	0.5768	0.6692	0.7650	0.9052	0.8221	0.8616
4	0.7883	0.9179	0.8371	0.8756	0.6150	0.9292	0.6142	0.7016	0.8150	0.9044	0.8858	0.8950
5	0.7833	0.9122	0.8371	0.8730	0.6033	0.9540	0.5824	0.6974	0.7583	0.9044	0.8146	0.8571
6	0.7950	0.9135	0.8502	0.8807	0.6583	0.9507	0.6498	0.6889	0.7717	0.9043	0.8315	0.8663
7	0.7267	0.9167	0.7622	0.8323	0.5933	0.9503	0.5730	0.7192	0.7750	0.9080	0.8315	0.8680
8	0.7800	0.9069	0.8390	0.8716	0.5717	0.9482	0.5487	0.6927	0.7900	0.8953	0.8652	0.8800
9	0.7817	0.9138	0.8333	0.8717	0.6117	0.9493	0.5955	0.6823	0.7800	0.9069	0.8390	0.8716
10	0.7833	0.9106	0.8390	0.8733	0.6133	0.9548	0.5936	0.7056	0.8017	0.9061	0.8670	0.8861
Mean	**0.7855**	**0.9118**	**0.8404**	**0.8743**	**0.6102**	**0.9485**	**0.5944**	**0.6956**	**0.7775**	**0.9045**	**0.8386**	**0.8701**

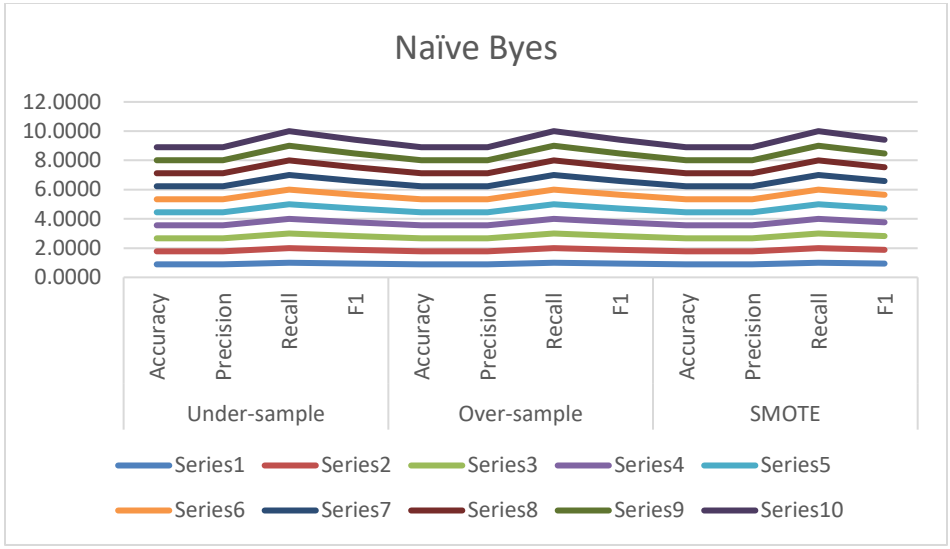
Min	**0.7267**	**0.9047**	**0.7622**	**0.8323**	**0.5717**	**0.9292**	**0.5487**	**0.6692**	**0.7583**	**0.8953**	**0.8127**	**0.8571**
Max	**0.8317**	**0.9179**	**0.9064**	**0.9055**	**0.6583**	**0.9548**	**0.6498**	**0.7192**	**0.8150**	**0.9080**	**0.8858**	**0.8950**

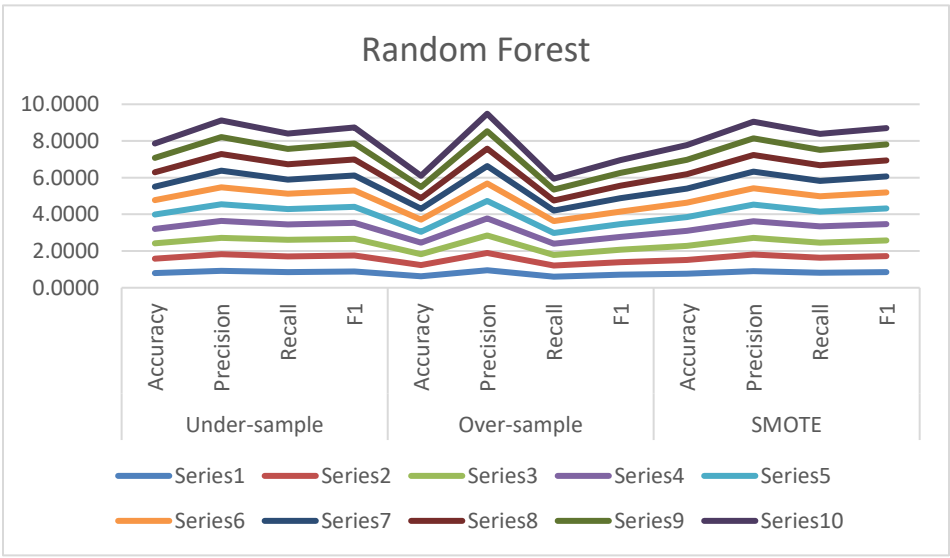
2.5.3.3 Skin Cancer disease: Naïve Bayes balanced dataset

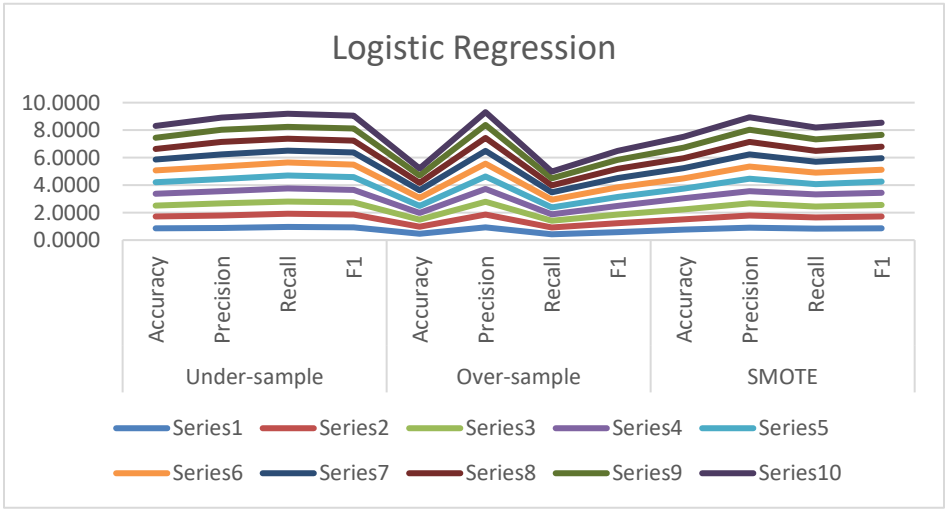
Below is the result of running the Naïve Bayes algorithm using under-sample, over-sample and SMOTE. The result shows the 10 iteration results.

	Naïve Byes											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418
2	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418
3	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418
4	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418
5	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418
6	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418
7	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418
8	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418
9	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418
10	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418	0.8900	0.8900	1.0000	0.9418
Mean	**0.8900**	**0.8900**	**1.0000**	**0.9418**	**0.8900**	**0.8900**	**1.0000**	**0.9418**	**0.8900**	**0.8900**	**1.0000**	**0.9418**
Min	**0.8900**	**0.8900**	**1.0000**	**0.9418**	**0.8900**	**0.8900**	**1.0000**	**0.9418**	**0.8900**	**0.8900**	**1.0000**	**0.9418**
Max	**0.8900**	**0.8900**	**1.0000**	**0.9418**	**0.8900**	**0.8900**	**1.0000**	**0.9418**	**0.8900**	**0.8900**	**1.0000**	**0.9418**

2.5.3.4 Skin Cancers disease: Graph show the models differences







2.6 Results

There are situations when the aim is to maximize either recall or precision at the expense of the other metric. However, for the disease screening of patients, the recall near 1.0 is desirable, need to find all patients who actually have the disease, and low precision can be acceptable.

After applying classification's data mining techniques algorithms which are Random Forest, Naïve Bayes, and Logistic Regression, and applying balancing data algorithms which are under-sample, over-sample, and SMOTE, for three classes, Alzheimer, Hypertension, and Skin Cancer diseases. The results obtained from the above experiments in the above tables and after comparing the correctly classified instance percent. It is found that balancing is required and both under-sample and SMOTE gave very close numbers.

Since the aim is to get the best recall, accuracy, and minimum variance numbers between iterations. It is found for all three diseases analysis that both Logistic Regression with Under-Sampling algorithms gave the best result.

3 Recommendations

The data analysis based on ML algorithms threw varied and interesting results. However, more attributes need to be added to the dataset for better results. Attributes such as when the patients migrated to the US, to see the impact of the living habit effecting their disease. Also, blood type, measuring of the blood pressure, and others related to the historical blood tests.

4 Conclusions

The patients' data was pre-processed wherein it was checked for diseases. The data was balanced using Under-sample, Over-sample, and SMOTE. Various machine learning algorithms such Random Forest, Naive Bayes, and Logistic Regression were implemented on the pre-processed data. It gave interesting results and the best model was selected on the basis of Accuracy, Recall and Positive Rate.

REFERENCES

1. Holmboe and Durning, 2014 https://www.ncbi.nlm.nih.gov/books/NBK338593/#
2. BMC Medical Genomics https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5223458/
3. https://www.researchgate.net/journal/1049-510X_Ethnicity_disease
4. https://www.lifewire.com/classification-1019653
5. Interpretable Machine Learning, A Guide for Making Black Box Models Explainable, by Christoph Molnar, 2019-09-18, https://christophm.github.io/interpretable-ml-book/
6. Tips and Tricks for Multi-Class Classification, https://medium.com/@b.terryjack/tips-and-tricks-for-multi-class-classification-c184ae1c8ffc
7. Understating Confusion Matrix, https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62
8. Bing Liu, Wynne Hsu, Yiming Ma, "Integrating Classification and Association Rule Mining", KDD-98, New York, Aug 27-31, 1998.
9. Carlos Ordonez, Norberto Ezquerra, Cesar A. Santana, "Constraining and Summarizing Association Rules in Medical Data", Knowledge and Information Systems, Volume 9, Issue 3, 2006, pp.. 259 - 283.
10. Chawla et al. (2002), "SMOTE: Synthetic Minority Over-Sampling Technique", Journal of Artificial Intelligence Research 16, pp321-357
11. Ian H. Witten, Eibe H. Frank+, Mark A. Hall (2016), "Data Mining Practical Machine Learning Tools and Technique", 4th edition, Morgan Kauffmann
12. Keith McCormick and Jesus Salcedo (2017), "SPSS Statistics for Data Analysis and Visualization", John Wiley edition
13. Krawczyk B. (2016), "Learning from Imabalanced data: open challenges and future directions", Prog Artif Intell, 5, pp 221-232
14. Logistic Function from "The Logistic Regression Algorithm", in machine-learning blog.com, 2018 https://machinelearning-blog.com/2018/04/23/logistic-regression-101/
15. Rajdeep Kaur Aulakh, "Association Rules Mining Using Effective Algorithm: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 3, March 2015, pp. 831-835.
16. Dr. Tamer Abdou, CIND 119, 2019, Ryerson University
17. Dr. Ceni Babaoglu, CIND 642, 2019, Ryerson University
18. Yiu,Tony in "Understanding Random Forests" in Towards DataScience, 2018 https://towardsdatascience.com/understanding-random-forest-58381e0602d2