Tejasvini Mavuleti
08/03/2022

MIS 64099: Capstone Project in Business Analytics

Final Project Report

Effects of Machine Learning Algorithms on Disease Recurrence and Diagnosis

Githib link - https://github.com/tmavulet/64099_tmavulet/tree/main/Capstone%20Project%20in%20Business%20Analytics

Executive Summary

This project aims to understate the patient profiles from their demographics and predict the type of patient and their disease diagnosis and treatment. It is very difficult to keep information about genetics separate from health information, and consumers of genetic tests are becoming more aware of the potential health risks associated with ancestral lineages. Therefore, I chose a project to analyze a patient demographic dataset containing standard information regarding individuals from various ancestral lines. However, because some of the proposed associations have received little attention from oversight agencies and professional genetic associations, scientific developments are outpacing governance regimes for consumer genetic testing. The significance of the research and findings will take me to preprocess the data and balance using under and oversampling and further discretized to enable the Machine Learning algorithms for accurate findings. I have the scope of using numerous machine learning algorithms such as Random Forest, Naive Bayes, and Logistic Regression to tabulate and compare based on parameters such as precision, error rate, and accuracy.

Project background

History - The disease diagnosis has important implications for patient care. When a diagnosis is accurate and made promptly, a patient has the best opportunity for positive health outcomes because clinical decision-making is tailored to a correct understanding of the patient's health problem. Every patient has a different diet and lifestyle. However, when people moved to North America, they brought their eating habits, level of education, and lifestyle with them. In addition, the large and highly diverse population resulted in different diseases affecting different ages and ethnic groups. I correlated to this topic when I found the data set because I have seen myself in their shoes, changing my lifestyle. And if I have a complication, I think I would look for a best bet for my treatment.

Solution - The questions I would like to raise and answer here 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 the United States only. So, I must look at the bigger picture and better analyze what is happening worldwide. Are they using these algorithms to treat patients? How are they affecting their results? After exploring the data, there is a possibility that a patient might have multiple diseases. Therefore, each instance can be assigned with various categories; this type of problem needs to use multi-label classification. There are 13 diseases given in the data set, and they are classified in the binary method. This would be my first task to cleanse and start preparing the dataset.

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

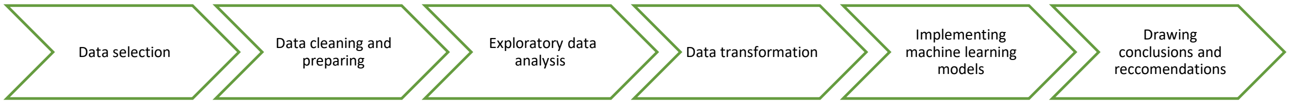
I used R programming to start with the coding and to better understand the data. I used a couple of scholarly articles and journal papers to support my research and conclusions.

What is the project about?

This project uses the patient's data; each patient is diagnosed with a disease. There are 13 diseases listed in the dataset. To simplify the analysis, I created 13 new attributes for the 13 diseases in the dataset to make the data a binary classification problem, where individuals are classified as having a disease or do not have that disease. The dataset contains 14 categorical/nominal variables (not including the class variable). The number of instances is 2000, and variables come from different hospitals. In this study, I focused on the top three diseases (Alzheimer's, hypertension, and skin cancer) for this case study.

Data Review

Steps



Data selection

Data cleaning and preparing

Exploratory data analysis

Data transformation

Implementing machine learning models

Drawing conclusions and reccomendations

1 – Data selection

Data monitoring and analysis improve the health care sector and solve local organizational issues, such as reducing workloads and rising profits for a medical agency. Global problems, such as predicting epidemics and fighting existing diseases more effectively, drive the use of Big Data in medicine. First task was to import the dataset and specifying into data frames.

Often, the data collected in healthcare is complex and governed by highly stringent regulations. Most healthcare data collection companies use a variety of forms such as patient forms, prescriptions, health assessment forms, and the entire patient history. Once this data is collected, it goes through a data entry process so the information can be available in the electronic health record systems and shared among providers.

2 – Data cleaning and preparing

The complexity and regulations around healthcare data and data cleansing is often more urgent and complicated. There are duplications happening for many reasons, including errors in spelling or other patient data. In addition, data cleaning is necessary for multiple healthcare data management activities, including data conversions, arching, and exchange. Here I found many errors like incomplete data and a lot of spelling mistakes. And I calculated the age through date of birth and group the ages into categories. Depending on the system's parameters, it may be unable to search for duplicates as new patients come in. And that is where I could use algorithms to remove the attributes.

3 – Exploratory data analysis

Exploratory Data Analysis helps Data Analysts and researchers represent the data visually and dig patterns from data to obtain deep knowledge ingrained in the dataset. In the medical domain, data analysis primarily helps physicians and researchers in health care, where data about the patients is available in text and images. It helps to make the right choice in terms of cure and treatment; the analysis of the previous records of the patients helps most of the time. Once the data is cleaned, I could start searching for existing correlations in the data as well as identification of attributes that will likely be useful in the machine learning models.

4 – Data transformation

In this step, I transformed the multi-label problem into single-label problem by applying binary classifier chains and label powerset. A data model's structural components are typically conveyed schematically in drawings that use symbols and notations to denote the features and relationships among data items. I prioritized and integrated many internal and external data sources to provide better transparency into their population health journey. This transparency helps organizations manage risks, opportunities, and strategies to improve health outcomes.

5 – Implementing machine learning models

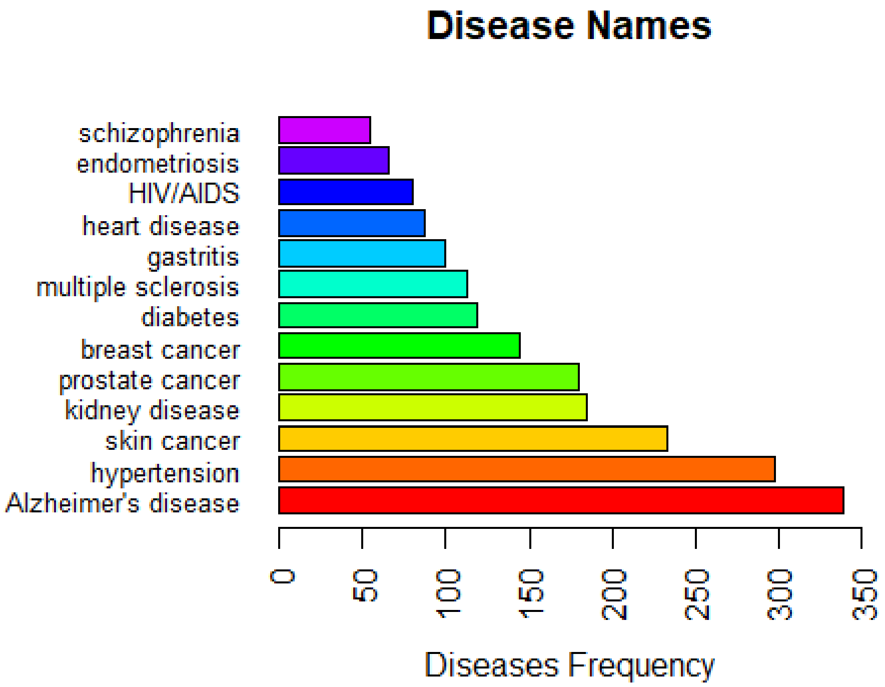
It is fair to start with this point because Machine Learning is very good at diagnosis; this is one of the most effective areas. Many types of cancer and genetic diseases are hard to detect; however, Machine Learning could handle many of them in the initial stages. In addition, machine Learning can offer predictive analytics to spot the best candidates for clinical trials based on factors like one’s history of doctor visits or social media activity. The technology will also lower the number of data-based errors and could suggest the best sample sizes tested. I used Naïve Bayes, k-nearest neighbors, and Random Forest to build a model to predict diseases given the inputs identified in the previous steps.

6 – Drawing conclusions and recommendations

In the end, there are situations when the aim is to maximize either recall or precision concerning the other metric. However, for the disease screening of patients, a recall near 1.0 is desirable; we can approve the need to find all patients with the disease and low precision. After applying classification's data mining techniques, Random Forest, Naïve Bayes, Logistic Regression, and applying balancing data algorithms, we see how the attributes make the under-sample, over-sample, and SMOTE attributes for three classes, Alzheimer's, Hypertension, and Skin Cancer diseases. The results show the above experiments in the tables after comparing the correctly classified disease percent. I found that balancing requires both under-sample and SMOTE, which gave very close numbers. The aim was to get the best recall, accuracy, and minimum variance numbers between iterations; we could see the best results for all three diseases analysis that both Logistic Regression with Under-Sampling algorithms.

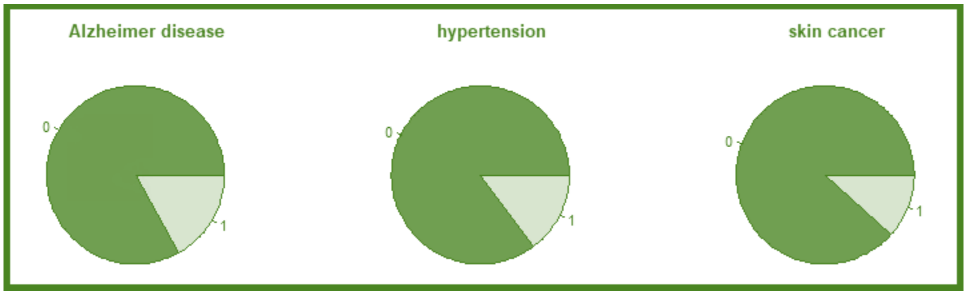
Data Preparation

Clinical study data can enhance health care experiences for individuals. It enables the expansion of knowledge about diseases and treatments and leads to an increase in the efficiency and effectiveness of health care systems. In this section, I explained the patient data based on their demographics and how they connect themselves to various diseases. I described some anomalies like an imbalanced data problem and solved them.



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

Figure 1: Pie chart 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, Figure 1 above shows the number of instances not diagnosed is around 85% and approximately 15% for diagnosed. Figures 2, 3, and 4 below show the variables' partition. The accuracy of the patient data depends on the class variable.

Figure 2: (Alzheimer Disease) Accuracy among the attributes of the patient data

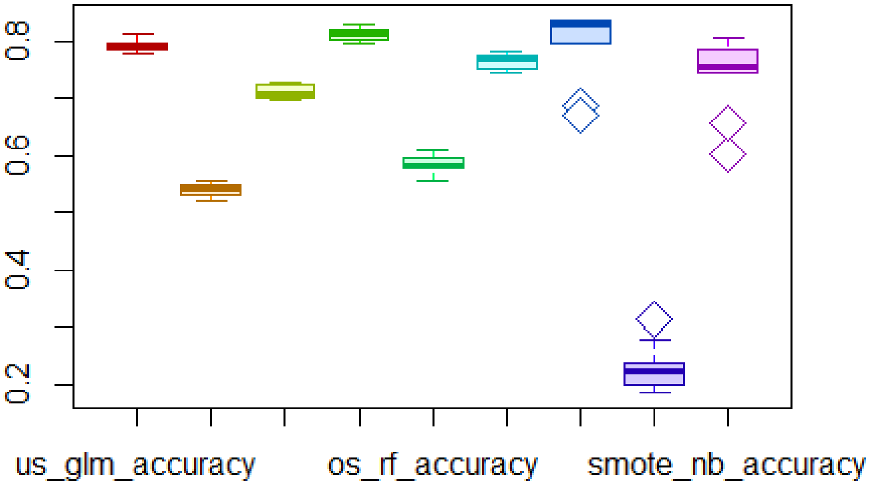


Figure 3: (Hypertension Disease) Accuracy among the attributes of the patient data

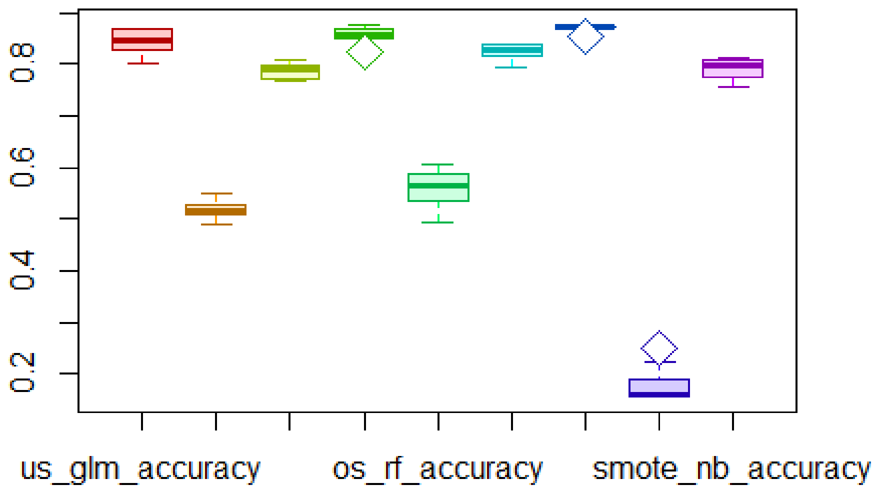
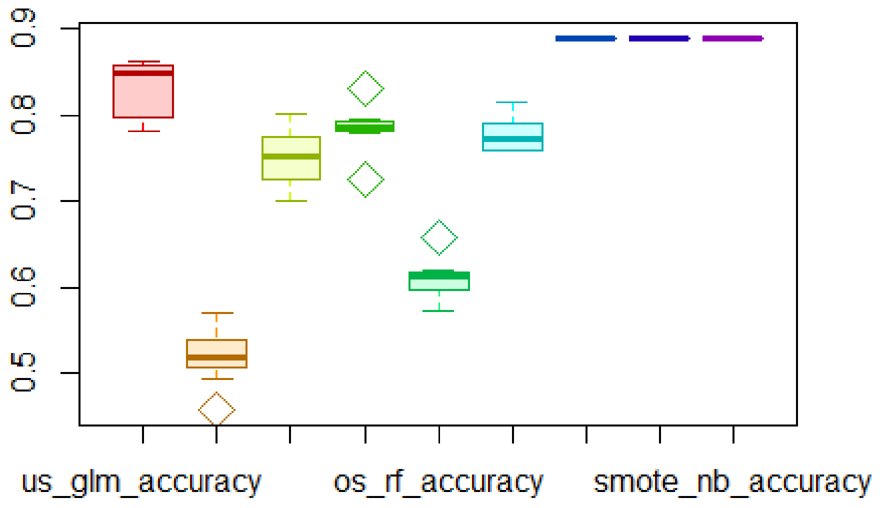


Figure 4: (Skin Cancer Disease) Accuracy among the attributes of the patient data



Working with this kind of data without solving the imbalanced data problem leads to bias, and the results become inclined towards the majority class. They can impact the decisions taken by data scientists. Imbalanced data problem and their resolution show some techniques and algorithms used to resolve the imbalanced data problem.

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 status	Input	Categorical	Employed
				Retired
				Student
				Unemployed
6	Education	Input	Categorical	Bachelors
				High school
				Masters
				PhD/Md
7	Marital status	Input	Categorical	Married
				Single
8	Children	Input	Quantitative	
9	Ancestry	Input	Categorical	Central Europe
				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

I analyzed their descriptive statistics and formed a correlation matrix for the numerical variables. Then, I used the Chi-squared method for the categorical/nominal variables to compute the correlation between those variables. Chi squares measure the strength of the association between one ordinal/nominal /categorical variable with another ordinal/nominal/categorical variable.

Extensive Data Analysis

I stared off by finding out if any Null values are present in any of the column, if FALSE then no null value(s). The were not missing values

```
gender                 False
dob                    False
employment status      False
education              False
marital status         False
children               False
ancestry               False
avg commute            False
daily internet use     False
available vehicles     False
military service       False
disease                False
```

Dealing with the Imbalance

From the exploratory analysis above the dependent variable is imbalanced. There are many alternatives to tackle this problem. Recent developments in the digitalization of medical systems have resulted in storing a sizable volume of health record data. However, it is not always simple to assess data, especially when the population's prevalence of the target condition is too low. The imbalanced data problem refers to this circumstance. Two strategies for balancing out an imbalance between minority and majority cases are over-sampling and under-sampling, which can be combined into ensemble algorithms. These methods, meanwhile, do not work when there are few minority examples overall.

- Over-sampling

According to research, patient diagnosis and treatment cycles frequently deviate considerably from established therapeutic paths. A further boost in the over-sampling from the standard of care, increased patient safety, higher levels of satisfaction from patients, and application optimization could result from examining these aberrations. The improved accessibility of reliable data from hospital information systems allows for a better understanding of routing behavior and deviations.

- Under-sampling

Patients at risk should undergo screening for a correct diagnosis as soon as possible. But sadly, because of under sampling the documented medical facts are frequently out of balance. Because of this, automated data processing is challenging or even impossible. Considering this, I aimed to find the most effective way to deal with the imbalance issue regarding the patients' analyzed data.

- Synthetic Minority Over-Sampling Technique (SMOTE) Sampling

The concept behind SMOTE, or synthetic oversampling, originates from the possibility that it may be simpler to particularize than to generalize a model. Thus, a simpler generating model can provide the diversity required for a more sophisticated classifying model.

Although a safe area can be found inside the minority class region without any majority samples, the fundamental problem with synthetic oversampling methods is that too many new instances there can actually reduce the model's accuracy for borderline cases. The borderline situations won't cause the model to experience enough loss, making it less accurate at identifying them. It goes against what we want to achieve.

- Cost Sensitive Learning

Health managers and clinicians frequently require models that aim to reduce a variety of healthcare expenses, such as attribute costs (such as the price of a particular diagnostic procedure) and misclassification costs (e.g., the cost of a false negative test). The utilization of unneeded resources and patient safety concerns are only two examples of how diagnostic tests and the misclassification errors connected to them can have a major financial or human cost. It is challenging for classifiers to properly understand and differentiate between the minority and majority classes due to the problem of class imbalance—techniques like resampling and cost-sensitive learning address the class imbalance issue. This project focuses on creating dependable, cost-sensitive classifiers that can effectively predict medical diagnoses by changing the objective functions of specific well-known algorithms, including logistic regression, decision tree, extreme gradient boosting, and random forest.

EDA using feature selection

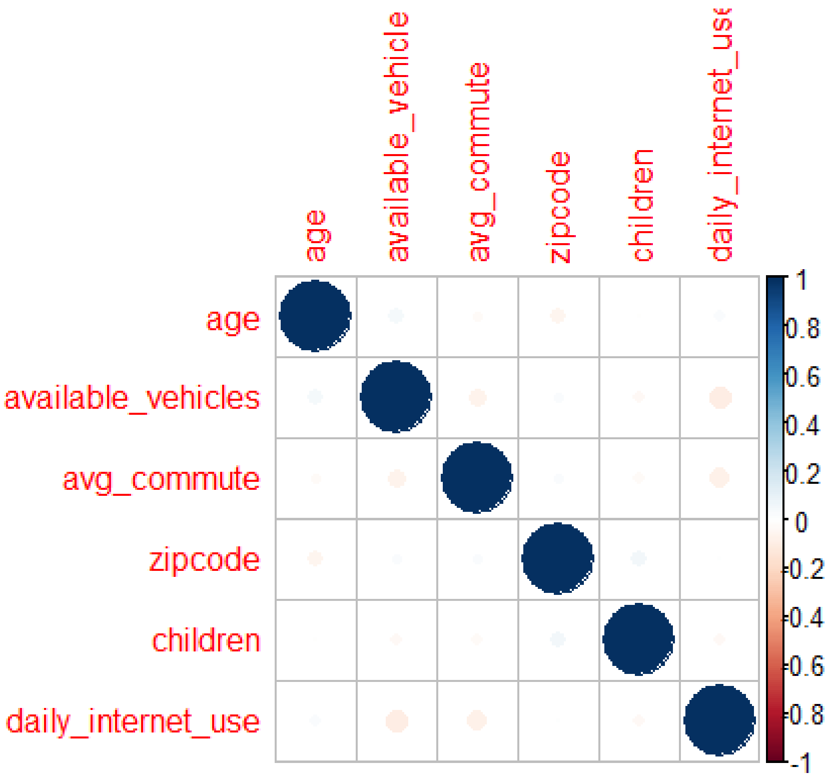
A robust system for supporting medical diagnostics has emerged: machine learning. A set of characteristics that are indicative of all the different illness manifestations is required in a medical diagnosis problem. This project aims to identify the presence of cardiovascular illness more precisely using fewer variables. I looked into intelligent systems that could generate feature subsets with improved diagnostic capabilities. To find a subset that provides an improved classification result, features ranked with distance measure are searched using forward inclusion, forward selection, and backward elimination search approaches. This experiment shows that, compared to forwarding inclusion and back-elimination strategies, this strategy finds smaller subsets and improves the diagnostic precision. Here is used feature selection using Chi-squared method after splitting and balancing the dataset for the topmost disease - Alzheimer's.

Feature the selection Alzheimer using Chi-squared

```
alzheimer_set <- select(patients, gender, age, employment_status, education,
marital_status, ancestry, available_vehicles, avg_commute,zipcode, children,d
aily_internet_use,military_service, alzheimer)
FeatureTrain <- sample(nrow(alzheimer_set), 0.7*nrow(alzheimer_set), replace
= FALSE)
FeatureTrainSet <- alzheimer_set[FeatureTrain,]
FeatureTestSet <- alzheimer_set[-FeatureTrain,]

response <- as.factor(patients$alzheimer)
input <- select(patients, gender, age, employment_status, education, marital_
status, ancestry, available_vehicles, avg_commute,zipcode, children,daily_int
ernet_use,military_service)

ubOver <- function(X, Y, k = 0, verbose=TRUE) {

}
data <- ubOver(X=input, Y=response)
alzheime_os_dataset <- cbind(data$X, class=data$Y)

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$gender)

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  alzheime_os_dataset$class and alzheime_os_dataset$gender
## X-squared = 0.030121, df = 1, p-value = 0.8622

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$age)

## Warning in chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$age):
## Chi-squared approximation may be incorrect

##
##  Pearson's Chi-squared test
##
## data:  alzheime_os_dataset$class and alzheime_os_dataset$age
## X-squared = 12.362, df = 3, p-value = 0.006241

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$education)

##
##  Pearson's Chi-squared test
##
```

```
## data:  alzheime_os_dataset$class and alzheime_os_dataset$education
## X-squared = 1.2066, df = 3, p-value = 0.7514

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$marital_status)

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  alzheime_os_dataset$class and alzheime_os_dataset$marital_status
## X-squared = 8.6472, df = 1, p-value = 0.003276

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$zipcode)

##
##  Pearson's Chi-squared test
##
## data:  alzheime_os_dataset$class and alzheime_os_dataset$zipcode
## X-squared = 48.141, df = 12, p-value = 2.953e-06

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$employment_status)

##
##  Pearson's Chi-squared test
##
## data:  alzheime_os_dataset$class and alzheime_os_dataset$employment_status
## X-squared = 37.411, df = 3, p-value = 3.767e-08

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$children)

##
##  Pearson's Chi-squared test
##
## data:  alzheime_os_dataset$class and alzheime_os_dataset$children
## X-squared = 17.862, df = 7, p-value = 0.01261

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$ancestry)

##
##  Pearson's Chi-squared test
##
## data:  alzheime_os_dataset$class and alzheime_os_dataset$ancestry
## X-squared = 17.201, df = 3, p-value = 0.0006427

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$avg_commute)

## Warning in chisq.test(alzheime_os_dataset$class,
## alzheime_os_dataset$avg_commute): Chi-squared approximation may be
## incorrect

##
##  Pearson's Chi-squared test
##
```

```
## data:  alzheime_os_dataset$class and alzheime_os_dataset$avg_commute
## X-squared = 2853.5, df = 1520, p-value < 2.2e-16

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$daily_internet_use)

## Warning in chisq.test(alzheime_os_dataset$class,
## alzheime_os_dataset$daily_internet_use): Chi-squared approximation may be
## incorrect

##
##  Pearson's Chi-squared test
##
## data:  alzheime_os_dataset$class and alzheime_os_dataset$daily_internet_us
e
## X-squared = 1612.5, df = 573, p-value < 2.2e-16

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$available_vehicles)

##
##  Pearson's Chi-squared test
##
## data:  alzheime_os_dataset$class and alzheime_os_dataset$available_vehicle
s
## X-squared = 8.9663, df = 4, p-value = 0.06195

chisq.test(alzheime_os_dataset$class, alzheime_os_dataset$military_service)

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  alzheime_os_dataset$class and alzheime_os_dataset$military_service
## X-squared = 2.1493, df = 1, p-value = 0.1426

alzheime_os_dataset %>%
  filter(class == "1") %>%
  select_if(is.numeric) %>%
  cor() %>%
  corrplot::corrplot()
```



Descriptive statistics and correlation the variables of the patient data

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

Given that parity cannot decrease with age, we can anticipate a positive linear relationship between age in years and parity. Still, we are unable to forecast how strong this association will be. Quantifying the association's strength is the problem at hand. In other words, since the direction in this situation is evident, we are more interested in the strength of the link between the two variables. Parity is ordinal and skewed, whereas maternal age is continuous and typically skewed. Correct correlation coefficient is used with these data measuring scales. The following are the standard deviation and variance measures for different attributes.

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 from the imbalanced data problem.

Discretization of the continuous quantitative attributes: Age and Ancestry

Since variables are not divided into classes, they are discrete, enabling them to be used with the others using the same methods. This will provide greater simplicity and readability.

There is a threshold of 65 years. No other threshold is clear. So, I decided to have the first band for people where Age lower than 25 and one band for the senior people (Age greater than 65), but make sure to have enough data in these bands. After that, I divided into 3 bands of equal size, the 'Age' between 25 and 65 years old. Finally, I got five bands.

Table 3: Age categories

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

The number of countries for ancestry is large; since all the countries are from Europe, the countries were grouped based on European ethnicities. So, finally, I made the ancestry group into four.

Table 4: Ancestry categories

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	Central Europe

Correlation of the nominal attributes of the patients’ data

Correlation measures the strength of the association between two nominal values, giving a value between 0 and +1. It is based on Pearson’s chi-squared statistic. I 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 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.

Guide to Association and Correlation Coefficients

Table 5: 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

Results and Interpretations

Tables 6, 7, and 8 show the variables' results and the patients' data target. It seems that the association between the most qualitative variables is either not generally applicable (values are between 0.0 and 0.15) or redundant (values are between 0.45 and 0.99). So, no use in making filter selections when running the models.

Table 6: Calculating the correlation attributes against Alzheimer disease

Variables	Correlation 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
Employment status	0.1061
Zip code	0.1204
Daily internet use	0.6967
Avg commute	0.9268

Table 7: Calculating correlation attributes against Hypertension disease

Variables	Correlation measure
Military service	0.0031
Employment 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
Zip code	0.1151
Daily internet use	0.6867
Avg commute	0.9367

Table 8: Calculating the correlation attributes against Skin Cancer disease

Variables	Correlation measure
Gender	0.0011
Military service	0.0126
Education	0.0175
Marital status	0.0217
Ancestry	0.0312
Age	0.0777
Zip code	0.0849
Available vehicles	0.0949
Employment status	0.1082
Children	0.1133
Daily internet use	0.7354

Avg commute	0.9498

Imbalanced data problem of patients' data and its resolution

I mentioned the problem of imbalanced patients' data in section 1.1 (description of patients' data). This section describes the imbalanced situation of data and defines some techniques and algorithms to solve the imbalanced data problem. Mainly, it will present three algorithms that will use to resolve the imbalanced problem of patients' data.

Imbalanced data problem

Classical machine learning algorithms assume that the number of objects in considered classes is roughly similar. In many real-life situations, the examples is skewed since representatives of some classes appear much more frequently. The classes with the more significant number of instances are called majority classes, and the classes with the smaller number of cases 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 many majorities class examples, a classification model tends to favor majority classes while incorrectly classifying the samples from the minority classes. In the real world, we will face the imbalanced data problem in many data independently the fields of research. The data imbalance problem is classified into two categories: a binary imbalanced data problem (binary class) and a multi-class imbalanced data problem (multi-class).

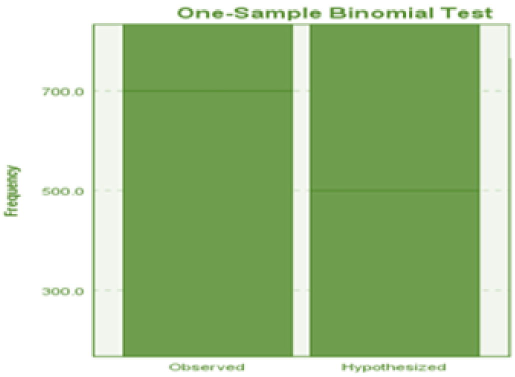
I used the One-Sample Binomial Test to confirm my observation concerning the presence of an imbalanced class. A One-Sample Binomial Test tests whether a proportion from a single dichotomous variable equals a presumed population value.

Table 9 below summarizes our imbalanced patients' dataset test. But all other diseases will follow the same solution scenarios.

Table 9: 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.
Result - Asymptotic significances are displayed. The significance level is .050.				

Figure 5: One-Sample Binomial Test



Handling imbalanced Datasets: Resampling techniques

Learning from imbalanced data has been a part of data analytics is active for about two decades in machine learning. A Data scientist facing this problem for the first time often asks, '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? It has no definite answer: it depends on the data.

My focus in this part is on resampling techniques because a resampling technique is a part of data preprocessing to deal with the imbalanced data problem. Indeed, data preprocessing plays a significant 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 must be proposed (Krawczyk, 2016). Algorithmic ensemble techniques are more a part of modeling algorithms.

Many techniques have been tried, with varying results and few clear answers. The main objective of balancing classes is to either increase the frequency of the minority class or decrease the frequency of the majority class. This is done to obtain approximately the same number of instances for both classes. Here are listed some techniques.

Naïve/Easy Approach: Do Nothing

The naïve approach consists of doing nothing. Sometimes you get lucky, and nothing needs to be done. For example, you can train on the so-called natural (or stratified) data; sometimes, it works without requiring modification.

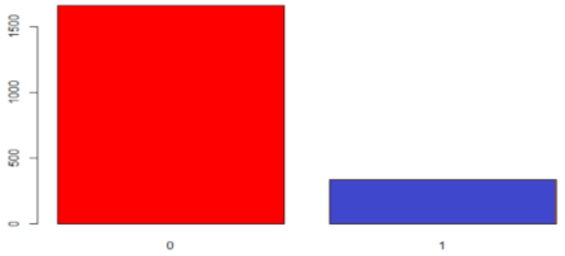
Under-sampling

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.

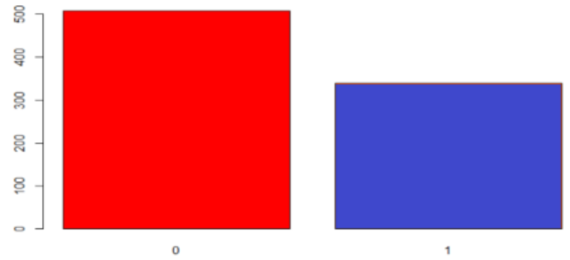
Advantage: It can help improve run time and storage problems by reducing the number of training data samples when the training data set is huge

Disadvantage: It can discard potentially useful information which could be essential for building rule classifiers. The sample chosen by random under-sampling may be biased. And it will not accurately represent the population, resulting in inaccurate results with the actual test data set.

Result after using under-sampling on the patients' data.



BEFORE USING UNDER-SAMPLING



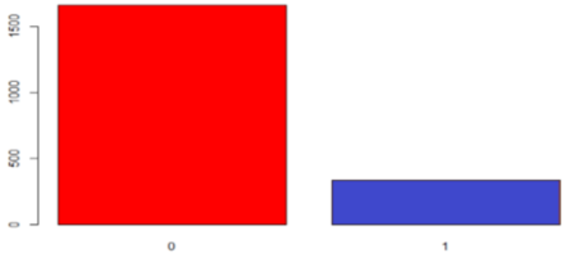
AFTER USING UNDER-SAMPLING

Over-sampling

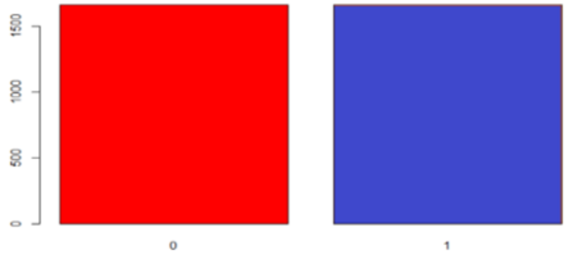
Over-Sampling increases the number of instances in the minority class by randomly replicating them to present a higher representation of the minority class in the sample.

Advantage: Unlike under-sampling, this method leads to no information loss. Outperforms under sampling

Disadvantage: It increases the likelihood of overfitting since it replicates the minority class events.



BEFORE USING OVER-SAMPLING



AFTER USING OVER-SAMPLING

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

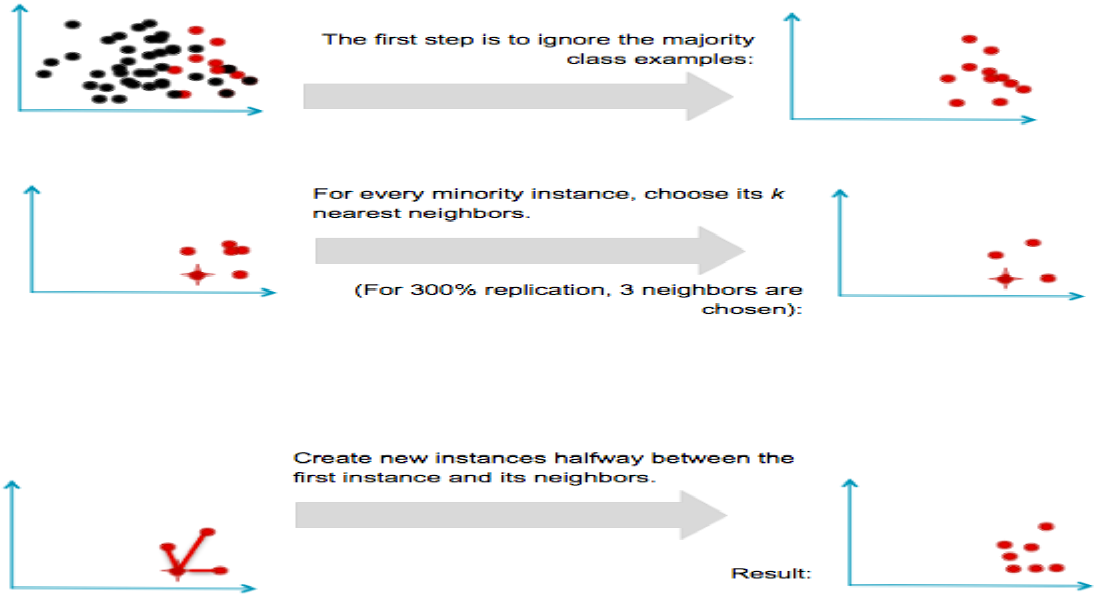
In this case, the K-means clustering algorithm is independently applied to minority and majority class instances. This is to identify clusters in the dataset. Subsequently, each cluster is oversampled such that all clusters of the same class have an equal number of instances, and all classes have the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

Advantages: This clustering technique helps overcome the challenge of class imbalance. The number of examples representing a positive class differs from the number 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.

Synthesizing new examples: SMOTE and descendants

This research direction has involved not resampling examples but synthesizing new ones. The best-known of this approach is SMOTE (Synthetic Minority Over-Sampling Technique). The idea is to create new minority examples by interpolating existing ones.

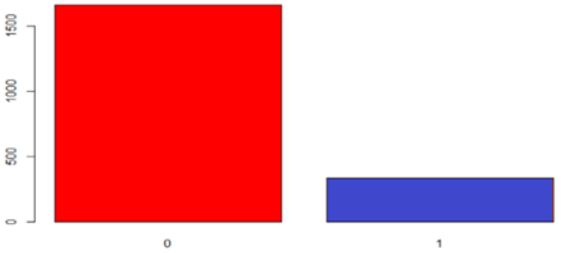

The first step is to ignore the majority class examples:
For every minority instance, choose its *k* nearest neighbors.
(For 300% replication, 3 neighbors are chosen):
Create new instances halfway between the first instance and its neighbors.
Result:

SMOTE successfully led to many variants, extensions, and adaptations to different concept learning algorithms; however, SMOTE has some advantages and disadvantages.

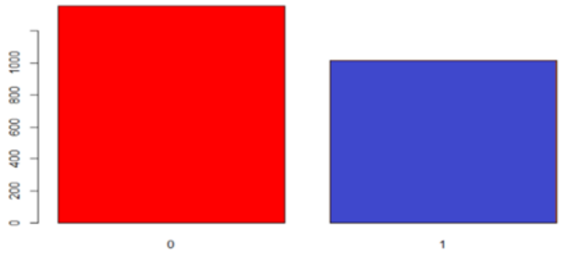
Advantages: Mitigates the problem of overfitting caused by random oversampling as synthetic examples are generated rather than replicating instances. No loss of valuable information

Disadvantages: While generating synthetic examples, SMOTE does not consider neighboring examples from other classes. This can increase the overlapping of classes and can introduce additional noise. As a result, SMOTE is not very practical for high-dimensional data.

BEFORE USING SMOTE



AFTER USING SMOTE



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 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 the most appropriate attributes to use for making 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 P-value. The importance of Pearson Statistics needs to be significant at 5%.

Table 10: 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
Employment status	3	0.0000	37.4110	3
Zip code	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 11: 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
Zip code	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
Employment 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 12: 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
Employment 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
Zip code	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

Predictive Modeling/Classification

In the analysis, I used three different classification algorithms 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 uses Under-sampling data, the second dataset uses over-sampling data, and the third dataset uses using SMOTE algorithm. For each evaluation, I ran ten different iterations.

Using R, the data was split into Train data (70% data) and Test data (30% data). Then, 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 for testing and train on the remaining 9. This gives 10 evaluation results, which are averaged.

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

Advantages: 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.

Random Forest Classifier

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

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

Advantages: It can handle binary, categorical, 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 a method for balancing error in class population unbalanced 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 tuning the hyperparameters is required. For extensive data sets, the size of the trees can take up a lot of memory.

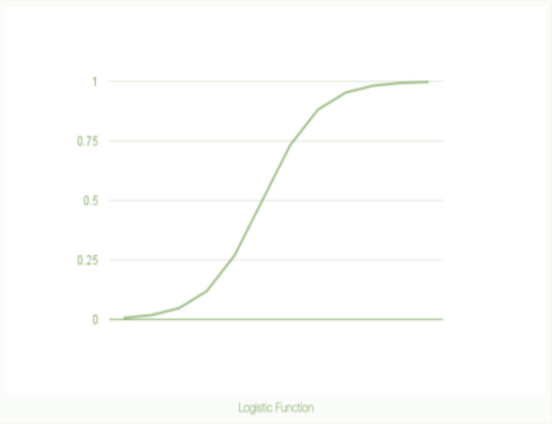
Logistic Regression Classifiers

A machine learning algorithm that is based on the concept of probability is used for classification and used to do predictive analysis. It makes linear regression do probabilities. Measures the relationship between a dependent variable and 1 or more independent variables; they never get below 0 or above 1; it has a smooth transition between 0 and 1.

Differs from linear regression in using a more complex cost function known as a sigmoid function or the 'logistic function.' Linear regression minimizes a squared error, while logistic regression maximizes the probabilistic function known as the 'log likelihood' function. It limits the cost function between 0 and 1. Sigmoid maps predictions to probabilities. It gives a 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 classifies values into Class 1 and below, which classifies values into the other Class 2. The 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 challenging to find the global minimum. Gradient Descent is used to reduce the cost value function. The gradient descent function is applied to each parameter. It works better when attributes unrelated to the output variable and attributes are correlated to each other.

Feature Engineering plays a vital role in logistic regression. For example, the logistic function looks like the following:



(Source: “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 straightforward, it doesn’t require too many computational resources. Highly interpretable. It 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 definite outcome as its outcome is discrete. Depends highly on the proper presentation of data. Can be easily outperformed by more complex algorithms.

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= TP/(TP+FN)
- true negative rate (specificity): TN rate= TN/(TN+FP)
- false positive rate: FP rate= FP/(TN+FP)
- false negative rate: FN rate= FN/(TP+FN)
- positive predictive value (precision) : PP value= TP/(TP+FP)
- negative predictive value: NP value= 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 β=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.

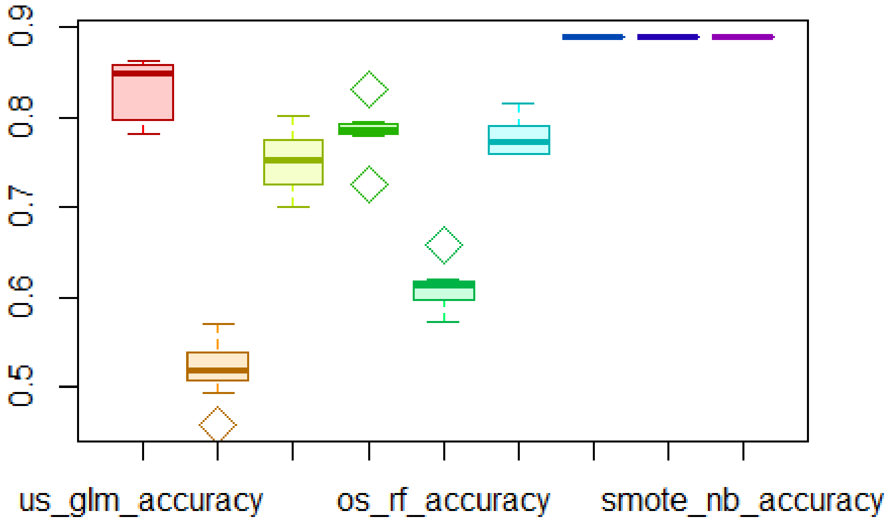
The receiver operating characteristics (ROC) curve and the corresponding area under the ROC curve (AUC) are also two popular tools used in imbalanced domains. The AUC allows the evaluation of the best model on average.

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

This was how I showed a box plot and the confusion matrix

```
c1 <- rainbow(10)
c2 <- rainbow(10, alpha=0.2)
```

```
c3 <- rainbow(10, v=0.7)
boxplot(df, col=c2, medcol=c3, whiskcol=c1, staplecol=c3, boxcol=c3, outcol=c
3, pch=23, cex=2)
```



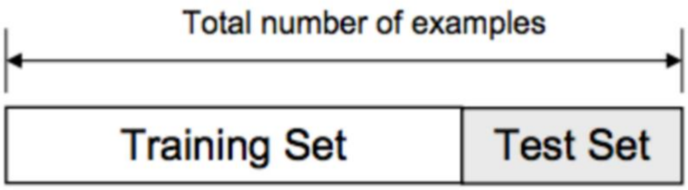
```
mean(us_nb_accuracy)

## [1] 0.89

mean(us_nb_precision)

## [1] 0.89

mean(us_nb_recall)

## [1] 1

mean(us_nb_f1)

## [1] 0.9417989

mean(os_nb_accuracy)

## [1] 0.89

mean(os_nb_precision)

## [1] 0.89

mean(os_nb_recall)
```

```
## [1] 1

mean(os_nb_f1)

## [1] 0.9417989

mean(smote_nb_accuracy)

## [1] 0.89

mean(smote_nb_precision)

## [1] 0.89

mean(smote_nb_recall)

## [1] 1

mean(smote_nb_f1)

## [1] 0.9417989

a <- matrix(
  c(mean(us_glm_accuracy),mean(us_glm_precision),mean(us_glm_recall),mean(us_
glm_f1),
    mean(os_glm_accuracy),mean(os_glm_precision),mean(os_glm_recall),mean(os_
glm_f1),
    mean(smote_glm_accuracy),mean(smote_glm_precision),mean(smote_glm_recall)
,mean(smote_glm_f1)),
  nrow=3,
  ncol=4,
  byrow = TRUE
)

a

##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.8298333 0.8921490 0.9200375 0.9054176
## [2,] 0.5206667 0.9308333 0.4985019 0.6485074
## [3,] 0.7510000 0.8930433 0.8181648 0.8535891
```

Running the datasets against the algorithms

Train/Test Split

The data 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 to be generalized to other data later. The test dataset is there to test the model’s prediction on this subset.



Training and Testing Hypertension using Chi-squared

```
hypertension_set <- select(patients, gender, age, employment_status, educatio
n, marital_status, ancestry, available_vehicles, avg_commute,zipcode, childre
n,daily_internet_use,military_service, hypertension)
FeatureTrain <- sample(nrow(hypertension_set), 0.7*nrow(hypertension_set), re
place = FALSE)
FeatureTrainSet <- hypertension_set[FeatureTrain,]
FeatureTestSet <- hypertension_set[-FeatureTrain,]

response <- as.factor(patients$hypertension)
input <- select(patients, gender, age, employment_status, education, marital_
status, ancestry, available_vehicles, avg_commute,zipcode, children,daily_int
ernet_use,military_service)

data <- ubOver(X=input, Y=response)
hypertension_os_dataset <- cbind(data$X, class=data$Y)

chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$gender)

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$gender
## X-squared = 6.2623, df = 1, p-value = 0.01233

chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$age)

## Warning in chisq.test(hypertension_os_dataset$class,
## hypertension_os_dataset$age): Chi-squared approximation may be incorrect

##
##  Pearson's Chi-squared test
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$age
## X-squared = 35.943, df = 3, p-value = 7.698e-08

chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$education)

##
##  Pearson's Chi-squared test
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$education
## X-squared = 1.463, df = 3, p-value = 0.6908
```

```
chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$marital_sta
tus)

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$marital_s
tatus
## X-squared = 1.2361, df = 1, p-value = 0.2662

chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$zipcode)

##
##  Pearson's Chi-squared test
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$zipcode
## X-squared = 45.135, df = 12, p-value = 9.771e-06

chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$employment_
status)

##
##  Pearson's Chi-squared test
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$employmen
t_status
## X-squared = 0.81971, df = 3, p-value = 0.8447

chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$children)

##
##  Pearson's Chi-squared test
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$children
## X-squared = 36.927, df = 7, p-value = 4.842e-06

chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$ancestry)

##
##  Pearson's Chi-squared test
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$ancestry
## X-squared = 2.9499, df = 3, p-value = 0.3994

chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$avg_commute
)

## Warning in chisq.test(hypertension_os_dataset$class,
## hypertension_os_dataset$avg_commute): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$avg_commu
te
## X-squared = 2986.7, df = 1521, p-value < 2.2e-16
```

```
chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$daily_inter
net_use)
```

```
## Warning in chisq.test(hypertension_os_dataset$class,
## hypertension_os_dataset$daily_internet_use): Chi-squared approximation may
## be incorrect

##
##  Pearson's Chi-squared test
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$daily_int
ernet_use
## X-squared = 1605.3, df = 573, p-value < 2.2e-16
```

```
chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$available_v
ehicles)
```

```
##
##  Pearson's Chi-squared test
##
## data:  hypertension_os_dataset$class and hypertension_os_dataset$available
_vehicles
## X-squared = 2.3449, df = 4, p-value = 0.6726
```

```
chisq.test(hypertension_os_dataset$class, hypertension_os_dataset$military_se
rvice)
```

Logistic regression analysis

The goal of logistic regression is to provide a formula for calculating the likelihood that a particular result will occur for each patient participating in a study. For example, I could predict this likelihood to equal only the success rate in the study group. By modifying the predicted probability for a given patient by a set of covariate values, logistic regression seeks to outperform this prior probability. The accuracy with which statistical models for the clinical prediction can forecast outcomes for brand-new patients must serve as the primary evaluation standard. This method includes splitting the entire patient population into two groups, a training set on which to base the model and a test set to assess the model's accuracy.

Alzheimer disease: Logistic Regression balanced dataset

	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**

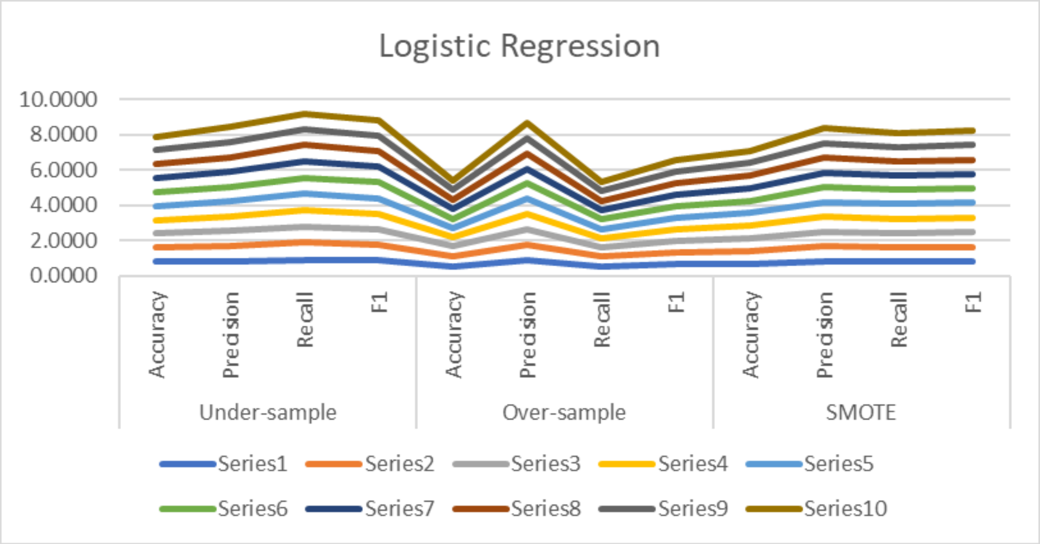
Alzheimer disease: Random Forest balanced dataset

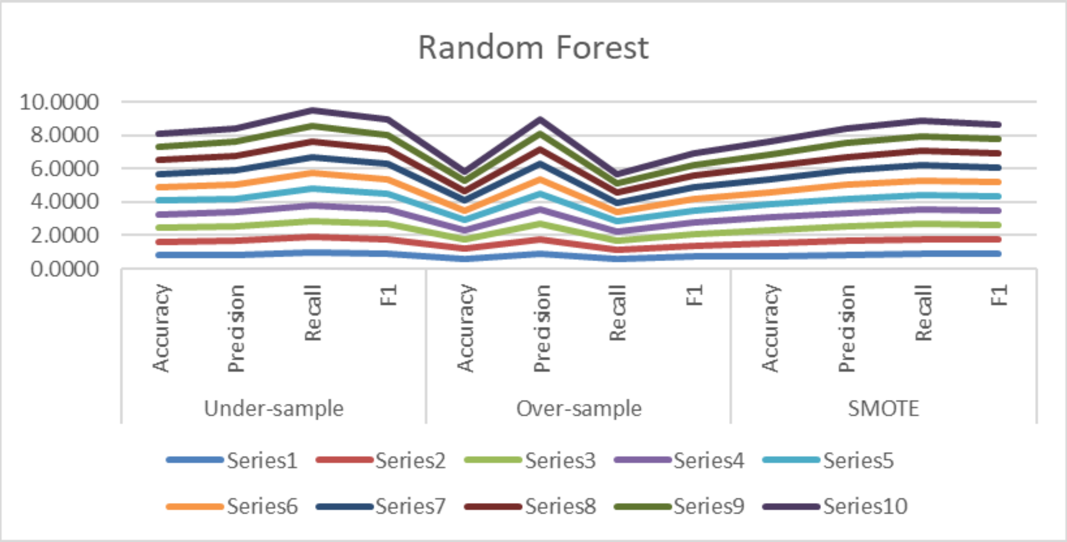
	Random Forest											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.8117	0.8394	0.9582	0.8949	0.5967	0.8939	0.5876	0.7091	0.7650	0.8412	0.8865	0.8632
2	0.8183	0.8441	0.9602	0.8984	0.5783	0.8952	0.5618	0.6903	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**

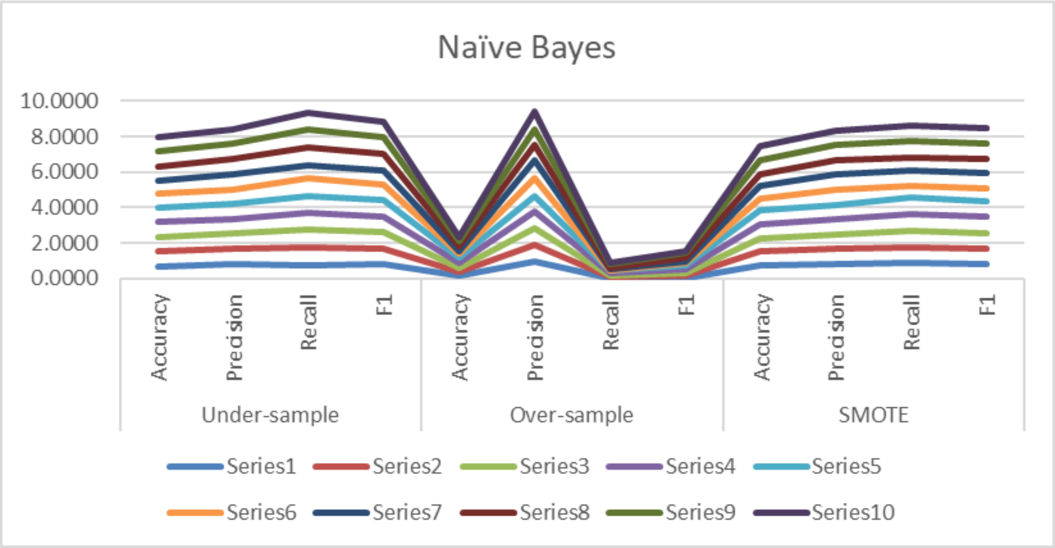
Alzheimer disease: Naïve Bayes balanced dataset

	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**

Alzheimer disease: Graph show the models differences







Hypertension disease: Logistic Regression balanced dataset

	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**

Hypertension disease Random Forest balanced dataset

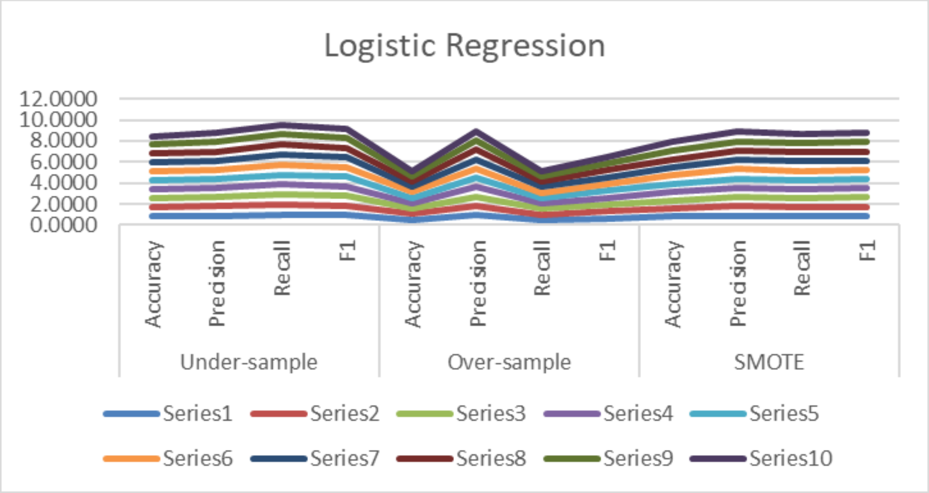
	Random Forest											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.8567	0.8895	0.9541	0.9207	0.5217	0.9436	0.4799	0.6362	0.8250	0.8899	0.9120	0.9008
2	0.8233	0.8897	0.9101	0.8998	0.4933	0.9294	0.4532	0.6093	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**

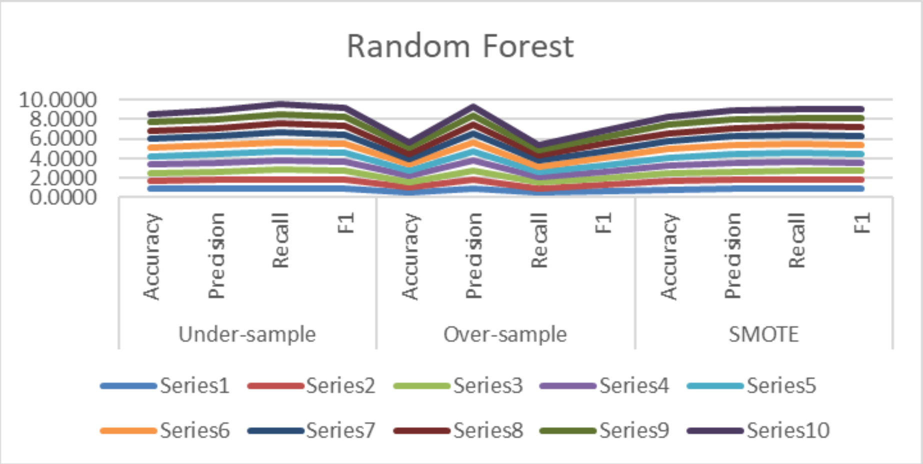
Hypertension disease: Naïve Bayes balanced dataset

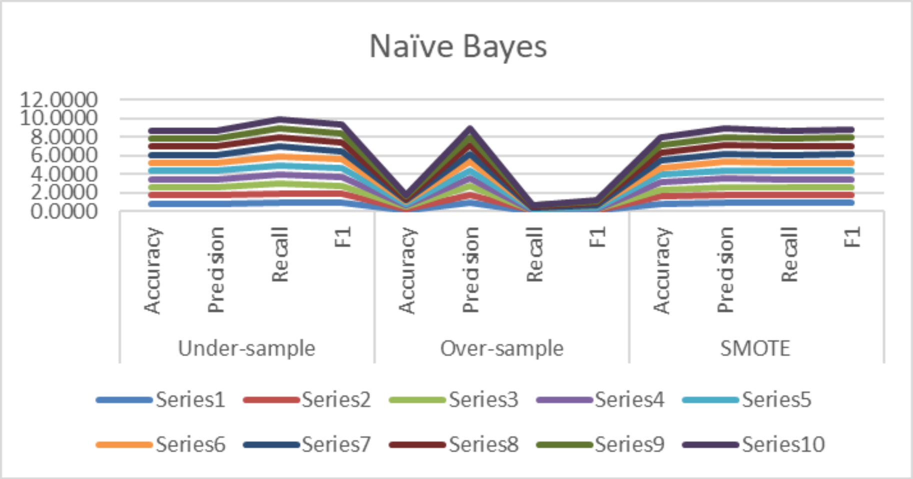
	Naïve Bayes											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.8717	0.8717	1.0000	0.9314	0.1567	0.9048	0.0363	0.0699	0.8117	0.8868	0.8987	0.8927
2	0.8550	0.8759	0.9713	0.9211	0.1567	0.9048	0.0363	0.0699	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**

Hypertension disease: Graph show the models differences







Skin Cancer disease: Logistic Regression balanced dataset

	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**

Skin Cancer disease: Random Forest balanced dataset

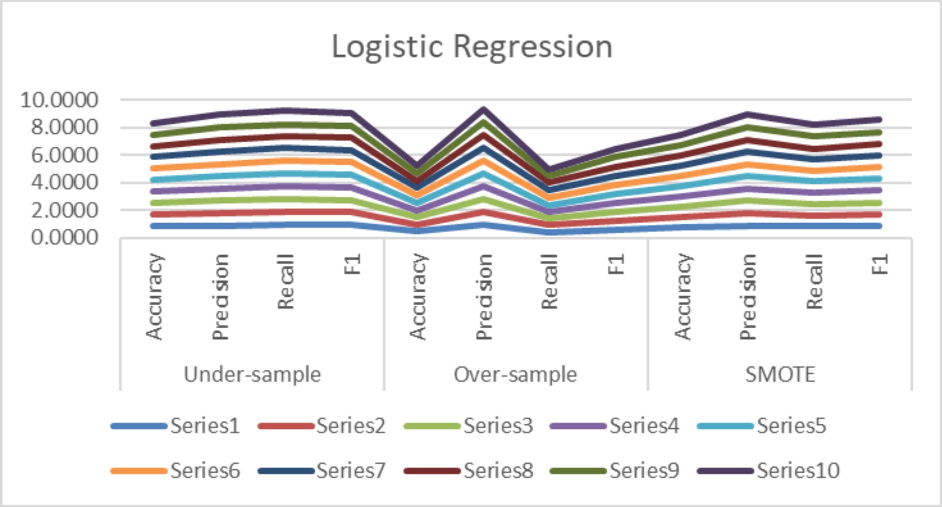
	Random Forest											
	Under-sample				Over-sample				SMOTE			
Iteration	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
1	0.7933	0.9133	0.8483	0.8796	0.6200	0.9527	0.6030	0.7091	0.7583	0.9027	0.8165	0.8574
2	0.7917	0.9082	0.8521	0.8792	0.6167	0.9419	0.6067	0.6903	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

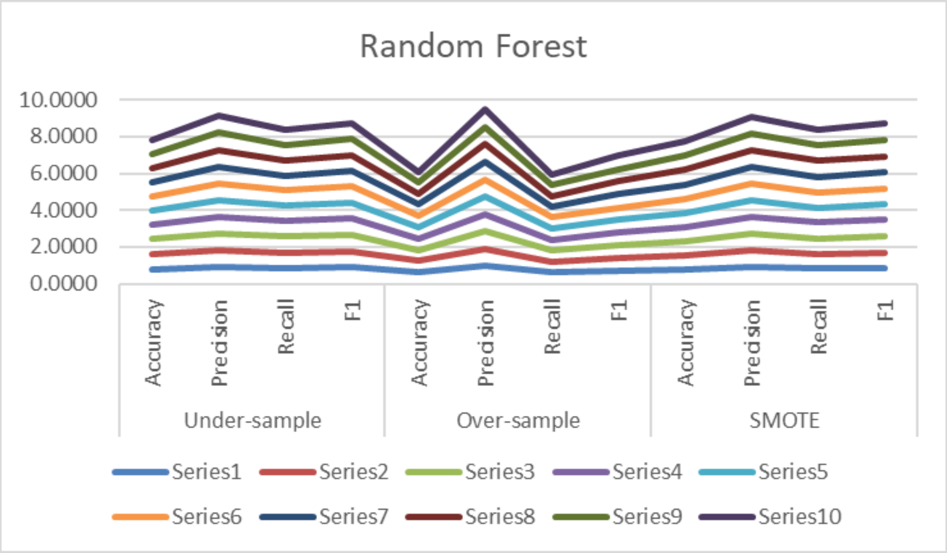
6	0.7950	0.9135	0.8502	0.8807	0.6583	0.9507	0.6498	0.6889	0.7717	0.9043	0.8315	0.8663
7	0.7267	0.9167	0.7622	0.8323	0.5933	0.9503	0.5730	0.7192	0.7750	0.9080	0.8315	0.8680
8	0.7800	0.9069	0.8390	0.8716	0.5717	0.9482	0.5487	0.6927	0.7900	0.8953	0.8652	0.8800
9	0.7817	0.9138	0.8333	0.8717	0.6117	0.9493	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**

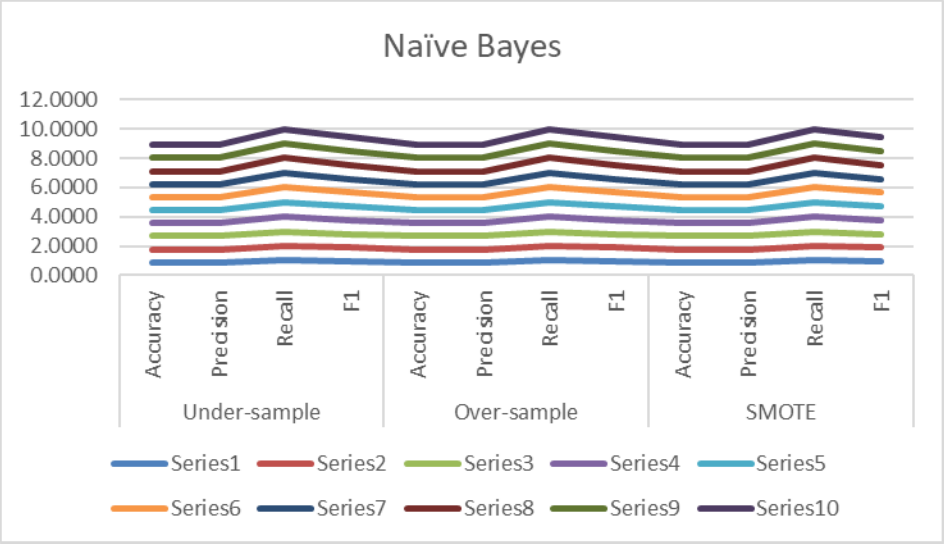
Skin Cancer disease: Naïve Bayes balanced dataset

	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**

Skin Cancers disease: Graph show the models differences







Results

There are situations when the aim is to maximize either recall or precision at the expense of the other metric. Medical professionals frequently consider specificity and sensitivity while assessing medical testing. These ideas are quite dissimilar yet similar at the same time. This disparity may lead to numerous misconceptions between the medical and data science communities when the two worlds collide, as when a medical test is a machine learning model. Precision and Accuracy are the same. However, Precision and Recall are different if we define a positive example as a person who has an illness. Making all predictions optimistic is not a good idea. The recall is 100% because all the positive cases are projected to be positive, but the precision is only somewhat high due to the population's imbalance. Nevertheless, because there isn't a negative case, the specificity is 0%. 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.

Recommendations

Machine learning will be used more and more in the healthcare industry due to the complexity and growth of data in that industry. Payers, care providers, and life sciences businesses currently use various ML techniques. The main application categories include recommendations for diagnosis and treatment, patient engagement and adherence, and administrative tasks. Although there are numerous situations where machine learning can do healthcare duties just as well as or better than humans, implementation issues will keep a significant portion of the workforce in the healthcare industry from becoming automated. In addition, the use of ML in healthcare and ethical concerns are also covered.

The data analysis based on ML algorithms threw varied and interesting results. However, we must add more attributes to the dataset for better results. Attributes involved when the patients migrated to the US to see the impact of the living habit on their disease. Also, blood type, blood pressure measurement, and other factors related to the historical blood tests. The quality of the data fed into machine learning algorithms determines your outcomes.

Unfortunately, the accuracy and standardization of medical data are not always as high as they should be. Records are incomplete, profiles are inaccurate, and there are other issues.

In general, the purpose of electronic health records was not to be a data source for an algorithm. Therefore, you would need to spend time acquiring, cleaning, validating, and organizing data for a machine learning tool's purpose before using it. While machine learning holds great promise for the healthcare sector, it also presents several obstacles, including the need for a large team of data professionals, physician-friendly product development, and healthcare data quality. There are also some ethical issues, including patient accountability and safety. Nevertheless, the advantages of ML in healthcare far exceed the drawbacks, notwithstanding some obstacles.

Conclusions

Researchers are interested in diseases, such as heart, Breast cancer, Diabetes, Alzheimer's, Parkinson's disease, and kidney disease, which correlate and depend on machine learning/deep learning-based techniques. Additionally, some other Machine Learning based disease diagnosis approaches are discussed as well. Many countries are worried about the privacy of patients' data and have also raised legal concerns about the ethics of AI and ML when used with real-world patient data. I pre-processed the patients' data, wherein I checked them for diseases. First, the data was balanced using Under-sample, Over-sample, and SMOTE. Then, I implemented machine learning algorithms such as Random Forest, Naive Bayes, and Logistic Regression on the pre-processed data. It gave interesting results, and I selected the best model based on Accuracy, Recall, and Positive Rate. As a result, future studies could try producing synthetic data instead of depending on data gathering and processing. In addition, future researchers and practitioners may be interested in some techniques to produce synthetic data for the experiment. And that brings me to take off with this data and provides the foundation for using this analysis and treating patients better.

REFERENCES

Brownlee, J. (2020, August 14). *Logistic regression for machine learning*. Machine Learning Mastery. Retrieved August 3, 2022, from https://machinelearningmastery.com/logistic-regression-for-machine-learning/

Chapple, M. (2020, July 20). *How classification helps make big data understandable*. Lifewire. Retrieved August 3, 2022, from https://www.lifewire.com/classification-1019653

Ethnicity & disease - researchgate.net. (n.d.). Retrieved August 3, 2022, from https://www.researchgate.net/journal/Ethnicity-Disease-1945-0826

Molnar, C. (2022, July 12). *Interpretable machine learning*. Christoph Molnar. Retrieved August 3, 2022, from https://christophm.github.io/interpretable-ml-book/

Mukaka, M. M. (2012, September). *Statistics corner: A guide to appropriate use of correlation coefficient in medical research*. Malawi medical journal : the journal of Medical Association of Malawi. Retrieved August 3, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3576830/

Narkhede, S. (2021, June 15). *Understanding confusion matrix*. Medium. Retrieved August 3, 2022, from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Smart, A., Bolnick, D. A., & Tutton, R. (2017, January 9). *Health and genetic ancestry testing: Time to bridge the gap*. BMC medical genomics. Retrieved August 3, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5223458/

Terry-Jack, M. (2019, May 1). *Tips and tricks for multi-class classification*. Medium. Retrieved August 3, 2022, from https://medium.com/@b.terryjack/tips-and-tricks-for-multi-class-classification-c184ae1c8ffc

U.S. National Library of Medicine. (n.d.). *Home - books - NCBI*. National Center for Biotechnology Information. Retrieved August 3, 2022, from https://www.ncbi.nlm.nih.gov/books

Yiu, T. (2021, September 29). *Understanding random forest*. Medium. Retrieved August 3, 2022, from https://towardsdatascience.com/understanding-random-forest-58381e0602d2