

***Predicting Human Disease using
imbalance Genomic Dataset***

EE997: Individual MSc Project

Name: Nilesh Dineshkumar Ohol – 202163901

Supervisor: Prof. Anil Fernando (Dept. of Computer
and Information Sciences)

Course: MSc Machine Learning and Deep Learning

The Department of Electronic and Electrical
Engineering

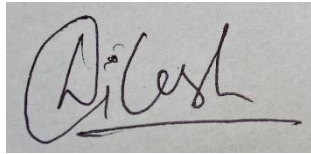
The University of Strathclyde

Date of Submission: 18th August 2022

Declaration of Authorship

I, Nilesh Dineshkumar Ohol, hereby declare that this work has not been submitted for any other degree/course at this University or any other institution and that, except where reference is made to the work of other authors, the material presented is original and entirely the result of my own work at the University of Strathclyde under the supervision of Prof. Anil Fernando.

Signed:

A handwritten signature in black ink, appearing to read 'Nilesh', with a horizontal line underneath.

Date: 18th August 2022

Abstract:Background:

Traditionally diseases are assessed by looking at their Symptoms by their family general practitioners. Although this method is effective, there is a slim chance that they might misinterpret a disease and mistreat it. Mistreating could result in being harmful to individuals receiving those treatments. But, understanding the genomic structure of the individuals could help us give a better view of the diseases that are likely to happen.

Methods:

We have used a Genomic sequencing dataset by Gene Ontology which has 11950 predictors of 18418 individuals. We will be predicting Heart Disease, Type II Diabetes, Alzheimer, Parkinson and Obesity. Since the target features of these diseases are imbalanced, we will be using upsampling methods ADASYN (Adaptive Synthetic) and SMOTE (Synthetic Minority Over-Sampling technique). We'll build logistic regression, random forest and neural network models for three versions of the dataset for each disease. Later the models were evaluated by looking at their Accuracy, Precision, Recall, F1 Score and ROCAUC Score using a confusion matrix.

Result & Conclusion:

We managed to build logistic regression, random forest and neural network model for each disease having different versions of the dataset with ROCAUC scores for the generic dataset, ADASYN dataset and SMOTE dataset between 0.49 to 0.63, 0.49 to 0.59 and 0.49 - 0.59 respectively.

Table of Content:

1. Introduction.....	3
2. Deoxyribonucleic (DNA) & Ribonucleic (RNA).....	5
3. Chromosomes & Gene.....	8
4. Gene Ontology.....	9
4.1. Gene Ontology Dataset.....	10
5. Diseases:	
5.1 Heart Disease.....	11
5.2 Type II Diabetes.....	13
5.3 Alzheimer's.....	16
5.4 Parkinson's.....	18
5.5 Obesity.....	21
6. Machine Learning Algorithms:	
6.1 Logistic Regression.....	23
6.1.1. Advantages & Disadvantages.....	25
6.2 Random Forest.....	26
6.2.1. Variety of Decision Tree.....	27
6.2.2. Quantifying Homogeneity.....	28
6.2.3. Information Gain.....	30
6.2.4. Advantages & Disadvantages.....	31
6.3 Neural Network.....	32
6.3.1 Activation Function.....	33
6.3.2 Evaluating Regression & Classification model.....	34
6.3.3 Binary Cross Entropy.....	37
6.3.4 Optimisation.....	38
6.3.5 Gradient Descent.....	40
6.3.6 Stochastic Gradient Descent.....	41
6.3.7 Backpropagation.....	42

7. Imbalanced Data.....	43
7.1 Upsampling Minority Class.....	44
7.2 Downsampling Majority Class.....	44
7.3 Generate Synthetic Data.....	45
7.4 Combine Oversampling & Undersampling.....	45
7.5 Balanced Class Weight.....	46
7.6 Adaptive Synthetic (ADASYN) Algorithm.....	46
7.6.1 Algorithm.....	48
7.6.2 Advantages & Disadvantages.....	50
7.7 K-Nearest Neighbours (KNN).....	47
7.8 Synthetic Minority Oversampling technique (SMOTE).....	50
7.8.1 Algorithm.....	51
7.8.2 Disadvantages.....	51
8. Results.....	51
9. Conclusion.....	53
10. References	57
11. Appendix A: GitHub Repository.....	69

Introduction:

This project intends to build meaningful Machine Learning Algorithms to predict diseases using datasets having genomic information from a credible source. According to a report, "The top 10 causes of death", published by the World Health Organisation (W.H.O.), 7 out of 10 leading causes of death in 2019 were non-communicable diseases. Ischaemic heart disease is alone responsible for 16% of the total deaths in the world. Also, the report says that for lower-middle-income countries, 5 out of 10 deaths are caused due to non-communicable diseases. The deaths from these diseases have nearly doubled since 2000[1]. One of the ways to detect these diseases is by looking at their symptoms. However, Symptoms take some time to develop.

Another way is to study an individual's genomic structure, observe the variations in the genomic sequence, and predict disease. Broad Institute researchers study individuals' genomic sequences, cumulate those and calculate the Polygenic Score (PGS) [2]. A polygenic score (PGS) or polygenic risk score (P.R.S.) is an estimated value that is calculated by using the genotype profile of an individual and relevant genomic-wide association study (GWAS) data [3]. Since these datasets have millions of data points to process, Here's where Machine Learning could help us. A report, "Using A.I. to find the disease-causing gene, " published at Stanford Medicine by Helen Santoro, mentions that A.I. helps scientists process thousands of datasets where each dataset could have up to a million data points. Also, specific software is used to scan through an organism or genome, which has the complete set of mice D.N.A. being used to model human diseases. They are used to identify the gene mutations that cause diseases [4].

In some cases, like clinical genomics, Deep Learning algorithms are used to process complex and humongous genomic datasets. A Convolutional Neural Network (CNN), used for image recognition and autonomous vehicles, is also used to predict diseases from genomic

data [5]. On 12th July 2022, World Health Organisation (W.H.O.) launched its first report calling for an equitable expansion of genomics [6]. Dr Soumya Swaminathan, WHO Chief Scientist, mentions that these Genomic technologies are driving some of the most ground-breaking research today. Yet the benefits of these tools are not fully utilized unless deployed worldwide. Through Equity, the science will reach its full potential impact and improve the health of individuals from anywhere [6]. W.H.O. wants to strengthen its international collaboration by requesting high-income countries to share their genomic data with low-income countries to promote research and development [7]. We'll be using a genomic dataset, an excel file with 18418 rows and 11953 columns. A label dataset has 29 columns and 20659 rows which includes 27 diseases like Alzheimer's, Sclerosis, Parkinson's, Brain Disease, Heart Disease, Immune Disease, Muscular Disease, Neoplasm, Nutrition, Artherosclerosis, Coronary Heart Disease, Hypertension, Myocardial infarction, Immune Hypersensitivity, Arthritis, Osteoporosis, Adenocarcinoma, Breast neoplasm, Colorectal neoplasm, Lung neoplasm, Prostatic neoplasm, Stomach Neoplasm, Diabetes type I, Diabetes type II, Obesity and Asthama. The genomic dataset has a feature "entrzId", which is unique ID column, 11951 features having particular GO (Gene Ontology) ID, which are genomes having the biological process, molecular function & cellular components, and the "go_total" feature has total occurrences those functions of an individual [8]. We'll use this data to predict five diseases, Heart Disease, Diabetes Type II, Alzheimer's, Parkinson's and Obesity. These diseases are the leading causes of death in the world. In 2019, 18.56 million, 1.53 million, and 362,907 people died worldwide due to Cardiovascular diseases, Diabetes and Parkinson's [9]. We'll understand more about these diseases further. The labels have imbalanced target features, hindering the model's performance. For example, the number of target variables having heart disease is 283, while someone not having heart disease is 17983(i.e. we have only 1.55% of predictors having heart disease). To overcome this situation,

we'll be using upsampling methods Adaptive Synthetic (ADASYN) and SMOTE (Synthetic Minority Over-sampling Technique).

We'll build three machine learning models, Logistic Regression, Random Forest and Neural Network, for each version of the dataset (i.e. Generic, ADASYN & SMOTE Dataset) for each disease and evaluate the performance.

Deoxyribonucleic Acid (DNA) & Ribonucleic Acid (RNA):

An average man comprises of 30 - 40 trillion human cells [10]. These cells are the building blocks of the human body and are responsible for structuring the body to convert food into energy to perform specialized functions [11]. Each cell has a cell membrane wall; within this wall, we have a cytoplasm which is a gelatinous liquid; within this liquid pool, we have mitochondria, lysosomes, ribosomes, nucleoli, Golgi apparatus & endoplasmic reticulum as shown in Fig.1 [12].

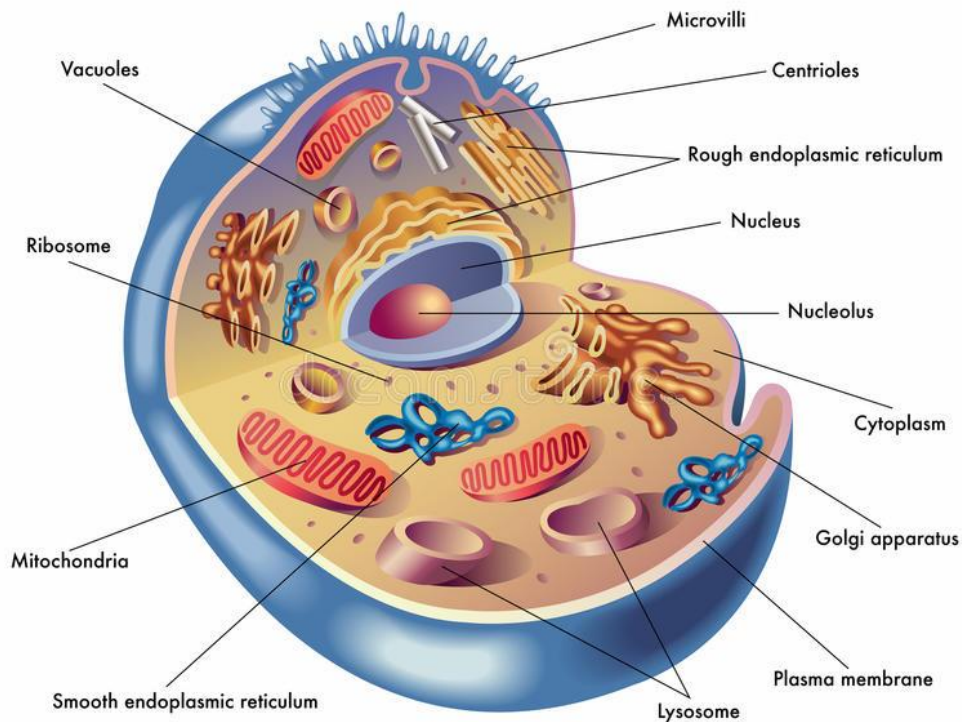


Fig.1. Human Cell [91]

The nucleoli are responsible for making ribosomes. These ribosomes are then transported to the cytoplasm to perform protein synthesis. Within the cytoplasm, these RNAs are transcribed (i.e., coded). Deoxyribonucleic acid(DNA) stores codes which multiple sequences of 4 chemical bases, namely: Adenine (A), Thymine (T), Cytosine (C)and Guanine (G) [14].

Human DNA has around 3 billion bases, and 99% of these bases are the same in all humans. The pair of these chemical bases form a unit called base pairs. A combination of a base, a sugar molecule and a phosphate molecule is called a nucleotide. These nucleotides are arranged on two long strands in a spiral to form a helical shape. To understand the structure, look at Fig. 2.

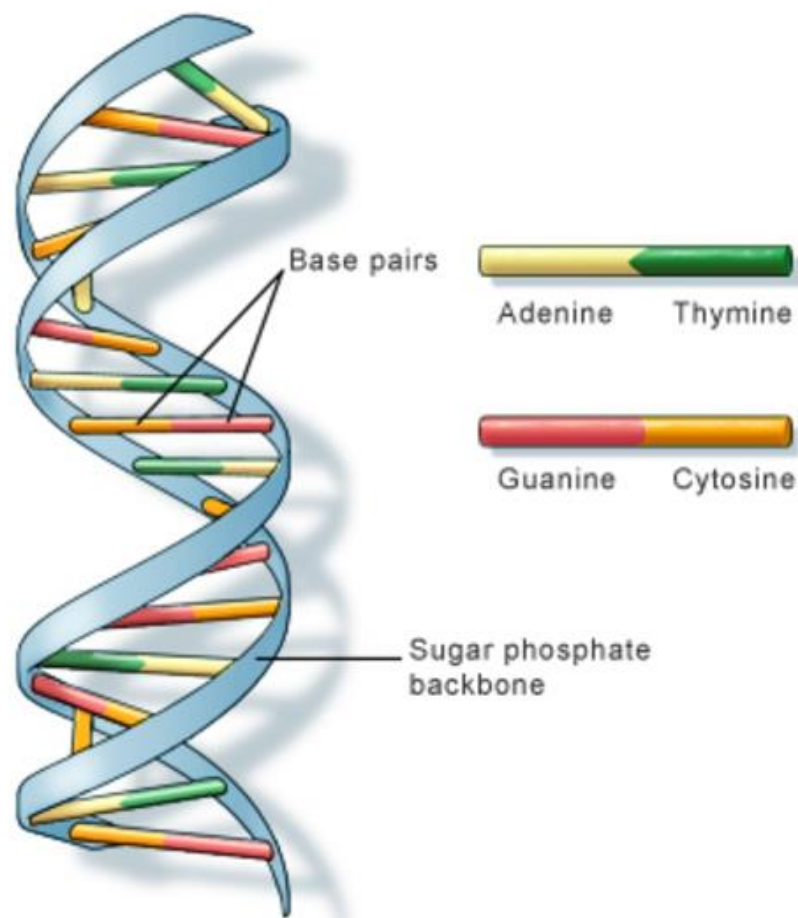


Fig.2. Deoxyribonucleic acid (DNA) [92]

This DNA can make copies of itself by duplicating the sequence of bases[14]. However, Unlike DNA, Ribonucleic Acid (RNA) is single-stranded and has alternate groups of phosphate molecules and sugar ribose. Multiple sequences of 4 chemical bases present in an RNA are Adenine (A), Uracil (U), Cytosine (C) and Guanine (G) [14]. They are responsible for the construction of a cell and responding with immunity, and carrying amino acids from one part of the cell to the other. To understand this structure, you may look at Fig.3. Different forms of RNA have specific functionality. The different form of RNA includes mRNA, tRNA, and rRNA [15].

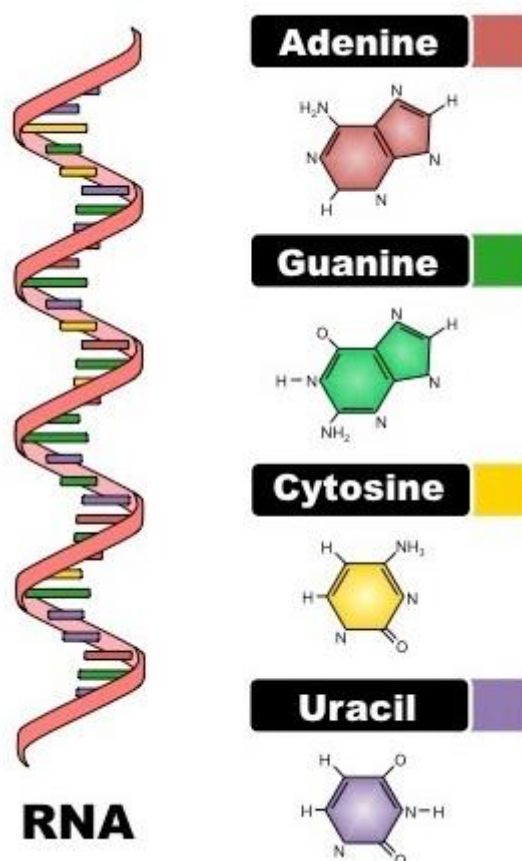


Fig.3. Ribonucleic Acid (RNA) [93]

Chromosomes and Genes:

A thread-like structure of DNA molecules is called chromosomes. These chromosomes are DNAs that are tightly coiled multiple times around a protein called Histones that support these DNA structures. These chromosomes are not visible under the microscope and have a point called the centromere, where they get divided into two arms. The short arm is the "p arm" while the long ones are the "q arms". You may look at Fig. 4. to understand the structure of a chromosome [16].

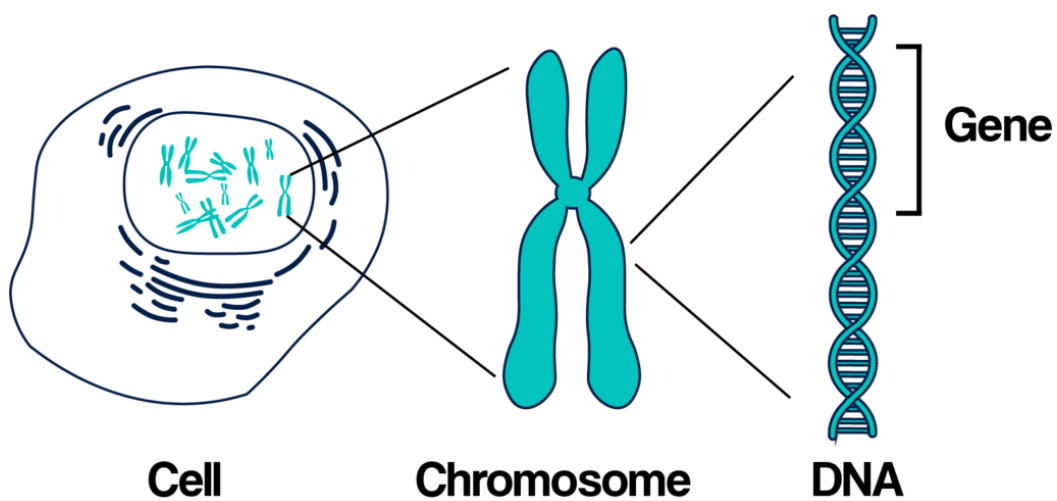


Fig.4. Chromosome & Gene [94]

Genes are functional units inherited from each parent. Humans have an estimated 20,000 to 25,00 genes. Each gene size could vary from a few hundred to more than 2 million DNA bases. Only 1% of these genes are barely different in all humans [17]. These human gene differences are responsible for a unique physical trait. Scientists have given them individual names to keep track of these genes; at times, genes could have long names. They are assigned a short combination of words. For example, a cystic fibrosis transmembrane conductance gene is known as CFTR [17]. You may have a look the Fig. 4. to look at a gene from a chromosome.

Gene Ontology (GO):

A body of knowledge within a given field is known as Ontology. Ontologies usually consist of classes (or terms or concepts) with relations. Gene Ontology provides information on different domains and can be freely used by the community. Gene Ontology (GO) describes our understanding of the biological domain concerning three aspects:

1. Biological Process.
2. Molecular Function.
3. Cellular Component.

Biological Process:

Multiple molecular activities, or 'biological programs', accomplish the more extensive processes. DNA repair or signal transduction are examples of broad biological process terms. More specific terms include pyrimidine nucleobase biosynthetic process or glucose transmembrane transport [18].

Molecular Function:

When gene products perform molecular activities such as transport or catalysis, the GO molecular functions are activities that are completed but don't specify the context of what, when and where it has occurred. A gene product usually performs these activities, but, at times, they could be achieved by molecular complexes with multiple gene products. Catalytic Activity and Transport Activity are some examples of broad functions; Adenylate cyclase activity or Toll-like receptor binding are examples of narrow functions [18].

Cellular Component:

The locations that are relative to cellular structures in which a gene product performs a function: cellular compartments (e.g., mitochondrion) or stable macromolecular complexes of which they are parts (e.g., the ribosome). Unlike the other aspects of GO, cellular component classes refer not to processes but rather to cellular anatomy [18].

GO (Gene Ontology) Dataset:

Source of the dataset:

<https://github.com/jhuaye/GenesRelateDiseases>

The Features dataset (go_features.csv) contains 18418 rows and 11953 columns, and the Labels dataset (class_labels.csv) contains 20659 rows and 29 columns. These features have unique GO identifiers. I had to download and convert the go.obb file, which includes identifiers having biological process, Molecular function & Cellular components with the description of each feature. We can only understand 11554 identifiers out of 11951 (i.e., 397 identifiers are not known yet to us). But it could be identified through the GO database website. Let's look at the imbalances in the dataset for each disease we intend to build the model in Table.1.

1 - Having a Disease, 0 - Not having a Disease

<i>Disease</i>	<i>1's</i>	<i>0's</i>	<i>% Of having disease</i>
<i>Heart</i>	283	17983	1.55%
<i>Diabetes – Type II</i>	214	18052	1.12%
<i>Alzheimer's</i>	115	18151	0.63%
<i>Parkinson's</i>	48	18218	0.26%
<i>Obesity</i>	127	18139	0.70%

Table 1. Imbalances in the datasets for each disease

Looking at the figures, we understand that the dataset is highly imbalanced. We only have 1.55%, 1.12%, 0.63, 0.26% and 0.7% examples of someone having Heart Disease, Type II Diabetes, Alzheimer's, Parkinson's & Obesity, respectively. We always aspire to train a model using a balanced target feature dataset. These imbalances would create an underfit model. However, We will use upsampling methods like ADASYN (Adaptive Synthetic) and SMOTE (Synthetic Minority Over-sampling Technique) to increase the samples of a minority class and train those models using those synthetic data points.

Heart Disease:

The article "Heart disease Is World's no. 1 Killer" by Robert Preidt mentions that the leading cause of death is Heart Disease. This disease accounted for one-third of deaths in 2019 alone [19]. An estimated 17.9 million lose their lives yearly due to Cardio Vascular disorders [20]. One-fourth of deaths in the UK are due to Heart and circulatory diseases. That's more than 160,000 deaths yearly, 460 deaths each day and one every three minutes. In the UK, roughly 7.6 Million (i.e. \$ Million Men & 3.6 Million Women) people live with this disease [21]. The report "Incidence of Cardiovascular Risk Factors in an Indian Urban Cohort", published in the National Library of Medicine, mentions that Annual deaths in India in 1990 were 2.26 Million, and it is estimated that total deaths would be 4.77 Million by the year 2020 [22]. It is estimated that 17.9 Million people have died due to some heart disease, representing 32% of the overall global deaths. Of these, 85% were due to heart attack and stroke [23]. Heart disease includes the following: Coronary Heart Disease, Heart Failure, Heart Arrhythmias, Heart Valve Disease(Endocarditis and Rheumatic Heart Disease), Pericardial Disease, Cardiomyopathy and Congenital Heart Disease [24]. The factors contributing to these heart conditions are Age, Sedentary Lifestyle, Diabetes and Metabolic Syndrome, Genetics,

High Blood Pressure, Bad Cholesterol levels, Obesity, Smoking & Stress [24].



Fig.4. Implantable Cardioverter Defibrillators (ICD) [95]

Cause of Inherited Cardiac Conditions (ICC) by the mutations of one or more of our genes. A 50% chance of it being passed to your children if you have a faulty gene. Someone could be unaware of such conditions as they don't constantly develop symptoms [25]. In contrast, some develop symptoms like dizzy spells, palpitations, blackouts & shortness of breath. Some treatments for this condition include a healthy lifestyle, medications and implantable cardioverter defibrillators. You may look at the implantable cardioverter defibrillators (ICD) in Fig.4.

We'll be merging the GO dataset with the "class_Disease.Heart" feature on the primary key "entrezId". We'll get 18418 rows and 11954 columns dataset. The target feature "class_Disease.Heart" has 17923 data points as 0, 283 data points as 1 and 152 data points as "?". We'll drop the "?" data points. And now, we are left with 18266 rows and 11954 columns heart dataset. Having 283 features having heart disease and 17923 features not having heart disease is a big concern. We have only 1.55% of examples of someone having heart disease. We'll perform up-sampling methods like ADASYN and SMOTE to increase the size of the dataset. We'll use three datasets versions and evaluate

the model's performance. ADASYN and SMOTE algorithms improve the balance by having 12619 examples and 12585 respectively of someone having a heart condition. To understand the performances of each model, we can look at the Table. 2.

DATA	MODEL	ACCURACY	PRECISION	RECALL	AUCROC
GENERIC	<i>Logistic Regression</i>	0.985584	0.347107	0.25609756	0.626381
	<i>Random Forest</i>	0.985401	0.625	0.06097561	0.536307
	<i>Neural Network</i>	0.969708	0.088235	0.1097561	0.546264
ADASYN	<i>Logistic Regression</i>	0.933759	0.062305	0.24390244	0.594071
	<i>Random Forest</i>	0.969891	0.053763	0.06097561	0.522337
	<i>Neural Network</i>	0.970985	0.076923	0.08536585	0.534902
SMOTE	<i>Logistic Regression</i>	0.934307	0.060127	0.23170732	0.588343
	<i>Random Forest</i>	0.969891	0.053763	0.06097561	0.522337
	<i>Neural Network</i>	0.956204	0.029762	0.06097561	0.51539

Table 2. Model Performance to predict Heart Disease

Type II Diabetes:

The cases of Diabetes rose from 108 million to 422 million from 1980 to 2014. The rapid rise in cases is happening worldwide in low- and middle-income countries. Diabetes causes kidney failure, heart attack, lower limb amputation, Stroke and blindness. Diabetes was the 9th leading cause of death in 2019, with an estimated 1.5 Million deaths caused due to it [26]. More than 95% of the diabetes cases have type 2 Diabetes. Approximately 462 million individuals have Type 2 Diabetes, roughly 6.28% of the world's population. 4.4% of people aged between 15 to 49 years, 15% between 50 to 69 years, and 22% of people above 70 [27]. In 2020, Due to lockdowns in the UK, Type 2 Diabetes diagnoses were down by 70% compared to an average throughout the last decade. Forty-five thousand diagnoses were missed or delayed between March 2020 to July 2020. The researchers at the University of Manchester found that England's death rate for individuals having

type 2 Diabetes was twice as high (110%) in April 2020. In Scotland, Wales, and Northern Ireland, death rates were 66% higher than expected [28]. In 2016, 1 out of 11 Indians was under formal diagnosis, which makes it the most affected country with Diabetes in the world after China [29]. According to International Diabetes Federation, their model forecasts that by 2045 India will have 134 Million cases of Diabetes [30]. In Diabetes, The pancreas is inefficient in producing insulin to transfer glucose from the bloodstream to adipose tissues as fat. High glucose in the bloodstream is known as "Hyperglycaemia"; this condition, over time, can cause severe damage to the body's system. The causes are excess body weight and having a sedentary lifestyle [26]. Also, according to "American Diabetes Association", race and genetic factors are essential in causing type 2 diabetes [31]. Symptoms include excessive excretion of urine (polyuria), thirst (polydipsia), vision change, weight loss and fatigue [26].

To observe the sugar levels in the blood, we use a Glucometer; we can look at normal sugar levels Fig.5.

BLOOD SUGAR CHART	
FASTING	
Normal for person without diabetes	70-99 mg/dl (3.9-5.5 mmol/L)
Official ADA recommendation for someone with diabetes	80-130 mg/dl (4.4-7.2 mmol/L)
2 HOURS AFTER MEALS	
Normal for person without diabetes	Less than 140 mg/dl (7.8 mmol/L)
Official ADA recommendation for someone with diabetes	Less than 180 mg/dl (10.0 mmol/L)
HBA1C	
Normal for person without diabetes	Less than 5.7 %
Official ADA recommendation for someone with diabetes	7.0 % or less

Fig.5. Normal Sugar Levels [96]

We'll merge the GO dataset with the "class_Nutritional.Diabetes.Type2" feature on the primary key "entrezId". We'll get 18418 rows and 11954 columns dataset. The target feature "class_Nutritional.Diabetes.Type2" has 18052 data points as 0, 214 data points as 1 and 152 as "?". We'll drop the "?" data points. And now, we are left with 18266 rows and 11954 columns Diabetes -Type 2 dataset. Having 214 features having Diabetes -Type 2 disease and 18052 features not having Diabetes -Type 2 disease is a big concern. We have only 1.12% of examples of someone having Diabetes -Type 2 disease. We'll perform up-sampling methods like ADASYN and SMOTE to increase the size of the dataset. We'll use three datasets versions and evaluate the model's performance. ADASYN and SMOTE algorithms improve the balance by having 12683 examples and 12634, respectively, of someone with a Diabetes -Type 2 condition.

To understand the performances of each model, we can look at the Table. 3.

DATA	MODEL	ACCURACY	PRECISION	RECALL	AUCROC
<i>GENERIC</i>	<i>Logistic Regression</i>	0.987409	0.347826	0.12903226	0.563132
	<i>Random Forest</i>	0.989234	0.8	0.06451613	0.532166
	<i>Neural Network</i>	0.963139	0.013889	0.03225806	0.532166
<i>ADASYN</i>	<i>Logistic Regression</i>	0.936679	0.029703	0.14516129	0.545449
	<i>Random Forest</i>	0.97427	0.024096	0.03225806	0.508654
	<i>Neural Network</i>	0.975	0.048193	0.06451613	0.508654
<i>SMOTE</i>	<i>Logistic Regression</i>	0.937044	0.0299	0.14516129	0.545633
	<i>Random Forest</i>	0.975	0.025316	0.03225806	0.509023
	<i>Neural Network</i>	0.974635	0.057471	0.08064516	0.509023

Table 3. Model Performance to predict Type 2 Diabetes.

Alzheimer's:

Alzheimer's is a neurological disorder that gradually demolishes memory, thinking ability, and the most straightforward task [32]. The damage initially occurs in the entorhinal cortex and hippocampus, which is responsible for memory; gradually, it damages the cerebral cortex, which is responsible for language, social behaviour and reasoning and eventually, other areas within the brain are damaged. To understand the areas of the brain, refer to Fig. 6.

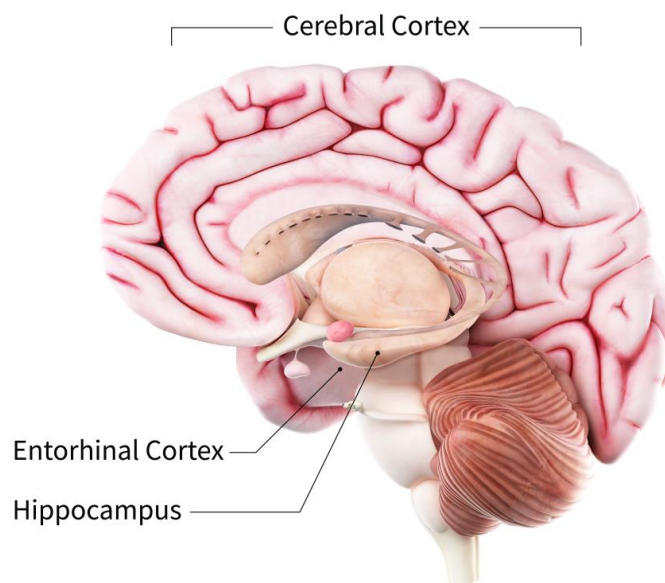


Fig.6. Areas of brain affected by Alzheimer's [97]

For individuals above 80, death varies from 3 Years to 10 or more years if the individual is young. Currently, there is no cure for the disorder [32]. As of 2020, there are over 50 Million individuals with Dementia worldwide. There are 8.92 Million and 4.31 Million cases of Dementia in Western Europe and Southeast Asia, respectively [33]. Alzheimer's is a specific brain disorder, while Dementia is a term used in general to represent a decline in mental ability. In 2020 there were 58.66 Million cases of Dementia, out of which 2.85%, 23.97%, 34.35%, and 38.83% belonged to Low, Lower-Middle, Upper-Middle and High-income groups, respectively.

It is forecasted that by 2040 there will be 114.83 Million cases, roughly twice the cases in 2020. The estimated worldwide cost of Dementia is USD (\$) 1.3 Trillion, which is forecast to rise to USD (\$) 2.8 Trillion by 2030 [33]. Those with late-onset symptoms of Alzheimer's will first appear in the mid-'60s, and early-onset symptoms will begin between the 30s and mid-'60s. Individuals with mild symptoms of Alzheimer's might seem healthy, but they can be troublemakers. Mild Alzheimers symptoms include memory loss, poor judgement (which leads to a wrong decision), trouble handling money, misplacing belongings, personality and mood change, increase in anxiety and/or aggression, repeating questions, and loss of spontaneity and initiative [34]. Researchers have not found a specific gene directly responsible for late-onset Alzheimer's disorder. However, a genetic variant of the apolipoprotein E(APOE) gene on chromosome 19 does increase the risk of individuals. This protein helps in carrying cholesterol and fats into the bloodstream [35].

We'll merge the GO dataset with the "class_Brain.Alzheimer" feature on the primary key "entrezId". We'll get 18418 rows and 11954 columns dataset. The target feature "class_Brain.Alzheimer" has 18151 data points as 0, 115 data points as 1 and 152 as "?". We'll drop the "?" data points. And now, we are left with 18266 rows and 11954 columns Alzheimer's dataset. Having 214 features having Alzheimer's disease and 18052 features not having Alzheimer's disease is a big concern. We have only 1.12% of examples of someone having Alzheimer's disease. We'll perform up-sampling methods like ADASYN and SMOTE to increase the size of the dataset. We'll use three datasets versions and evaluate the model's performance. ADASYN and SMOTE algorithms improve the balance by having 12723 examples and 12717, respectively, of someone with Alzheimer's.

To understand the performances of each model, we can look at the Table. 4.

DATA	MODEL	ACCURACY	PRECISION	RECALL	AUCROC
GENERIC	<i>Logistic Regression</i>	0.991058	0.285714	0.04347826	0.521279
	<i>Random Forest</i>	0.991423	0	0	0.510778
	<i>Neural Network</i>	0.979562	0.028571	0.04347826	0.510778
ADASYN	<i>Logistic Regression</i>	0.961861	0.017751	0.06521739	0.517334
	<i>Random Forest</i>	0.987409	0	0	0.497884
	<i>Neural Network</i>	0.973175	0.028037	0.06521739	0.498068
SMOTE	<i>Logistic Regression</i>	0.962409	0.018072	0.06521739	0.517611
	<i>Random Forest</i>	0.987956	0	0	0.49816
	<i>Neural Network</i>	0.97646	0	0	0.49816

Table 4. Model Performance to predict Alzheimer's.

Parkinson's:

The number of individuals with Parkinson's doubled to over 6 Million between 1990 and 2015 [38]. More than 10 Million individuals worldwide live with Parkinson's Disorder [37]. According to a report on Parkinson's by World Health Organisation (W.H.O.) Global estimates in 2019 showed around 8.5 million individuals with Parkinson's Disease. Current estimates suggest that, in 2019, Parkinson's Disease resulted in 5.8 million disability-adjusted life years, an increase of 81% since 2000, and caused 329 000 deaths, an increase of over 100% since 2000 [36]. Estimates show that In 2020 there were around 145,000 individuals diagnosed with Parkinson's. 121,000 cases in England, 12,400 in Scotland, 7,600 in Wales and 3,900 in Northern Ireland. 1 in 37 individuals in the UK will be diagnosed with Parkinson's in their lifetime. Every hour 2 people are getting diagnosed in the UK. It is forecast that by 2030, 172,000 individuals will have Parkinson's [38]. In India, there are 7 million individuals who are diagnosed with Parkinson's [39]. And according to health experts in

India, India could see a whopping 200-300% increase in cases over the two to three decades [40].

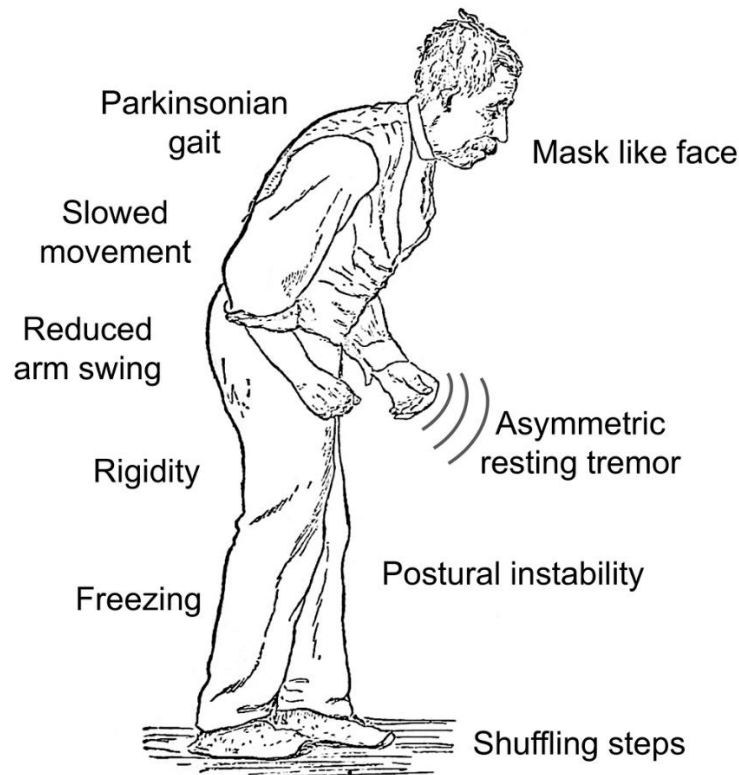


Fig.7. Some signs of having Parkinson's [98]

Parkinson's is a degenerative condition of areas of the brain that causes slow movements, tremors, rigidity, and imbalance of the body and includes other complications like sleep disorders and sensory disturbances [36]. Men are more impacted by Parkinson's (2 men out of 3 individuals). Risk factors include repeated head injury, improper diet, significant exposure to pesticides or solvents and a sedentary lifestyle. Also, certain variations in the gene appear to increase the risk of this disease slightly [39]. A wide range of symptoms related to Parkinson's include tremors, muscle stiffness, lousy posture, trouble with coordination, change in speech and writing, difficulty in chewing and swallowing and problems with urination. We also have unmotorized symptoms that include anxiety, depression, sleep-related issues, fatigue, weight loss & constipation [39]. To understand these symptoms visually, you may have a look at Fig.7.

We'll merge the GO dataset with the "class_Brain.Parkinson" feature on the primary key "entrezId". We'll get 18418 rows and 11954 columns dataset. The target feature "class_Brain.Parkinson" has 18218 data points as 0, 48 data points as 1 and 152 as "?". We'll drop the "?" data points. And now, we are left with 18266 rows and 11954 columns Parkinson's dataset. Having 48 features having Parkinson's disease and 18218 features not having Parkinson's disease is a big concern. We have only 0.26% of examples of someone having Parkinson's disease. We'll perform up-sampling methods like ADASYN and SMOTE to increase the size of the dataset. We'll use three datasets versions and evaluate the model's performance. ADASYN and SMOTE algorithms improve the balance by having 12723 examples and 12717, respectively, of someone with Parkinson's.

To understand the performances of each model, we can look at the Table. 5.

DATA	MODEL	ACCURACY	PRECISION	RECALL	AUCROC
<i>GENERIC</i>	<i>Logistic Regression</i>	0.996715	1	0.05263158	0.526316
	<i>Random Forest</i>	0.996533	N/A	0	0.5
	<i>Neural Network</i>	0.992336	0.04	0.05263158	0.5
<i>ADASYN</i>	<i>Logistic Regression</i>	0.97281	0.007576	0.05263158	0.514322
	<i>Random Forest</i>	0.984672	0	0	0.494049
	<i>Neural Network</i>	0.990146	0	0	0.494049
<i>SMOTE</i>	<i>Logistic Regression</i>	0.972628	0.007519	0.05263158	0.51423
	<i>Random Forest</i>	0.984489	0	0	0.493957
	<i>Neural Network</i>	0.991423	0	0	0.493957

Table 5. Model Performance to predict Parkinson's.

Obesity:

According to World Health Organisation (W.H.O), more than 1.9 billion adults (over 18 Years) were overweight worldwide in 2016. Out of these, 650 million were Obese. Around 340 Million children and adolescents (between 5 to 19 Years) were Overweight or Obese in 2016. About 39 Million children (under five years) are Overweight or Obese in 2020 [41]. As of November 2020, 62.8% of adults were Overweight or Obese. Black adults were the highest, with 67.5% of adults being overweight or Obese.

In comparison, the Chinese adults were the lowest, with 32.2% of adults being Overweight or Obese [42]. In 2016, Overweight and Obesity in White British people went up from 62% to 63.7%, whereas in Black people, it went down from 72.8% to 67.5% [42]. With 27.8% of the population being obese, the UK stands at 33rd Rank in the percentage of the population being Obese [43]. In India, Obesity is very high across all the zones within the country. Obesity was higher among women than men. Higher in Urban areas compared to Rural and Higher in Educated to Uneducated [44]. Although India ranks 3rd after China in the number of individuals being obese [45], With 3.9% of the population being obese, India stands at 187th Rank in the percentage of the population being Obese [43]. Obesity or being Overweight is an abnormal or excessive collection of fat that may interfere with our health [41]. Body Mass Index (BMI) is used to classify someone as obese of both sexes [46]. The formula to calculate the BMI of an adult is given below.

$$\text{Body Mass Index (BMI)} = \text{kg/m}^2$$

To understand your positioning of the index, you may look at Fig. 8. Although BMI is an excellent metric to understand if someone is obese but, It doesn't account for body composition [48]. If we take bodybuilders as an example, they have high muscles and low body fat (i.e. although the person has low body fat still, the BMI might give an

index that could be interpreted as being Overweight or Obese) [48]. Some contributing factors are excess intake of energy-dense food high in sugar and fats, physical inactivity, urbanisation and genetics [47]. Obesity could cause life-threatening diseases, Type 2 Diabetes, Coronary Heart Disease, and some cancers like breast cancer or bowel cancer and stroke. Also, it affects the quality of life and can cause psychological issues like depression and lower self-esteem [46].

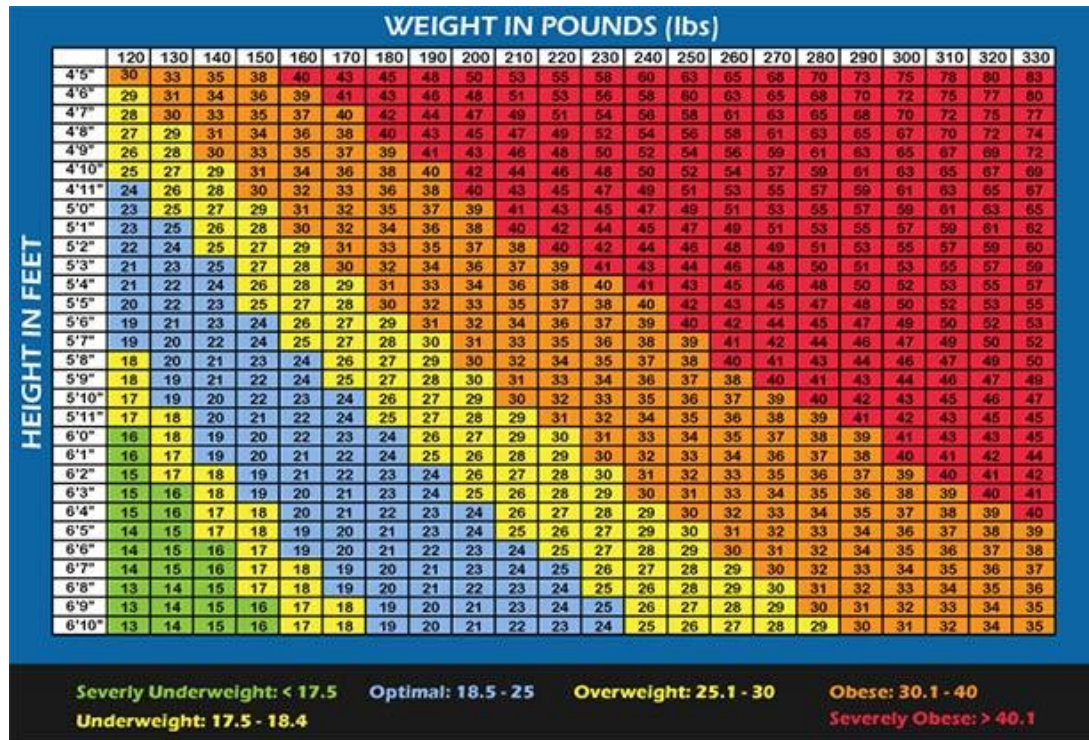


Fig.8. Body Mass Index (BMI) Chart [99]

We'll merge the GO dataset with the "class_Nutritional.Obesity" feature on the primary key "entrezId". We'll get 18418 rows and 11954 columns dataset. The target feature "class_Nutritional.Obesity" has 18139 data points as 0, 127 data points as 1 and 152 as "?". We'll drop the "?" data points. And now, we are left with 18266 rows and 11954 columns Obesity dataset. Having 127 features having Obesity and 18218 features not having Obesity is a big concern. We have only 0.26% of examples of someone's Obesity. We'll perform up-sampling methods like ADASYN and SMOTE to increase the size of the dataset. We'll use three datasets versions and evaluate the model's

performance. ADASYN and SMOTE algorithms improve the balance by having 12723 examples and 12717, respectively, of someone with Obesity.

To understand the performances of each model, we can look at the Table. 6.

DATA	MODEL	ACCURACY	PRECISION	RECALL	AUCROC
GENERIC	<i>Logistic Regression</i>	0.993066	0.5	0.15789474	0.578396
	<i>Random Forest</i>	0.993248	1	0.02631579	0.513158
	<i>Neural Network</i>	0.985766	0.02381	0.02631579	0.509391
ADASYN	<i>Logistic Regression</i>	0.962226	0.043243	0.21052632	0.589001
	<i>Random Forest</i>	0.980109	0	0	0.493477
	<i>Neural Network</i>	0.992336	0	0	0.499632
SMOTE	<i>Logistic Regression</i>	0.962774	0.043956	0.21052632	0.589276
	<i>Random Forest</i>	0.979745	0	0	0.493293
	<i>Neural Network</i>	0.987409	0.121951	0.13157895	0.562482

Table 6. Model Performance to predict Obesity.

Logistic Regression:

Logistic Regression is a statistical model used for classification. We calculate the probability of an occurring event. In the industry, we have certain problem statements with a classification task [49]. For example, Is the individual like to have a specific disease, Does the user have a good credit score, Is the transactional fraudulent and more? The value ranges from 0 to 1. If the value is "1", the condition is true, or the condition is false if "0" [49]. There are different types of Logistic Regression models based on categorical responses:

Binary Classification:

It is one of the most commonly used approaches. This approach results in responses or target feature, which will have two possible outcomes "1" and "0". For example, it is classifying whether a mail is a scam or not [50].

Multiclass Classification:

In this approach, we have more than two possible outcomes. We have only one possible response or target feature from these outcomes. For example, to classify the genre of music or film [50].

Ordinal Classification:

This logistic regression type is used when we have three or more possible outcomes. But they have a defined order. For example, a rating scale from 1 to 5 for a restaurant [50].

First, model a linear equation with 11954 weights (or coefficients) and a constant. We can denote this linear regression model like this:

$$\hat{y} = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \dots + \beta_{11954} * x_{11954}$$

Where, \hat{y} - Output from Linear Regression

β_0 - Constant

β_n - Weights of the feature

n - Number of features

The problem with this model is that it gives a value between the interval $(-\infty, +\infty)$. Whereas probability has values $[0,1]$. We need to transform using the sigmoid function, which gives values $[0,1]$. Sigmoid function has a characteristic S-shaped curve. It is also referred to as a Logistic function [52].

$$p = \frac{1}{1 + e^{-\hat{y}}}$$

Where, p - Probability value between 0 to 1.

\hat{y} - Output from Linear Regression.

e - Euler's number.

Later we can choose a threshold greater than 0 and less than 1. If the values are more significant than the threshold, we'll predict them as "1" or "0". To build a logistic regression model, we'll be using Sci-kit learn library in Python.

Advantages [51]:

1. It is easier to implement, train and efficient at training.
2. It can easily classify multiple classes.
3. Good performance when the dataset is linearly separable.
4. Coefficients of the features are indicators of feature importance.
5. It is less inclined to over-fitting in low dimensions.

Disadvantages [51]:

1. It cannot construct non-linear boundaries.
2. It assumes that there is a linear relationship between dependent features.
3. Requires average or no multicollinearity between independent variables.
4. Challenging to obtain complex relationships between dependent variables.
5. It Overfits if the observations are less than the number of features.

Random Forest:

We need to know how the Decision Tree works to briefly understand the Random Forest algorithm. To understand, let's make a decision tree model, to predict whether someone is likely to buy a vehicle in Fig. 9.

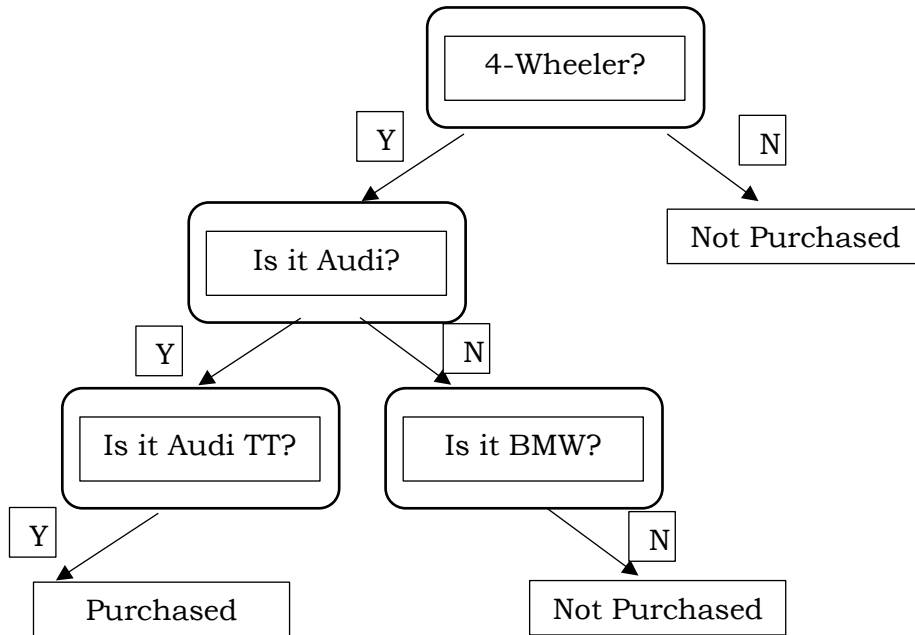


Fig.9. Decision Tree

It's a binary classification model in the first node; there is a rule to understand whether the vehicle is a 4-wheeler. If this condition is satisfied, we move to the other node, or the car will not be purchased. Since we moved to the next node, we ask if the make is Audi. If it's an Audi, we move towards the next node or ask if the car is a BMW. If it's not a BMW, we do not purchase the vehicle. Since it's an Audi, we move toward the next node and ask if it's a TT model. In case it is a TT model, we purchase the car or else do not purchase the vehicle. So we can observe how specific rules are associated with the decision to buy a car. This approach is known as a Decision Tree algorithm. This approach is suitable for building models having non-linear features, but, It is a lazy learner as it tends to overfit.

Random Forest is one of the most potent and popular machine learning algorithms, which is an Ensemble ML algorithm also called Bootstrap Aggregation or Bagging. Ensemble algorithms combine two or more ML models [53]. Similarly, Random Forest is a combination of multiple decision tree models. As the name suggests, we first perform bootstrapping, a technique where we sample data from the original dataset with replacements and build a model using these sampled data [58]. We achieve this multiple times and build different decision tree models using different sampled datasets. Later we combine the results from each model and find the majority. If we build 20 Decision Trees in a Random Forest and 15 of those are predicted as someone likely to buy a car, we take "1" as the final predicted value [53].

Now, Let us understand how a Decision Tree chooses the rule of a node in a Random Forest. We have sampled data to build the tree.

There are a variety of decision trees algorithms, namely:

1. ID3 (Iterative Dichotomiser 3): It is an extension of the D3 algorithm. Ross Quinlan developed it in 1986. A greedy algorithm is an instant approach that seems best at that moment. The algorithm uses the top-down greedy search through all the branches without backtracking [55]. The features that will yield the highest information gain for the target feature with less entropy. They are grown to the maximum size, and later pruning steps are usually applied to improve the ability of the tree to generalize. This model is only used for classification problems, and features must be categorical [54].
2. C4.5 (Successor of ID3): This model removes the restriction of having all features as categorical. C4.5 converts the output of the ID3 algorithm into a set of if-then rules. The order is then determined by calculating the accuracy of each rule. Then, pruning is performed by removing the precondition rules if the accuracy improves without those rules [54].

3. C5.0: It is a proprietary licensed algorithm that Ross Quinlan developed. It uses fewer rules compared to C4.5 simultaneously, being accurate and using less memory [54].
4. CART (Classification and Regression Trees): The algorithm builds binary trees using features and their threshold to generate significant information gain at each node. Unlike C4.5, It also supports numerical target feature (Regression) and doesn't need to compute rules [54].

Scikit learn package claims to use an optimized version of the CART(Classification and Regression Tree) algorithm [54].

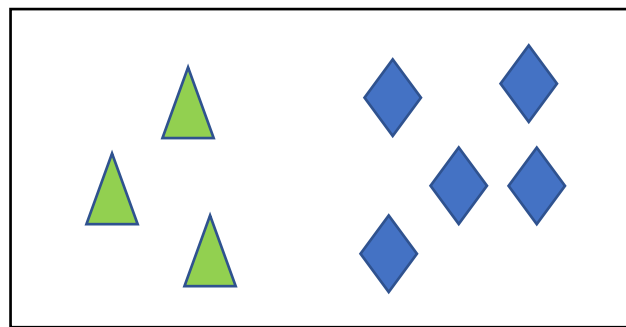


Fig 10. Terminal Node have 3 green triangle and 5 blue diamonds.

Quantifying Homogeneity:

While building a decision, we want the terminal node to be homogeneous as possible (a clear majority) [53]. To quantify these homogeneities, we use different mathematical measures like:

1. Gini Index.
2. Entropy.
3. Deviance.

To understand how these are calculated, let us have a look at Fig. 8 We have three green triangles and five blue diamonds at the terminal node.

Gini Index:

$$\textbf{Gini Index} = 1 - \sum_{i=1}^k p_i^2$$

Where, k – number of classes

p – proportion of a class

Looking at the Fig.8, the Gini Index can be computed as follows:

$$\textbf{Gini Index} = 1 - \left[\left(\frac{3}{8} \right)^2 + \left(\frac{5}{8} \right)^2 \right] = 0.46875$$

Entropy:

$$\textbf{Entropy} = - \sum_{i=1}^k p_i * \log(p_i)$$

Where, k – number of classes

p – proportion of a class

Looking at the Fig.8, the Entropy can be computed as follows:

$$\textbf{Entropy} = - \left[\left(\frac{3}{8} \right) * \log \left(\frac{3}{8} \right) + \left(\frac{5}{8} \right) * \log \left(\frac{5}{8} \right) \right] = 0.6616$$

Deviance:

$$\textbf{Deviance} = - \sum_{i=1}^k n_i * \log(p_i)$$

Where, k – number of classes

p – proportion of a class

n – no. of observations in a class.

Looking at the Fig.8, the Deviance can be computed as follows:

$$\textbf{Deviance} = - \left[3 * \log \left(\frac{3}{8} \right) + 5 * \log \left(\frac{5}{8} \right) \right] = 5.2925$$

Information Gain:

It measures the expected reduction in entropy. The feature which has minimum impurity will be considered as the root node. Generally, It is used to decide which feature to be used to split on at each step in building the tree. Creating sub-nodes increases the homogeneity that decreases the entropy of these nodes. The more the child node is homogeneous, the more the variance is reduced after each split. Thus Information Gain is the variance reduction and can calculate how much the variance decreases after each division [57].

Observations	Car	Outcome
1	Audi	Yes
2	BMW	Yes
3	Audi	No
4	BMW	Yes
5	Audi	No
6	BMW	Yes
7	BMW	Yes
8	Audi	Yes
9	BMW	No
10	Audi	No

Table 7. Example Dataset

Let's take a small example to calculate Information Gain. We have a dataset [Table 6] with ten observations belonging to two classes, Yes and No. 5 observations belonging to class Yes and five observations belonging to class No.

Audi has 2 Yes and 3 No outcomes, whereas BMW has 3 Yes outcomes and 2 No outcomes.

$$E(\text{Parent Node}) = - \left[\left(\frac{6}{10} \right) * \log \left(\frac{6}{10} \right) + \left(\frac{4}{10} \right) * \log \left(\frac{4}{10} \right) \right] = 0.6730$$

$$E(\text{Audi}) = - \left[\left(\frac{2}{5} \right) * \log \left(\frac{2}{5} \right) + \left(\frac{3}{5} \right) * \log \left(\frac{3}{5} \right) \right] = 0.6730$$

$$E(BMW) = - \left[\left(\frac{3}{5} \right) * \log \left(\frac{3}{5} \right) + \left(\frac{2}{5} \right) * \log \left(\frac{2}{5} \right) \right] = 0.6730$$

$$\begin{aligned} \text{Weighted Average} &= \frac{5}{10} * E(\text{Audi}) + \frac{5}{10} * E(\text{BMW}) \\ &= \frac{5}{10} * 0.6730 + \frac{5}{10} * 0.6730 \\ &= 0.673 \end{aligned}$$

$$\begin{aligned} \text{Information Gain} &= E(\text{Parent Node}) - \text{Weighted Average} \\ &= 0.6730 - 0.6730 = 0 \end{aligned}$$

It means there has been no information gained in this case.

Note: We have used Log to the base e.

Advantages [59]:

1. Reduces overfitting by using the ensembling learning technique.
2. It can solve Classification and Regression targets.
3. Good performance even if the dataset is non-linearly separable.
4. Robust to outliers in the dataset.
5. Performs better as compared to the Logistic Regression model.

Disadvantages [59]:

1. Since we are creating many trees, it requires high power and computational Resources.
2. Longer time to train as compared to Logistic Regression and Decision Tree model.

Neural Network:

Deep Learning is a type of Machine Learning Algorithm inspired by neurons in a human's brain. In 1943, Walter Pitts & Warren McCulloch built the first Neural Network-inspired computational model. It was a combination of algorithms and maths. They called it "Threshold Logic" as it imitates the thought process. Deep Learning is usually used for object detection and recognition and Natural Language Processing [61].

One of the techniques in Deep Learning is Artificial Neural Network (ANN). An ANN comprises perceptron layers containing an input layer, one or more hidden layers, and output layers. Each perceptron within these layers is connected and has its associated weight and threshold [61]. To visually understand the structure, look at Fig.11.

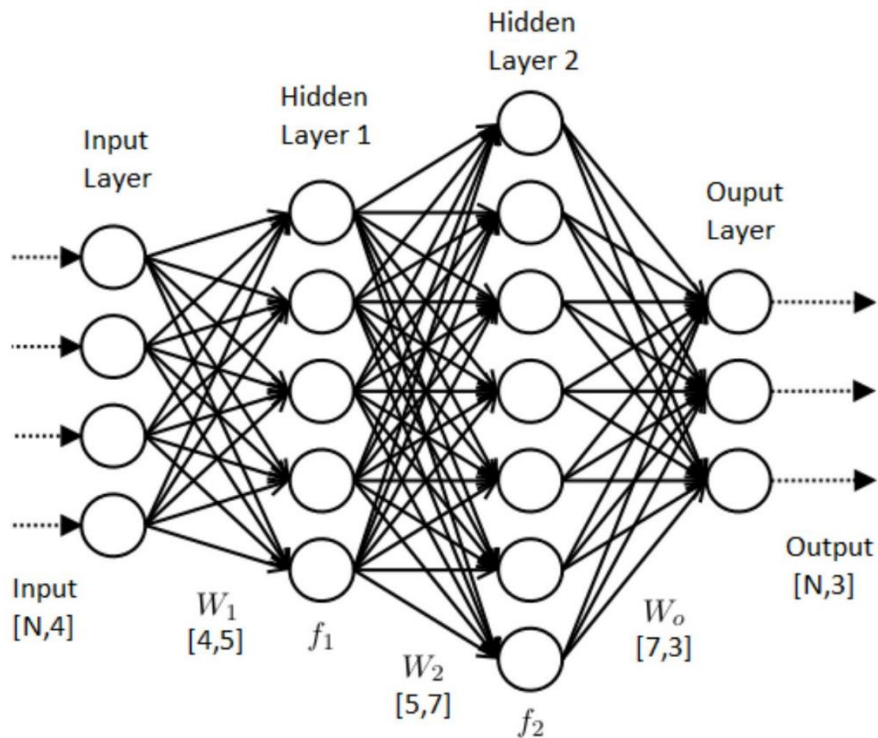


Fig.11. Artificial Neural Network (ANN) with two hidden layers [100]

Each perceptron in these layers acts as a linear regression model. They have weights, bias(or threshold) and output [61]. To understand this, let's have a look at the equation below.

We perform a summation on the product weights and inputs and add bias. Later we use the activation function on this result to give a single output [62].

$$Output = activation((\sum_{i=0}^m w_i * x_i) + bias)$$

Activation function:

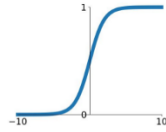
It is one of the critical components of Deep Learning. The output of the activation function in the perceptron will be shared with the next perceptron [62].

Types of Activation Functions can be seen in Fig. 12.

Activation Functions

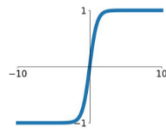
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



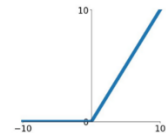
tanh

$$\tanh(x)$$



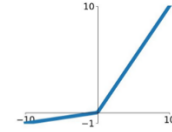
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$



Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

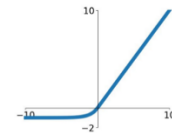


Fig.12. Activation Functions [101]

We have built the model and would like to evaluate its performance. For regression, we use Mean Absolute Error (MAE), Mean Square Error (MSE) or Root Mean Square Error (RMSE) [63].

Evaluating Regression & Classification model:

We can look at the formula below:

Mean Absolute Error:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y} - y_i|$$

Where, \hat{y} - Predicted Value

n - Total number of values

Mean Squared Error:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y} - y_i)^2$$

Where, \hat{y} - Predicted Value

n - Total number of values

Root Mean Squared Error:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y} - y_i)^2}$$

Where, \hat{y} - Predicted Value

n - Total number of values

These error metrics are usually used for regression problems [61], but for classification, we use Accuracy, Precision, Recall, F1 Score and Area under the curve of Region of Characteristics (AUCROC)[64].

To calculate these metrics, we need to create a confusion matrix.

Confusion Matrix is a table that evaluates the performance of a classification model. The rows of this table have predicted values, and the columns have actual values. To understand visually, we may look at Fig.12.

Confusion Matrix:

<i>Predicted Values</i>	<i>Actual Values</i>	
	Positive (1)	Negative (0)
	Positive (1)	Negative (0)
	TP	FP
	FN	TN

True Positive (TP): Predicted 1, Real 1

True Negative (TN): Predicted 0, Real 0

False Positive (FP): Predicted 1, Real 0

False Negative (FN): Predicted 0, Real 1

Using these, we can calculate the metrics. Formulas for these metrics are given below.

Accuracy:

It measures how often the classifier correctly predicts; the ratio of correct predictions (True Positive and True Negative) and the total number of predictions is defined as accuracy [64].

Even though accuracy is a good metric to evaluate performance, It could be misleading. Having a model with 99% doesn't necessarily mean it is a good model. There is a situation where a model misclassifies the target. For example, We have a binary classification model to predict cancer, and we have a test dataset with 100 targets, out of which only one observation has cancer. If the model tends to classify everything as zero, we get 99% accuracy which is very misleading and could be destructive. To overcome this limitation, we also use other metrics like Precision, Recall, F1 Score and AUC-ROC [64].

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Precision:

It explains how many positive labelled targets are predicted as positive by the classifier model. The precision is close to 1 if we have high True Positives, low False Positive, and close to 0 if the other way around [64].

$$Precision = \frac{TP}{(TP + FP)}$$

Recall:

It explains how many positive labelled targets the model predicted correctly if the recall is close to 1 if we have high True Positives, low False Negatives, and close to 0 if the other way around. It is also known as Sensitivity [64].

$$Recall = \frac{TP}{TP + FN}$$

F1 Score:

It gives us a combined metric for Precision and Recall. We get the maximum value if the precision is equal to recall. It is the harmonic mean of precision and recall by punishing extreme values [64].

This is a practical evaluation metric for the following situation:

1. When the False Positives and False Negatives are the same.
2. High True Negative.
3. The outcome doesn't change with the addition of more data.

$$F1\ Score = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}$$

AUC-ROC (Area under the curve of region of characteristics):

It is a plot with True Positive Rate as the y-axis and False Positive Rate as the x-axis. Which explains how perfectly the classifier can generalise. When the AUC is near 1, we say the model performs better in generalising between positive and negative classes. If the AUC score is 0.5, we say that the classifier cannot distinguish between positive and negative classes [64].

Binary Cross Entropy:

It is a combination of Sigmoid activation and a Cross-Entropy loss as shown in Fig.13. It is independent for each vector component (class), unlike Softmax loss, which means that the loss calculated for each CNN output vector component is not affected by other component values. This is why it is used for multi-label classification, where the insight of an element belonging to a particular class should not influence the decision for another class. It is also called Sigmoid Cross-Entropy loss [66].

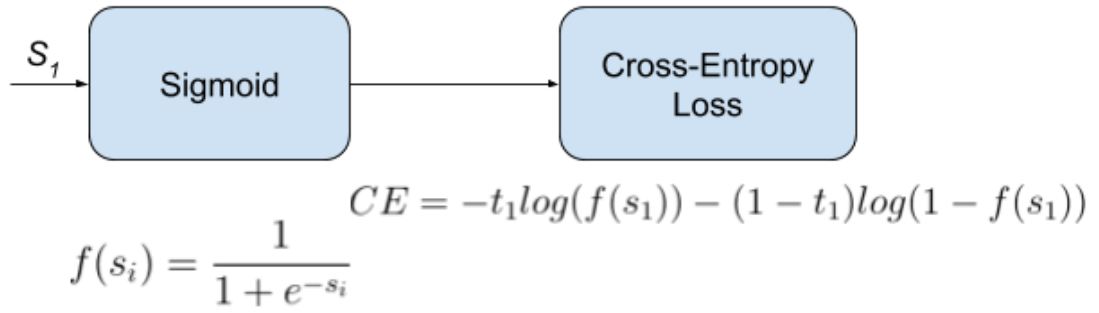


Fig.13: Binary Cross Entropy [66]

To calculate the loss for our classification model, we have used BCE Loss as the criterion to measure the loss between the input probabilities and the target. Formula for Binary Cross-Entropy Loss:

$$CE = - \sum_{i=1}^{C'=2} t_i \log(f(s_i)) = -t_1 \log(f(s_1)) - (1 - t_1) \log(1 - f(s_1))$$

Optimisation:

While making an ML model, we aim to have low loss and high accuracy. We spoke about three metrics we use to evaluate the regression model. One of these metrics is used to create a Cost Function [65].

We aim to find the value of weights with low loss or high accuracy. We use optimisation algorithms to find the lowest loss point in a cost function. Optimisation becomes significantly more accessible when the gradient of the cost function can be calculated [65].

Gradients are simply the derivative of a multivariate continuous cost function. The derivative of each variable in the vector is called the partial derivative of that point, assuming all other variables are constant [65].

Groups of algorithms that use gradients are as follows:

1. Bracketing Algorithm.
2. Local Descent Algorithm.
3. First-Order Algorithms.
4. Second-Order

Bracketing Optimisation Algorithms:

These methods are used with one variable where the optimal value (Maxima or Minima) is known within a specified range. The algorithm assumes that there is only one optimal value and can navigate effectively through known ranges to locate the optimal value [65].

The Bracketing Optimisation Algorithms include:

1. Fibonacci Search.
2. Golden Section Search.
3. Bisection Search.

The Local Descent Algorithm:

The algorithm uses one or more variables with a single optimal value. A direction is chosen, and a bracketing type search is performed in a hyperplane or line.

The variations of Line Search algorithms are used to locate the optimal value [65].

First-Order Algorithms:

It involves the first derivative, which is used to choose the direction to navigate in the search space. Later, using the gradient, we move in the opposite direction using the learning rate. A learning rate controls how far to move in a search space [65].

The algorithm includes:

1. Gradient Descent.
2. Momentum.
3. Adagrad.
4. RMSProp
5. Adam.

Stochastic Gradient Descent (SGD) is a popular stochastic version of Gradient Descent which is used to train a neural network.

Second-Order Algorithms:

Instead of a first-order derivative, it uses second-order (Hessian) to choose the direction to move in the search space. This algorithm can only be used for objective functions where Hessian Matrix can be approximated or calculated [65].

The algorithms include.

1. For univariate objective functions:
 1. Newton's Method.
 2. Secant Method.
2. For multivariate objective functions:
 1. Quasi-Newton Methods
 2. Davidson-Fletcher-Powell.
 3. Broyden-Fletcher-Goldfarb-Shanno (BFGS).
 4. Limited-memory BFGS (L-BFGS).

Gradient Descent:

Gradient Descent is the steepest descent.

The model's loss function, also known as the cost function, tells you how well the model fits the training data. The more we minimize the cost function, the better the model [86].

The function itself needs to be a differentiable convex function to find the global minima of a function.

The formula to calculate the next position is given below.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \times \nabla f(\mathbf{w}_t)$$

Where, w_t – Current Position

w_{t+1} – Next Position

$\nabla f(w_t)$ – Gradient of the function

α – Learning Rate or the Step Size.

Learning Rate is also known as alpha and step size. It influences how big or small would be the difference from the current position (w_t). The issue is that if we take a high learning rate, we take more significant steps, and we might lose the minima in the function. The algorithm takes tiny steps for a low learning rate, and we might be taking too many steps, which could take a long time [82].

If the gradient doesn't decrease, we can assume that the learning rate is high [82].

Limitations of Gradient Descent:

1. Computing gradients for the entire dataset will take a long time.
2. The larger the dataset, the more memory is required.

To overcome these limitations, An American Statistician and Mathematician, Herbert Robbins, invented the stochastic gradient method, also known as Robbins-Monro Method, in the early 50s. This revolution in AI is now widely used to perform high-level optimization [87, 88].

Stochastic Gradient Descent:

This is a probabilistic approximation method of Gradient Descent. Instead of picking the entire dataset, we randomly choose the observation to calculate the gradients. This approach makes the algorithm faster and more sensible to perform on larger datasets [86].

Though the algorithm is faster than Gradient Descent, we risk the updated value having a high variance, which makes the cost function have more fluctuations for each iteration [86, 87].

Currently, We have a variation of the Stochastic Gradient Descent that could overcome these limitations, like the Mini-Batch Stochastic

Gradient Descent, which randomly chooses n observations instead of just one for each iteration [89].

We'll be using Stochastic Gradient Descent as the optimizer for our predictions. This optimizer performed better than ADAM (Adaptive Moment Estimation), which is also an elongation of the Stochastic Gradient Descent algorithm [90].

Backpropagation:

Originally the algorithm was introduced in 1970. Later, David Rumelhart, Geoffrey Hinton and Ronald Williams published a paper which describes how Backpropagation is faster than the earlier approaches, making the neural network models for solving problems. These days it is considered the workhorse of learning for Neural Networks [67].

The backpropagation algorithm calculates the gradients of the loss function with respect to variables of a model using a partial first-order derivative [68]. The goal is to minimise the loss function, also known as the Sum of Residuals (SSR), by tweaking the weights.

The Sum of Residuals for regression is calculated by using the formula below:

$$\text{Sum of Residuals(SSR)} = \sum_{i=1}^n (\text{Observed}_i - \text{Predicted}_i)^2$$

Where, n – Total number of observed values.

For classification, we have used Binary Cross Entropy to calculate the loss.

Algorithm [67]:

1. Input(x): Set the related activation function for the input layer.
2. Perform Feedforward for each layer and calculate the output.
3. Compute the errors.
4. Back propagate and update weights starting from the last layer for each layer.
5. Perform 2 to 4 till we get the minimum loss.

Now, Backpropagation stops with the following conditions [69]:

Criterion 1.1: if the loss is below the specified threshold.

Criterion 1.2: No drastic changes in the error between two epochs.

Criterion 2: Updating the weights at an epoch is insignificant.

We have an input layer with **11951 perceptron's, a tanh activation function, four hidden layers with 10500, 8000, 500 and 3000 perceptron's, respectively, with tanh and sigmoid activation functions repeating twice and an output layer with a sigmoid activation function. Binary Cross Entropy** is the loss we will try to minimise using **Stochastic Gradient Descent** optimiser and **Backpropagation** to update the weights.

Imbalanced Data:

We may encounter a situation where the target feature is oddly distributed across classes. It is known as the Imbalanced Target feature. If we train the model on such datasets, there is a high chance that the model becomes biased towards the majority class. Hence, Handling Imbalance data becomes essential to building a meaningful and robust machine learning model [84].

The following handling techniques can be used to manage the imbalances with the target feature [84]:

1. Upsampling Minority Class.
2. Downsampling Majority Class.
3. Generate Synthetic Data.
4. Combine Upsampling & Downsampling Techniques.
5. Balanced Class Weight.

Upsampling Minority Class:

Upsampling in machine learning is used to synthetically generate data points of the minority class into the dataset. The counts of the minority class labels are almost the same compared to the other classes. This equalization procedure prevents the model from underfitting. Furthermore, the interaction between the target classes remains unchanged [72]. Various upsampling methods can generate synthetic data points [85].

Upsampling methods include [85]:

1. Random Sampling.
2. SMOTE.
3. BorderLine SMOTE
4. KMeans SMOTE
5. SVM SMOTE.
6. ADASYN
7. SMOTE-NC

Downsampling Majority Class:

Downsampling or Undersampling refers to removing or reducing the majority of class samples to balance the class label [85]. Some of the undersampling techniques include:

1. Random Under Sampling
2. Tomek Links
3. NearMiss Sampling
4. ENN (Edited Nearest Neighbours)

These methods can be implemented using the imblearn python package.

Generate Synthetic Data:

Undersampling techniques are not recommended as it removes the majority of class data points. Generating synthetic data points of minority samples is a type of oversampling technique. The idea is to generate synthetic data points of minority class samples in the nearby region or neighbourhood of minority class samples [85].

SMOTE (Synthetic Minority Over-Sampling Technique) is a popular synthetic data generation oversampling technique which has the following variations:

1. SMOTENC: SMOTE variant for continuous and categorical features.
2. SMOTEN: SMOTE variant for data with only categorical features.
3. Borderline SMOTE: New Synthetic samples will be generated using the borderline samples.
4. SVMSMOTE: Use an SVM algorithm to detect samples to use for generating new synthetic samples.
5. KMeansSMOTE: Over-sample using k-Means clustering before oversample using SMOTE.
6. Adaptive Synthetic (ADASYN): Similar to SMOTE, it generates a different number of samples depending on an estimate of the local distribution of the class to be oversampled.

Combine Oversampling and Undersampling Technique:

Undersampling techniques are not recommended as it removes the majority of class data points. Oversampling is often considered better than undersampling techniques. It combines the undersampling and oversampling techniques to create a robust, balanced dataset for model training [85].

The idea is first to use an oversampling technique to generate synthetic data points and undersampling techniques to remove undesirable and unnecessary synthetic data points [85].

Some of the Oversampling and Undersampling techniques are:

1. Smote-Tomek: Smote (Oversampler) combined with TomekLinks (Undersampler).
2. Smote-ENN: Smote (Oversampler) combined with ENN (Undersampler).

These techniques can be implemented using the imblearn python package.

Balanced Class Weight:

The undersampling technique dismisses the majority class target data points, which results in data loss, whereas upsampling creates artificial data points for the minority class. During machine learning training, one can use the **class_weight** parameter to handle the imbalance in the dataset [85].

This technique can be implemented using the sci-kit learn python package.

Adaptive Synthetic (ADASYN):

Generate synthetic for instances in the minority using the ADASYN algorithm. The number of majority neighbours of each minority instance determines the number of synthetic cases generated from the minority instance [70].

In August 2014, Fernando Nogueira, Guillaume Lemaitre, Dayvid Victor, and Christos Aridas started a project focused on the implementation of SMOTE (Synthetic Minority Over-sampling Technique). Additional under-sampling and over-sampling methods have been implemented, as well as significant changes in the ImBalanced learn to be fully compatible with the Sci-kit learn package in Python [71].

Look at the scatter plots below Fig.14. We can observe that the left plot has fewer data points with class 0. We call such a category "Minority". Now we use the ADASYN algorithm on the minority class and generate synthetic data using a machine learning algorithm called K-Nearest Neighbours.

Let's understand the K-Nearest Neighbour Algorithm.

K- Nearest Neighbours (KNN):

It is a supervised machine learning algorithm used for classification and regression. It is an instance-based model, which is also referred to as memory-based learning [73,74].

Let's take an example to understand how it works; We have three class data points in a 2-D space. We have a new point P_t for which we need to predict the class.

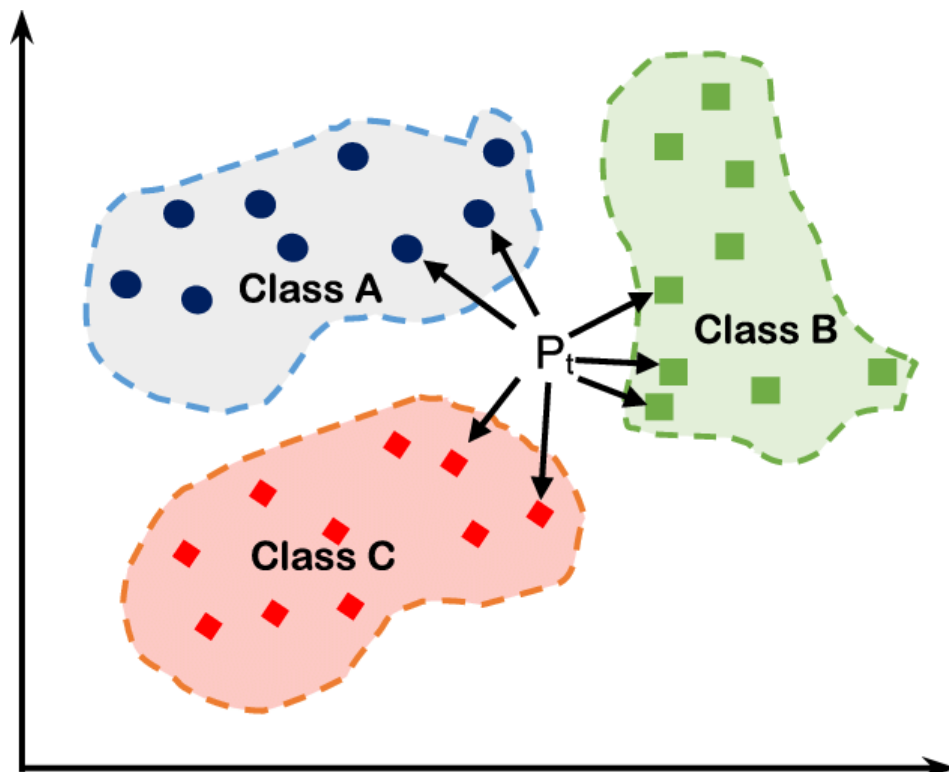


Fig.14: K-Nearest Neighbour classification example [102]

The algorithm to predict the class is [76]:

1. We plot the datapoint (P_i) whose class is unknown.

2. Calculate the distances between the P_i and the other data points. We either use Euclidean, Manhattan or Minkowski distance to calculate the distance [77].
3. Rank the distances from the class points, and the voting predicts class by the nearest neighbour.

For regression, we take the average value of the closest data points. In Fig. 15. we take the averages of all the data points within the dotted circle and predict the value.

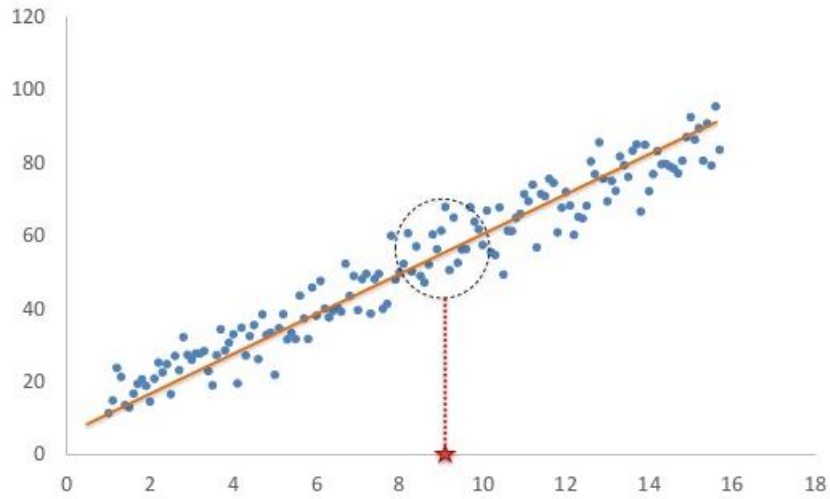


Fig.15: K-Nearest Neighbour regression example [103]

ADSYN Algorithm [82]:

1. The ratio of minority to majority examples is calculated, which can be done by using the formula below.

$$d = \frac{\text{no. of minority class}}{\text{no. of majority class}}$$

2. The total number of synthetic minority data is calculated to generate.

Total number of minority class

$$= (\text{no. of majority class} - \text{no. of minority class})\beta$$

3. Find the k-Nearest Neighbours of each minority example and calculate the r_i value. After this step, each minority example should be associated with a different neighbourhood. The r_i value indicates the dominance of the majority class in each specific neighbourhood. Higher r_i neighbourhoods contain more majority class examples and are more challenging to learn.

$$r_i = \frac{\text{no. of majority}}{k}$$

4. Normalize the r_i values so that the sum of all r_i values equals 1.

$$\hat{r}_i = \frac{r_i}{\sum r_i}$$

$$\sum \hat{r}_i = 1$$

5. Calculate the number of synthetic examples to generate per neighbourhood.

$$G_i = G\hat{r}_i$$

6. Generate G_i data for each neighbourhood. First, take the minority example for the neighbourhood, x_i . Then, randomly select another minority example within that neighbourhood, x_{zi} .

$$s_i = x_i + (x_{zi} - x_i)\lambda$$

Disadvantages:

1. For minority examples that are sparsely dispersed, each neighbourhood may only contain one minority example.
2. The precision of the ADASYN algorithm may suffer due to its adaptability nature.

SMOTE:

SMOTE is performing the same essential task as basic resampling (creating new data points for the minority class), but instead of simply duplicating observations, it establishes new observations along the lines of a randomly chosen point and its nearest neighbours [78, 80].

The algorithm handles unbalanced classification problems. It generates a new data set that addresses the class unbalance problem. It can also run a classification algorithm on this new data set and return the resultant model [79].

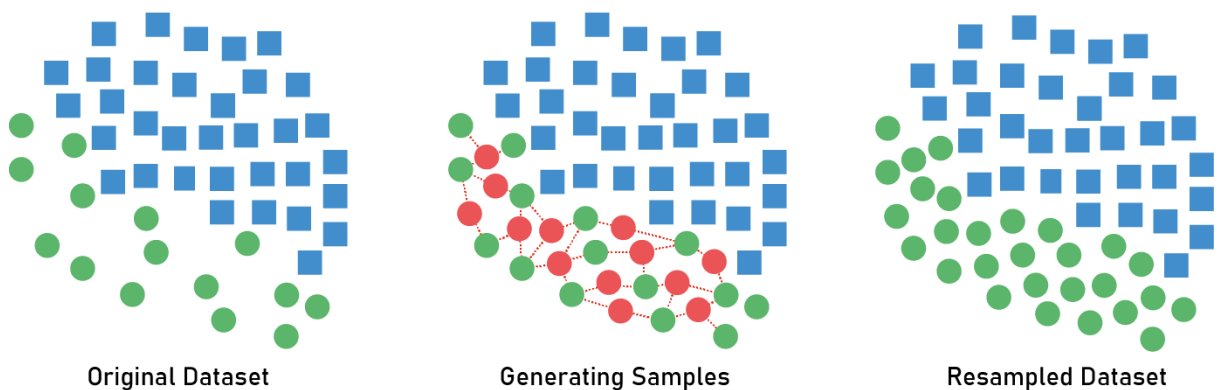


Fig.16: Synthetic Minority Over-sampling Technique (SMOTE) example [104]

Algorithm [81]:

1. We draw a random sample from the minority class.
2. For the observations in this sample, We will identify the K-Nearest Neighbours (KNN).
3. We will then take one of those neighbours and identify the vector between the current data point and the selected neighbour.
4. We multiply the vector by a random number between 0 and 1.
5. We add this to the current data point to obtain the synthetic data point.

Disadvantages [83]:

1. Oversampling uninformative samples.
2. Oversampling of noisy samples.
3. Difficult to determine the number of nearest neighbours, and there is strong blindness in the selection of nearest neighbours for the synthetic samples.

Results:

<i>Data</i>	<i>Model</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>	<i>AUCROC</i>
<u><i>Generic</i></u>	Logistic Regression	0.985584	0.347107	0.25609756	0.34711	0.626381
	Random Forest	0.985401	0.625	0.06097561	0.111111	0.536307
	Neural Network	0.969708	0.088235	0.1097561	0.09783	0.546264
<u><i>ADASYN</i></u>	Logistic Regression	0.933759	0.062305	0.24390244	0.09926	0.594071
	Random Forest	0.969891	0.053763	0.06097561	0.05714	0.522337
	Neural Network	0.970985	0.076923	0.08536585	0.08092	0.534902
<u><i>SMOTE</i></u>	Logistic Regression	0.934307	0.060127	0.23170732	0.09548	0.588343
	Random Forest	0.969891	0.053763	0.06097561	0.05714	0.522337
	Neural Network	0.956204	0.029762	0.06097561	0.04	0.51539

Table 8: Heart Disease Result

Data	Model	Accuracy	Precision	Recall	F1 Score	AUCROC
<u>Generic</u>	Logistic Regression	0.987409	0.347826	0.12903226	0.18824	0.563132
	Random Forest	0.989234	0.8	0.06451613	0.1194	0.532166
	Neural Network	0.963139	0.013889	0.03225806	0.01942	0.532166
<u>ADASYN</u>	Logistic Regression	0.936679	0.029703	0.14516129	0.04932	0.545449
	Random Forest	0.97427	0.024096	0.03225806	0.02759	0.508654
	Neural Network	0.975	0.048193	0.06451613	0.05517	0.508654
<u>SMOTE</u>	Logistic Regression	0.937044	0.0299	0.14516129	0.04959	0.545633
	Random Forest	0.975	0.025316	0.03225806	0.02837	0.509023
	Neural Network	0.974635	0.057471	0.08064516	0.06711	0.509023

Table 9: Type II Diabetes Result

Data	Model	Accuracy	Precision	Recall	F1 Score	AUCROC
<u>Generic</u>	Logistic Regression	0.991058	0.285714	0.04347826	0.07547	0.521279
	Random Forest	0.991423	0	0	NA	0.510778
	Neural Network	0.979562	0.028571	0.04347826	0.03448	0.510778
<u>ADASYN</u>	Logistic Regression	0.961861	0.017751	0.06521739	0.06522	0.517334
	Random Forest	0.987409	0	0	NA	0.497884
	Neural Network	0.973175	0.028037	0.06521739	0.03922	0.498068
<u>SMOTE</u>	Logistic Regression	0.962409	0.018072	0.06521739	0.0283	0.517611
	Random Forest	0.987956	0	0	N/A	0.49816
	Neural Network	0.97646	0	0	N/A	0.49816

Table 10: Alzheimer's Result

Data	Model	Accuracy	Precision	Recall	F1 Score	AUCROC
<u>Generic</u>	Logistic Regression	0.996715	1	0.05263158	0.1	0.526316
	Random Forest	0.996533	N/A	0	N/A	0.5
	Neural Network	0.992336	0.04	0.05263158	0.04545	0.5
<u>ADASYN</u>	Logistic Regression	0.97281	0.007576	0.05263158	0.01325	0.514322
	Random Forest	0.984672	0	0	N/A	0.494049
	Neural Network	0.990146	0	0	N/A	0.494049
<u>SMOTE</u>	Logistic Regression	0.972628	0.007519	0.05263158	0.01316	0.51423
	Random Forest	0.984489	0	0	N/A	0.493957
	Neural Network	0.991423	0	0	N/A	0.493957

Table 11. Parkinson's Result

Data	Model	Accuracy	Precision	Recall	F1 Score	AUCROC
<u>Generic</u>	Logistic Regression	0.993066	0.5	0.15789474	0.24	0.578396
	Random Forest	0.993248	1	0.02631579	0.05128	0.513158
	Neural Network	0.985766	0.02381	0.02631579	0.025	0.509391
<u>ADASYN</u>	Logistic Regression	0.962226	0.043243	0.21052632	0.07175	0.589001
	Random Forest	0.980109	0	0	N/A	0.493477
	Neural Network	0.992336	0	0	N/A	0.499632
<u>SMOTE</u>	Logistic Regression	0.962774	0.043956	0.21052632	0.07273	0.589276
	Random Forest	0.979745	0	0	N/A	0.493293
	Neural Network	0.987409	0.121951	0.13157895	0.12658	0.562482

Table 12. Obesity Result

Conclusion:

The problem we are dealing with is classifying whether someone has a disease. Since the AUC-ROC score is an excellent metric to understand how perfectly the model has classified the class, we'll use it to evaluate the model. We can look at the scores in the Table. 12.

70% of Train Data and 30% Test Data split is used to train and evaluate the model's performance.

	<u>Generic</u>	<u>ADASYN</u>	<u>SMOTE</u>		<u>MAX</u>	<u>MIN</u>	<u>AVG</u>
<u>Heart</u>	0.62638	0.594071	0.588343	LG	0.6264	0.5883	0.6029
	0.53631	0.522337	0.522337	RF	0.5363	0.5223	0.527
	0.54626	0.534902	0.51539	NN	0.5463	0.5154	0.5322
<u>Type II Diabetes</u>	0.56313	0.545449	0.545633	LG	0.5631	0.5454	0.5514
	0.53217	0.508654	0.509023	RF	0.5322	0.5087	0.5166
	0.53217	0.508654	0.509023	NN	0.5322	0.5087	0.5166
<u>Alzheimer's</u>	0.52128	0.517334	0.517611	LG	0.5213	0.5173	0.5187
	0.51078	0.497884	0.49816	RF	0.5108	0.4979	0.5023
	0.51078	0.498068	0.49816	NN	0.5108	0.4981	0.5023
<u>Parkinson's</u>	0.52632	0.514322	0.51423	LG	0.5263	0.5142	0.5183
	0.5	0.494049	0.493957	RF	0.5	0.494	0.496
	0.5	0.494049	0.493957	NN	0.5	0.494	0.496
<u>Obesity</u>	0.5784	0.589001	0.589276	LG	0.5893	0.5784	0.5856
	0.51316	0.493477	0.493293	RF	0.5132	0.4933	0.5
	0.50939	0.499632	0.562482	NN	0.5625	0.4996	0.5238

	<u>Generic</u>	<u>ADASYN</u>	<u>SMOTE</u>
<u>MAX</u>	0.62638	0.594071	0.589276
<u>MIN</u>	0.5	0.493477	0.493293
<u>AVG</u>	0.53377	0.520792	0.523392

Table 13. Performance of the models on different datasets

Abbreviations:

LG – Linear Regression.

RF – Random Forest.

NN – Neural Network

Looking at the figures, we can observe the following:

1. Heart Disease:

In the generic dataset, even after 1.55% of the features had the disease, the Logistic Regression model with an AUC-ROC score of 0.63 performed better than other models with ADASYN and SMOTE datasets. Neural Network trained on SMOTE dataset having 12585 targets for each class performed worst with an AUC-ROC score of 0.52 compared to the different models trained of other datasets. The average AUC-ROC score of a Logistic Regression, Random Forest and Neural Network model trained on a different version of datasets is 0.60, 0.53 and 0.53, respectively.

2. Type II Diabetes:

In the generic dataset, even after 1.12% of the features had the disease, the Logistic Regression model with an AUC-ROC score of 0.56 performed better than other models with ADASYN and SMOTE datasets. Neural Network and Random Forest trained on ADASYN and SMOTE dataset having 12583 and 12584

datapoints having Type II diabetes respectively performed worst with an AUC-ROC score of 0.50 compared to the different models trained of other datasets. The average AUC-ROC score of a Logistic Regression, Random Forest and Neural Network model trained on a different version of datasets is 0.55, 0.52 and 0.52, respectively.

3. Alzheimer's:

In the generic dataset, even after 0.63% of the features had the disease, the Logistic Regression model with an AUC-ROC score of 0.52 performed better than other models with ADASYN and SMOTE datasets. Neural Network and Random Forest trained on ADASYN and SMOTE dataset having 12723 and 12717 observations having Alzheimer's respectively performed worst with an AUC-ROC score of 0.50 compared to the different models trained of other datasets. The average AUC-ROC score of a Logistic Regression, Random Forest and Neural Network model trained on a different version of datasets is 0.52, 0.50 and 0.50, respectively.

4. Parkinson's:

In the generic dataset, even after 0.26% of the features had the disease, the Logistic Regression model with an AUC-ROC score of 0.52 performed better than other models with ADASYN and SMOTE datasets. Compared to other diseases, this dataset has the most imbalanced dataset, with 48 and 18218 observations of someone having Parkinson's and not having it, respectively. Neural Network and Random Forest trained on ADASYN and SMOTE dataset having 12754 and 12757 observations having Parkinson's, respectively, performed worst with an AUC-ROC

score of 0.49 compared to the different models trained of other datasets. The average AUC-ROC score of a Logistic Regression, Random Forest and Neural Network model trained on a different version of datasets is 0.52, 0.50 and 0.50, respectively.

5. Obesity:

In the generic dataset, even after 0.70% of the features had the disease, the Neural Network model with an AUC-ROC score of 0.59 performed better than other models trained with the Generic and SMOTE datasets. Random Forest trained on ADASYN and SMOTE dataset having 12690 and 12697 observations having Obesity respectively performed worst with an AUC-ROC score of 0.49 compared to the different models trained of other datasets. The average AUC-ROC score of a Logistic Regression, Random Forest and Neural Network model trained on a different version of datasets is 0.59, 0.50 and 0.52, respectively.

On average, the model trained on a generic dataset performed well compared to the upsampled ADASYN and SMOTE datasets.

The more the data imbalance, the less the AUC-ROC score and the less meaningful model since it couldn't generalise the class.

If we look at the generic heart disease dataset, we can observe that it had the tiniest imbalance (i.e. 1.55%) compared to other disease datasets. Out of which trained the models with 12585 observations (0: 12585 and 1:201). But we still managed to get a better performing model than others.

If we recall, one of the limitations of these Upsampling algorithms is that they oversample uninformed and noisy samples; hence we get a less efficient model.

My recommendations would be to collect more data about these diseases, resulting in a more balanced dataset which could be later used to build a meaningful model. Even if we had a 60:40 imbalance ratio, we could still perform undersampling and build a more acceptable model. Or, if we have unrestricted access to the dataset, like UK Biobank and eMERGE (Electronic Medical Records and Genomics, USA) Dataset, we could use those structured data.

References:

- [1] World Health Organization, “The Top 10 Causes of Death,” World Health Organization, Dec. 09, 2020.
<https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>
- [2] “Using genomics to predict disease risk,” Broad Institute, Sep. 19, 2019. <https://www.broadinstitute.org/developing-diagnostics-and-treatments/using-genomics-to-predict-disease-risk> (accessed Aug. 14, 2022).
- [3] S. W. Choi, T. S.-H. Mak, and P. F. O’Reilly, “Tutorial: a guide to performing polygenic risk score analyses,” *Nature Protocols*, vol. 15, no. 9, pp. 2759–2772, Jul. 2020, doi: 10.1038/s41596-020-0353-1.
- [4] H. Santoro, “Using AI to find disease-causing genes,” *Scope*, Jun. 10, 2022.
<https://scopeblog.stanford.edu/2022/06/10/using-ai-to-find-disease-causing-genes/> (accessed Aug. 14, 2022)
- [5] R. Dias and A. Torkamani, “Artificial intelligence in clinical and genomic diagnostics,” *Genome Medicine*, vol. 11, no. 1, Nov. 2019, doi: 10.1186/s13073-019-0689-8.
- [6] “WHO’s Science Council launches report calling for equitable expansion of genomics,” [www.who.int](https://www.who.int/news/item/12-07-2022-who-s-science-).
<https://www.who.int/news/item/12-07-2022-who-s-science->

- council-launches-report-calling-for-equitable-expansion-of-genomics (accessed Aug. 14, 2022).
- [7] Accelerating access to genomics for global health, A report of the WHO Science Council.
<https://www.who.int/publications/i/item/9789240052857>
- [8] M. Ashburner et al., “Gene Ontology: tool for the unification of biology,” *Nature Genetics*, vol. 25, no. 1, pp. 25–29, May 2000, doi: 10.1038/75556.
- [9] H. Ritchie and M. Roser, “Causes of Death,” *Our World in Data*, Feb. 2018. <https://ourworldindata.org/causes-of-death>
- [10] “How many cells are in the human body?,”
www.medicalnewstoday.com, Jul. 12, 2017.
<https://www.medicalnewstoday.com/articles/318342#Lack-of-coordinated-effort>
- [11] “What is a cell?: MedlinePlus Genetics,” [medlineplus.gov](https://medlineplus.gov/genetics/understanding/basics/cell).
<https://medlineplus.gov/genetics/understanding/basics/cell>
- [12] “What is a cell?,” @yourgenome · Science website.
<https://www.yourgenome.org/facts/what-is-a-cell/>
- [13] J. A. Segre, “Nucleolus,” *Genome.gov*, 2019.
<https://www.genome.gov/genetics-glossary/Nucleolus>
- [14] “What is DNA?: MedlinePlus Genetics,” [medlineplus.gov](https://medlineplus.gov/genetics/understanding/basics/dna/).
<https://medlineplus.gov/genetics/understanding/basics/dna/>
- [15] National Human Genome Research Institute,
 “Ribonucleic Acid (RNA),” *Genome.gov*, 2019.
<https://www.genome.gov/genetics-glossary/RNA-Ribonucleic-Acid>.
- [16] “What is a chromosome?: MedlinePlus Genetics,”
[medlineplus.gov](https://medlineplus.gov/genetics/understanding/basics/chromosome/), Jan. 19, 2021.
<https://medlineplus.gov/genetics/understanding/basics/chromosome/>
- [17] MedLine Plus, “What is a gene?: MedlinePlus Genetics,”
medlineplus.gov, Sep. 17, 2020.

- <https://medlineplus.gov/genetics/understanding/basics/gene/>
- [18] “Gene Ontology overview,” Gene Ontology Resource.
<http://geneontology.org/docs/ontology-documentation>
 (accessed Aug. 14, 2022).
 - [19] “Heart Disease Is World’s No. 1 Killer,” WebMD.
<https://www.webmd.com/heart-disease/news/20201209/heart-disease-is-worlds-no-1-killer>
 - [20] “Cardiovascular diseases,” www.who.int.
<https://www.who.int/health-topics/cardiovascular-diseases>
 - [21] British Heart Foundation, “Facts and figures,”
[Bhf.org.uk](https://www.bhf.org.uk/what-we-do/news-from-the-bhf/contact-the-press-office/facts-and-figures), 2022. <https://www.bhf.org.uk/what-we-do/news-from-the-bhf/contact-the-press-office/facts-and-figures>.
 - [22] M. D. Huffman et al., “Incidence of Cardiovascular Risk Factors in an Indian Urban Cohort,” *Journal of the American College of Cardiology*, vol. 57, no. 17, pp. 1765–1774, Apr. 2011, doi: 10.1016/j.jacc.2010.09.083.
 - [23] World Health Organization, “Cardiovascular Diseases (CVDs),” [who.int](http://www.who.int), Jun. 11, 2021. [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
 - [24] WebMD, “Heart Disease: Types, Causes, and Symptoms,” WebMD, Feb. 2002. <https://www.webmd.com/heart-disease/heart-disease-types-causes-symptoms>
 - [25] “Inherited heart conditions,” [Nhsinform.scot](https://www.nhsinform.scot/illnesses-and-conditions/heart-and-blood-vessels/conditions/inherited-heart-conditions), 2019.
<https://www.nhsinform.scot/illnesses-and-conditions/heart-and-blood-vessels/conditions/inherited-heart-conditions>
 - [26] WHO, “Diabetes,” World Health Organization, Nov. 10, 2021. <https://www.who.int/news-room/fact-sheets/detail/diabetes>
 - [27] M. A. Khan, M. J. Hashim, J. King, R. D. Govender, H. Mustafa, and J. Al Kaabi, “Epidemiology of Type 2 Diabetes – Global Burden of Disease and Forecasted Trends,” *Journal of*

Epidemiology and Global Health, vol. 10, no. 1, Mar. 2020, doi: 10.2991/jegh.k.191028.001.

- [28] “New research shows the impact of the first lockdown on type 2 diabetes care and deaths,” Diabetes UK.
https://www.diabetes.org.uk/about_us/news/lockdown-diabetes-care-deaths
- [29] “Diabetes in India,” Wikipedia, Jan. 02, 2021.
https://en.wikipedia.org/wiki/Diabetes_in_India
- [30] “India diabetes report 2010 — 2045,”
www.diabetesatlas.org.
<https://www.diabetesatlas.org/data/en/country/93/in.html>
- [31] American Diabetes Association, “Genetics of Diabetes,”
Diabetes.org, 2019.
<https://www.diabetes.org/diabetes/genetics-diabetes>.
- [32] National Institute on Aging, “What Is Alzheimer’s Disease?,” National Institute on Aging, Jul. 08, 2021.
<https://www.nia.nih.gov/health/what-alzheimers-disease>.
- [33] A. D. International, M. Guerchet, and M. Prince,
“Numbers of people with dementia worldwide: An update to the estimates in the World Alzheimer Report 2015,” www.alzint.org, Nov. 2020, [Online]. Available:
<https://www.alzint.org/resource/numbers-of-people-with-dementia-worldwide/>
- [34] National Institute on Aging, “What are the signs of Alzheimer’s Disease?,” National Institute on Aging, May 16, 2017. <https://www.nia.nih.gov/health/what-are-signs-alzheimers-disease>
- [35] National Institute on Aging, “Alzheimer’s Disease Genetics Fact Sheet,” National Institute on Aging, Dec. 24, 2019. <https://www.nia.nih.gov/health/alzheimers-disease-genetics-fact-sheet>

- [36] "Parkinson disease," [www.who.int](https://www.who.int/news-room/fact-sheets/detail/parkinson-disease).
<https://www.who.int/news-room/fact-sheets/detail/parkinson-disease>
- [37] Parkinson's Foundation, "Statistics," Parkinson's Foundation, Jun. 13, 2019.
<https://www.parkinson.org/Understanding-Parkinsons/Statistics>
- [38] Parkinson's UK, "Reporting on Parkinson's: information for journalists," Parkinson's UK, 2020.
<https://www.parkinsons.org.uk/about-us/reporting-parkinsons-information-journalists>
- [39] DR. N, "Parkinsons disease and the ageing indian population," HEALTHCARERADIUS.
<https://www.healthcareradius.in/clinical/28890-parkinsons-disease-and-the-ageing-indian-population>
- [40] F. H. / T. / U. Apr 11, 2022, and 06:45 Ist, "Experts Warn Of 200-300% Rise In Parkinson's In Next Few Decades | Bengaluru News - Times of India," The Times of India.
<https://timesofindia.indiatimes.com/city/bengaluru/experts-warn-of-200-300-rise-in-parkinsons-in-next-few-decades/articleshow/90767559.cms> (accessed Aug. 14, 2022).
- [41] World Health Organization, "Obesity and Overweight," World Health Organization, Jun. 09, 2021.
<https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight>
- [42] Gov.UK, "Overweight Adults," Service.gov.uk, Jul. 25, 2019. <https://www.ethnicity-facts-figures.service.gov.uk/health/diet-and-exercise/overweight-adults/latest>
- [43] ProCon.org, "Global Obesity Levels - Obesity - ProCon.org," Obesity, 2016. <https://obesity.procon.org/global-obesity-levels/>

- [44] M. Venkatrao, R. Nagarathna, V. Majumdar, S. S. Patil, S. Rathi, and H. Nagendra, "Prevalence of Obesity in India and Its Neurological Implications: A Multifactor Analysis of a Nationwide Cross-Sectional Study," *Annals of Neurosciences*, vol. 27, no. 3–4, pp. 153–161, Jul. 2020, doi: 10.1177/0972753120987465.
- [45] I. Today, "India the third most obese country in the world," *India Today*, Jun. 08, 2014.
<https://www.indiatoday.in/mail-today/story/obesity-india-weighs-third-on-obesity-scale-196126-2014-06-08>
- [46] NHS, "Obesity," www.nhs.uk, May 16, 2019.
<https://www.nhs.uk/conditions/obesity/>
- [47] National Institutes of Health, "What causes obesity & overweight?," <http://www.nichd.nih.gov/>, Dec. 2016.
<https://www.nichd.nih.gov/health/topics/obesity/conditioninfo/cause>
- [48] M. Reinagel, "Can You Be Overweight and Still Be Healthy?," *Scientific American*.
<https://www.scientificamerican.com/article/can-you-be-overweight-still-be-healthy> (accessed Aug. 14, 2022).
- [49] L. Sachan, *Data Science in R*. Mumbai: Edvancer Eduventures, pp. 171–176.
- [50] "What is Logistic regression? | IBM," www.ibm.com.
<https://www.ibm.com/uk-en/topics/logistic-regression>
- [51] A. R. Rout, "Advantages and Disadvantages of Logistic Regression," *GeeksforGeeks*, Aug. 25, 2020.
<https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/>
- [52] T. Wood, "Sigmoid Function," *DeepAI*, May 17, 2019.
<https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function>
- [53] L. Sachan, *Machine Learning using Python*. Mumbai: Edvancer Eduventures, Chapter 5: Decision Tree and Random.

- [54] “1.10. Decision Trees — scikit-learn 0.24.2 documentation,” scikit-learn.org. <https://scikit-learn.org/stable/modules/tree.html#minimal-cost-complexity-pruning>
- [55] “Decision Tree Algorithm, Explained - KDnuggets,” KDnuggets, 2020. <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html>
- [56] P. Gupta, “Decision Trees in Machine Learning,” Towards Data Science, May 17, 2017. <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>
- [57] S. Krishnan, “Decision Tree for Classification, Entropy, and Information Gain,” CodeX, Sep. 01, 2021. <https://medium.com/codex/decision-tree-for-classification-entropy-and-information-gain-cd9f99a26e0d>
- [58] “Bootstrapping,” CORP-MIDS1 (MDS). <https://www.mastersindatascience.org/learning/machine-learning-algorithms/bootstrapping/> (accessed Aug. 14, 2022).
- [59] Naresh Kumar, “Advantages and Disadvantages of Random Forest Algorithm in Machine Learning,” Blogspot.com, 2019. <http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-random.html>
- [60] “A Brief History of Deep Learning - DATAVERSITY,” DATAVERSITY, Feb. 07, 2017. <https://www.dataversity.net/brief-history-deep-learning/>
- [61] IBM Cloud Education, “What are Neural Networks?,” www.ibm.com, Aug. 17, 2020. <https://www.ibm.com/uk-en/cloud/learn/neural-networks>
- [62] J. Brownlee, “How to Choose an Activation Function for Deep Learning,” Machine Learning Mastery, Jan. 17, 2021.

- <https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/>
- [63] S. Wu, “What are the best metrics to evaluate your regression model?,” Medium, Jun. 14, 2020.
<https://towardsdatascience.com/what-are-the-best-metrics-to-evaluate-your-regression-model-418ca481755b>
- [64] S. K. Agrawal, “Evaluation Metrics For Classification Model | Classification Model Metrics,” Analytics Vidhya, Jul. 20, 2021.
<https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions>
- [65] J. Brownlee, “How to Choose an Optimization Algorithm,” Machine Learning Mastery, Dec. 22, 2020.
<https://machinelearningmastery.com/tour-of-optimization-algorithms>
- [66] “Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names,” Github.io, 2018.
https://gombu.github.io/2018/05/23/cross_entropy_loss/
- [67] M. A. Nielsen, “Neural Networks and Deep Learning,” Neuralnetworksanddeeplearning.com, 2019.
<http://neuralnetworksanddeeplearning.com/chap2.html>
- [68] J. Brownlee, “Difference Between Backpropagation and Stochastic Gradient Descent,” Machine Learning Mastery, Jan. 31, 2021. <https://machinelearningmastery.com/difference-between-backpropagation-and-stochastic-gradient-descent/>
- [69] N. Tokipris, “When does Back Propagation terminate?,” Analytics Vidhya, Mar. 27, 2021.
<https://medium.com/analytics-vidhya/when-does-back-propagation-terminate-69bf00447674> (accessed Aug. 14, 2022).
- [70] “ADASYN function - RDocumentation,” www.rdocumentation.org.

- <https://www.rdocumentation.org/packages/smotefamily/versions/1.3.1/topics/ADASYN> (accessed Aug. 14, 2022).
- [71] “About us — imbalanced-learn 0.3.0.dev0 documentation,” [glemaitre.github.io](http://glemaitre.github.io/imbalanced-learn/about.html).
<http://glemaitre.github.io/imbalanced-learn/about.html> (accessed Aug. 14, 2022).
- [72] “Handling Imbalanced Data- Machine Learning, Computer Vision, NLP,” Analytics Vidhya, Nov. 07, 2020.
<https://www.analyticsvidhya.com/blog/2020/11/handling-imbalanced-data-machine-learning-computer-vision-and-nlp/>
- [73] “k-Nearest Neighbors (KNN),” [www.ibm.com](https://www.ibm.com/docs/en/ias?topic=procedures-k-nearest-neighbors-knn).
<https://www.ibm.com/docs/en/ias?topic=procedures-k-nearest-neighbors-knn>
- [74] “background of k-Nearest Neighbors (KNN),” [www.ibm.com](https://www.ibm.com/docs/en/ias?topic=knn-background). <https://www.ibm.com/docs/en/ias?topic=knn-background>
- [75] S. K. Gandhi, “Finding out Optimum Neighbours (n) number in the KNN classification using Python,” Analytics Vidhya, Jul. 09, 2021. <https://medium.com/analytics-vidhya/finding-out-optimum-neighbours-n-number-in-the-knn-classification-using-python-9bdcfefff58c>
- [76] A. Christopher, “K-Nearest Neighbor,” Medium, Feb. 03, 2021. <https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4>
- [77] “KNN Classification,” Saedsayad.com, 2019.
https://www.saedsayad.com/k_nearest_neighbors.htm
- [78] “SMOTE — Version 0.9.0,” [imbalanced-learn.org](https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html).
https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
- [79] “SMOTE function | R Documentation,” [Rdocumentation.org](https://www.rdocumentation.org/), 2010.

- <https://www.rdocumentation.org/packages/DMwR/versions/0.4.1/topics/SMOTE>
- [80] “imblearn.over_sampling.SMOTE — imbalanced-learn 0.3.0.dev0 documentation,” [glemaitre.github.io](http://glemaitre.github.io/imbalanced-learn/generated/imblearn.over_sampling.SMOTE.html). http://glemaitre.github.io/imbalanced-learn/generated/imblearn.over_sampling.SMOTE.html (accessed Aug. 14, 2022).
- [81] J. Korstanje, “SMOTE,” Medium, Aug. 30, 2021. <https://towardsdatascience.com/smote-fdce2f605729>
- [82] R. Nian, “An Introduction to ADASYN (with code!),” Medium, Dec. 13, 2019. <https://medium.com/@ruinian/an-introduction-to-adasyn-with-code-1383a5ece7aa>
- [83] Z. Jiang, T. Pan, C. Zhang, and J. Yang, “A New Oversampling Method Based on the Classification Contribution Degree,” *symmetry - MDPI*, Jan. 2021.
- [84] S. Kumar, “5 Techniques to work with Imbalanced Data in Machine Learning,” Medium, Sep. 20, 2021. <https://towardsdatascience.com/5-techniques-to-work-with-imbalanced-data-in-machine-learning-80836d45d30c> (accessed Aug. 17, 2022).
- [85] S. Kumar, “7 Over Sampling techniques to handle Imbalanced Data,” Medium, Nov. 12, 2020. <https://towardsdatascience.com/7-over-sampling-techniques-to-handle-imbalanced-data-ec51c8db349f> (accessed Aug. 17, 2022).
- [86] C. Bento, “Stochastic Gradient Descent explained in real life: predicting your pizza’s cooking time,” Medium, Jun. 02, 2021. <https://towardsdatascience.com/stochastic-gradient-descent-explained-in-real-life-predicting-your-pizzas-cooking-time-b7639d5e6a32>
- [87] H. Robbins and S. Monroe, “A Stochastic Approximation Method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, Sep. 1951, doi: 10.1214/aoms/1177729586.

- [88] Bianp.net, 2016.
https://fa.bianp.net/teaching/2018/eecs227at/stochastic_gradient.html#:~:text=Herbert%20Robbins%20was%20an%20American (accessed Aug. 17, 2022).
- [89] S. Patrikar, “Batch, Mini Batch & Stochastic Gradient Descent,” Medium, Oct. 01, 2019.
<https://towardsdatascience.com/batch-mini-batch-stochastic-gradient-descent-7a62ecba642a>
- [90] Jason Brownlee, “Gentle Introduction to the Adam Optimization Algorithm for Deep Learning,” Machine Learning Mastery, Jul. 02, 2017.
<https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
- [91] A. Limited, “Medical illustration of elements of human cell Stock Vector Image & Art - Alamy,” www.alamy.com.
<https://www.alamy.com/medical-illustration-of-elements-of-human-cell-image230598171.html?imageid=BAD9D0B2-41B7-4E4E-AF8E-73B0CC136E51&p=648830&pn=1&searchId=a67faddbe6071129e151c5095d311ea&searchtype=0> (accessed Aug. 18, 2022).
- [92] Medline Plus, “What is DNA?,” medlineplus.gov, Jan. 19, 2021.
<https://medlineplus.gov/genetics/understanding/basics/dna/>
- [93] Microbenotes.com, 2022. <https://microbenotes.com/wp-content/uploads/2018/08/RNA-Structure.jpeg> (accessed Aug. 18, 2022).
- [94] “Are genes located on chromosomes?,” Quora.
<https://www.quora.com/Are-genes-located-on-chromosomes>
- [95] “Implantable Cardioverter Defibrillator (ICD) | Liverpool Heart and Chest Hospital,” www.lhch.nhs.uk.
<https://www.lhch.nhs.uk/our-services/cardiac-diagnostics/implantable-cardioverter-defibrillator-icd/> (accessed Aug. 18, 2022).

- [96] D. S. Bawa, "Chart of Normal Blood Sugar Levels for Adults with Diabetes Age Wise," Breathe Well-Being, May 31, 2021. <https://www.breathewellbeing.in/blog/chart-of-normal-blood-sugar-levels-for-adults-with-diabetes/>
- [97] NIH, "What Is Alzheimer's Disease?," National Institute on Aging, May 16, 2017. <https://www.nia.nih.gov/health/what-alzheimers-disease>
- [98] A. Noyce and P. Lewis, "Parkinson's: four unusual signs you may be at risk," The Conversation. <https://theconversation.com/parkinsons-four-unusual-signs-you-may-be-at-risk-112035>
- [99] C. Seltzer and MD, "BMI Chart For Men: Is BMI Misleading?," BuiltLean, Jul. 17, 2013. <https://www.builtlean.com/bmi-chart/>
- [100] "machine learning - When to use a neural network with just one output neuron and when with multiple output neurons?," Stack Overflow. <https://stackoverflow.com/questions/65600387/when-to-use-a-neural-network-with-just-one-output-neuron-and-when-with-multiple>
- [101] S. Jadon, "Introduction to Different Activation Functions for Deep Learning," Medium, Feb. 03, 2022. <https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092>
- [102] S. K. Gandhi, "Finding out Optimum Neighbours (n) number in the KNN classification using Python," Analytics Vidhya, Jul. 09, 2021. <https://medium.com/analytics-vidhya/finding-out-optimum-neighbours-n-number-in-the-knn-classification-using-python-9bdcfeff58c>
- [103] I. F. C. Juchnowicz, Chapter 2 Machine Learning tools | ADVANCED REGRESSION AND PREDICTION: MACHINE LEARNING TOOLS. [Online]. Available:

<https://bookdown.org/f100441618/bookdown-regresion/ml-tools.html>

- [104] Z. Jefferson, “Bank Data: SMOTE,” Analytics Vidhya, Aug. 31, 2020. <https://medium.com/analytics-vidhya/bank-data-smote-b5cb01a5e0a2> (accessed Aug. 18, 2022).

GitHub Repository Link:

github.com/nileshredz/Human-Disease-using-Genomic-Dataset