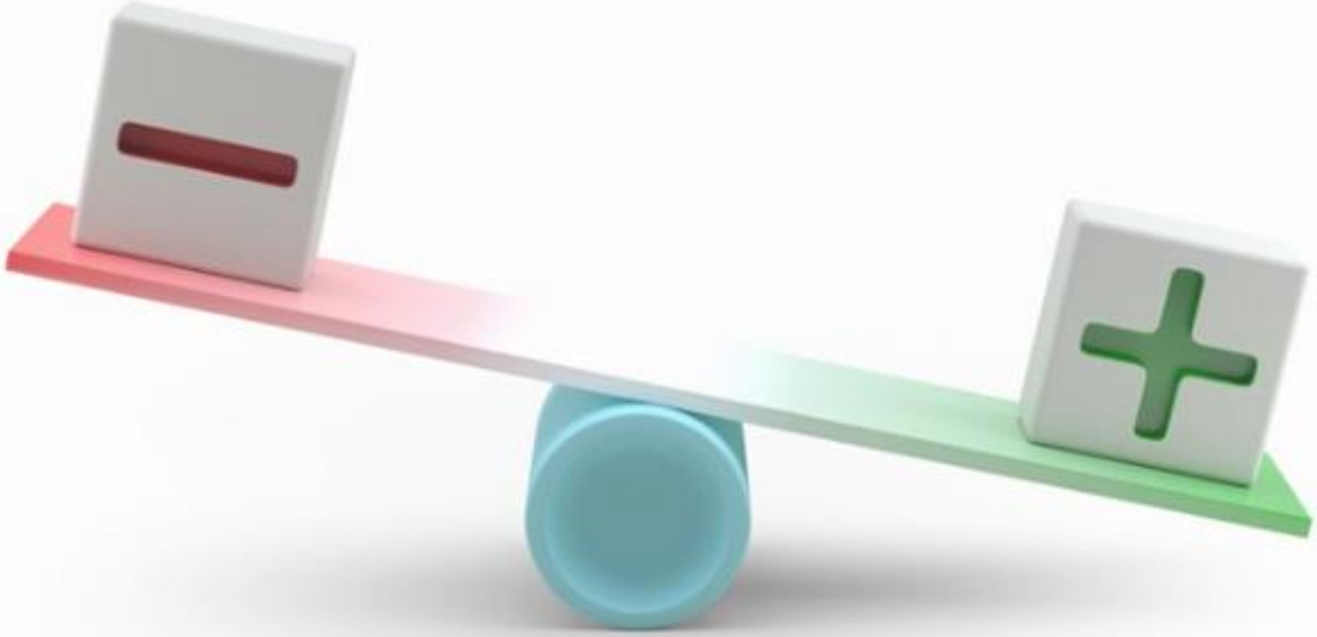


Metrics and Techniques for Handling Imbalanced Datasets

Under the Supervision of:

Dr. Doaa Mahmoud

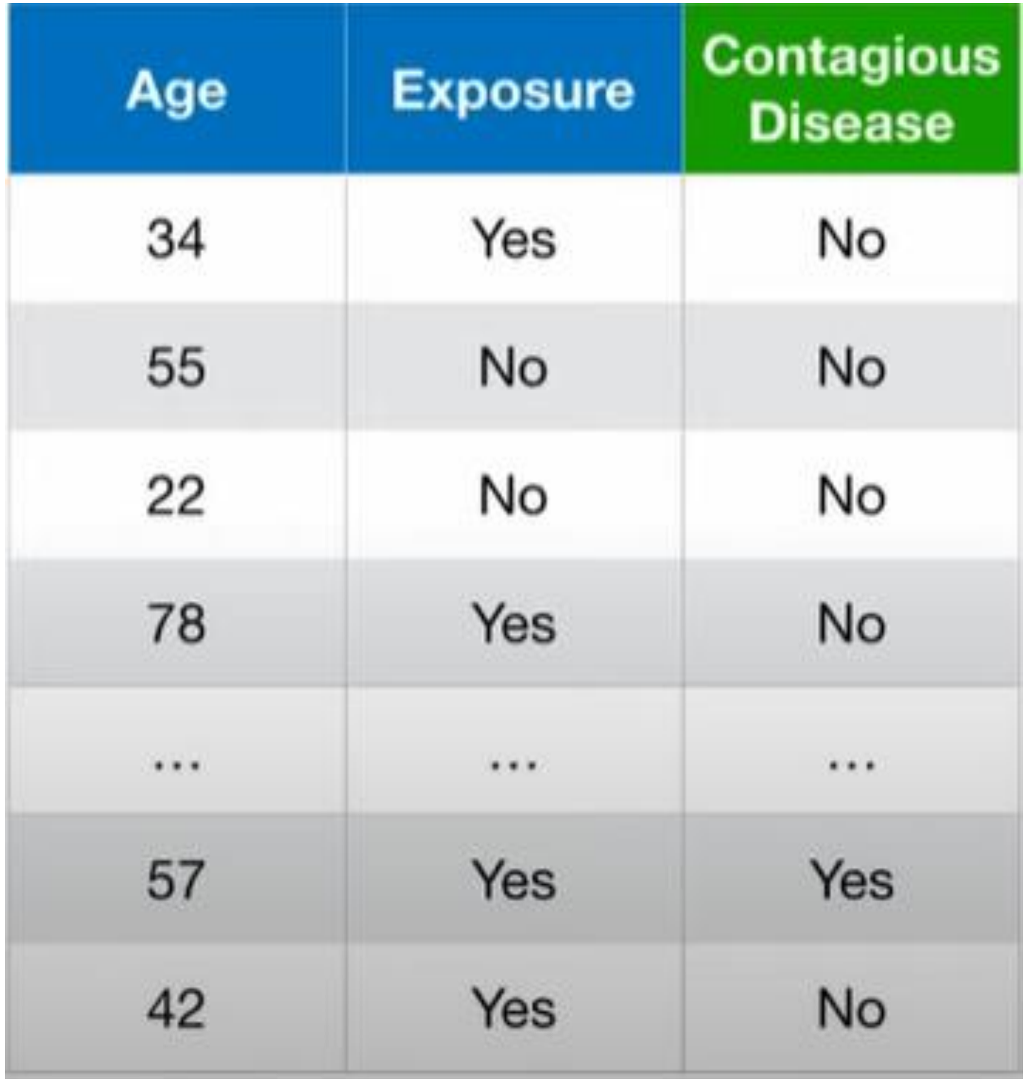
Collaborators

- Eng. Raed Habib
  https://www.linkedin.com/in/raedhabib
  https://github.com/RaedHabib - https://www.kaggle.com/raedhabib
- Eng. Zeyad Elsayed Ibrahim
  https://www.linkedin.com/in/ziad-elsayed-49b740231
  https://www.kaggle.com/ziadelsayed002
- Eng. Zeyad Mohammed Mahmoud
  https://www.linkedin.com/in/ziad-muhammad-249986283
  https://www.kaggle.com/ziadmuhammadbarro
- Eng. Zeyad Elsayed Usf
  https://www.linkedin.com/in/zeyadusf/
  https://github.com/zeyadusf - https://www.kaggle.com/zeyadusf

Agenda

- Introduction .
- Technique for Handling Imbalanced Data .
- Dataset Overview .
- Initial Questions to be Answered.
- Exploratory Data Analysis.
- Problems and Challenges.
- Modeling and Results.
- Conclusion.

Introduction



Age	Exposure	Contagious Disease
34	Yes	No
55	No	No
22	No	No
78	Yes	No
...	...	...
57	Yes	Yes
42	Yes	No

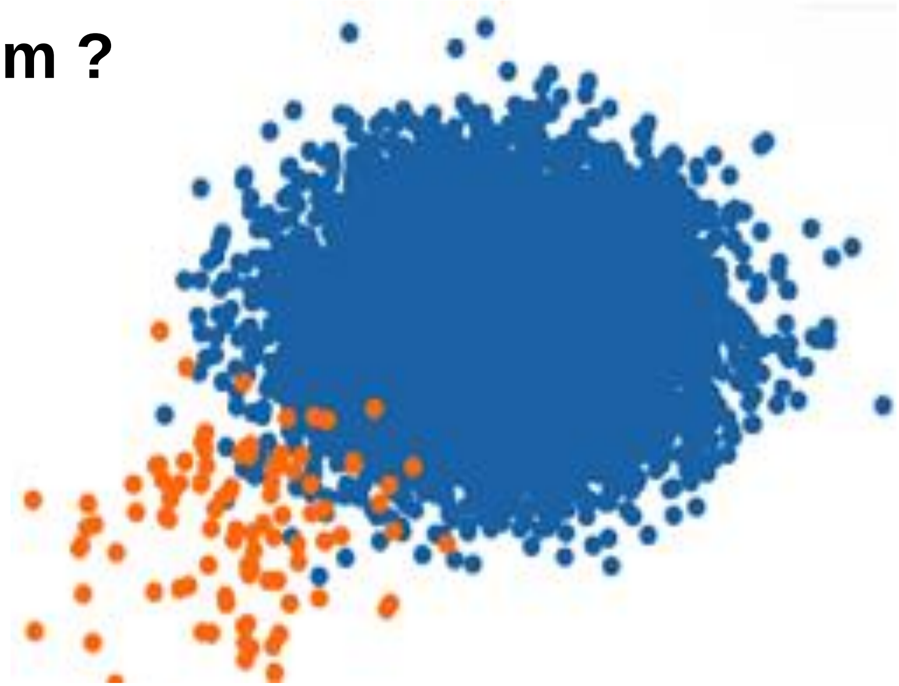
What if, we were trying to decide if someone has a rare, but contagious disease?
According to the presented image, we used **"Age"** and **"Exposure"** to predict whether or not someone has a rare, but contagious disease. If we simply predicted that no one had this rare disease, we would be correct almost all of the time.

But we would misclassify every single person that has the rare disease; because our model became biased to the majority of people in the training dataset that don't have the disease, and that would be terrible.

Most of the sophisticated classification algorithms or methods are expecting a relatively equal proportion of **Yeses** and **Nos** in the training dataset.

Introduction

What is Imbalanced Data For a Classification Problem ?



A classification data set with skewed class proportions is called imbalanced. Classes that make up a large proportion of the data set are called **majority classes ,** Those that make up a smaller proportion are **minority classes .** (where one class is much more abundant than the other).

What is effect imbalance data on model ?

- Algorithms may get biased towards the majority class and thus tend to predict output as the majority class.
- Minority class observations look like noise to the model and are ignored by the model.
- Imbalanced dataset gives misleading accuracy score.

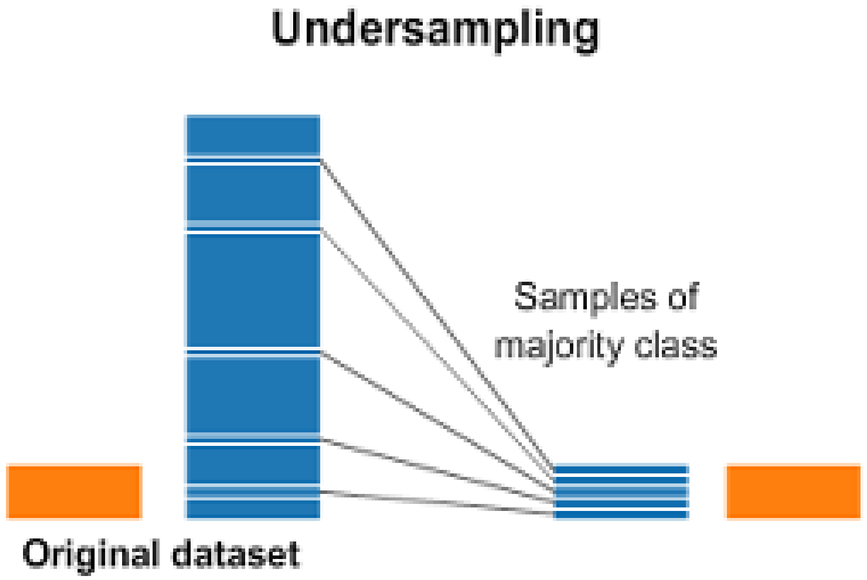
Technique for Handling Imbalanced Data

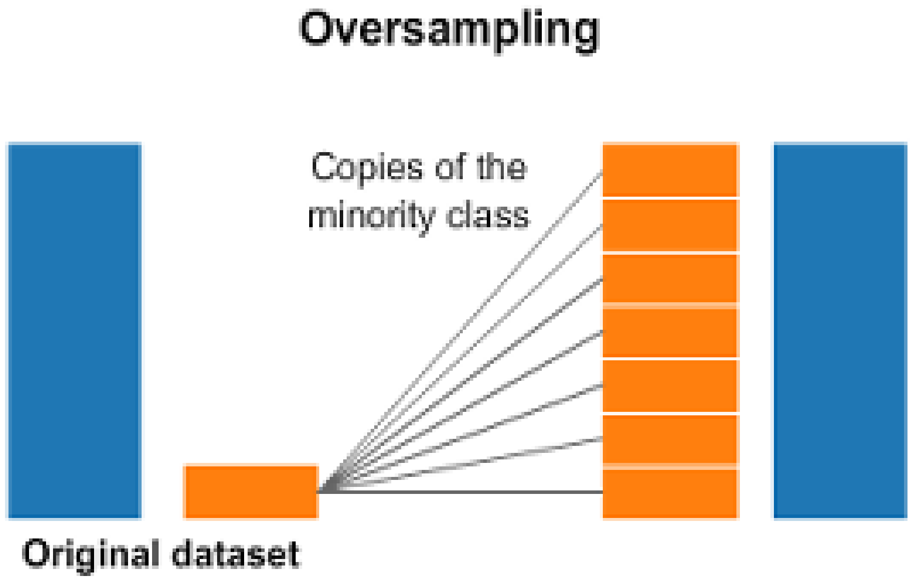
We have some common techniques :

- **Under-sampling** can be defined as removing some observations of the majority class. This is done until the majority and minority class is balanced out.

- **Oversampling** can be defined as adding more copies to the minority class. Oversampling can be a good choice when you don't have a ton of data to work with.

- **Combine Oversampling and Under-sampling Techniques** The idea is to first use an oversampling technique to create duplicate and artificial data points and use under-sampling techniques to remove noise or unnecessary generated data points.



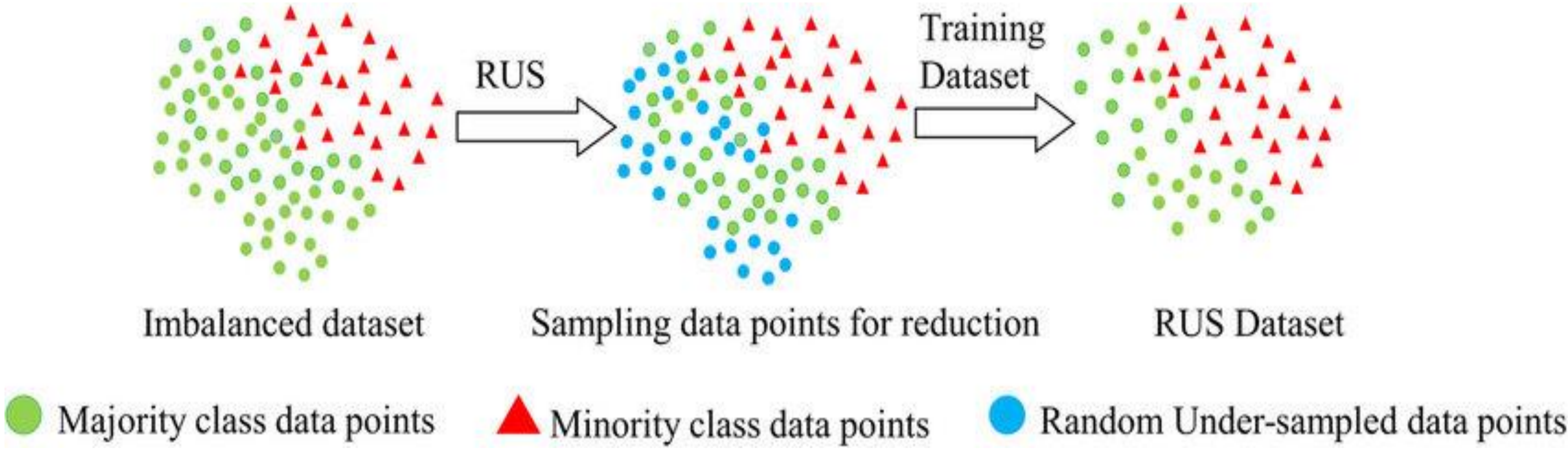


Under Sampling

Now we will show the possible methods using Python libraries to Handling Under-sampling

1- Random Under-Sampling

Randomly selecting examples from the majority class and deleting them from the training dataset. In the random under-sampling, the majority class instances are discarded at random until a more balanced distribution is reached.

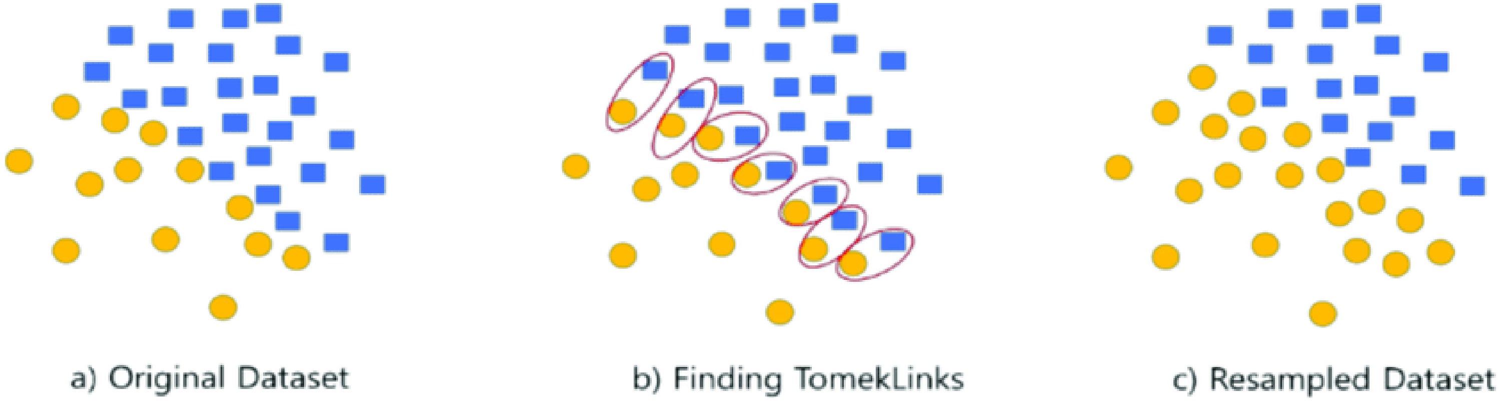


RUS
Training Dataset
Imbalanced dataset
Sampling data points for reduction
RUS Dataset
Majority class data points
Minority class data points
Random Under-sampled data points

2- Tomek Links

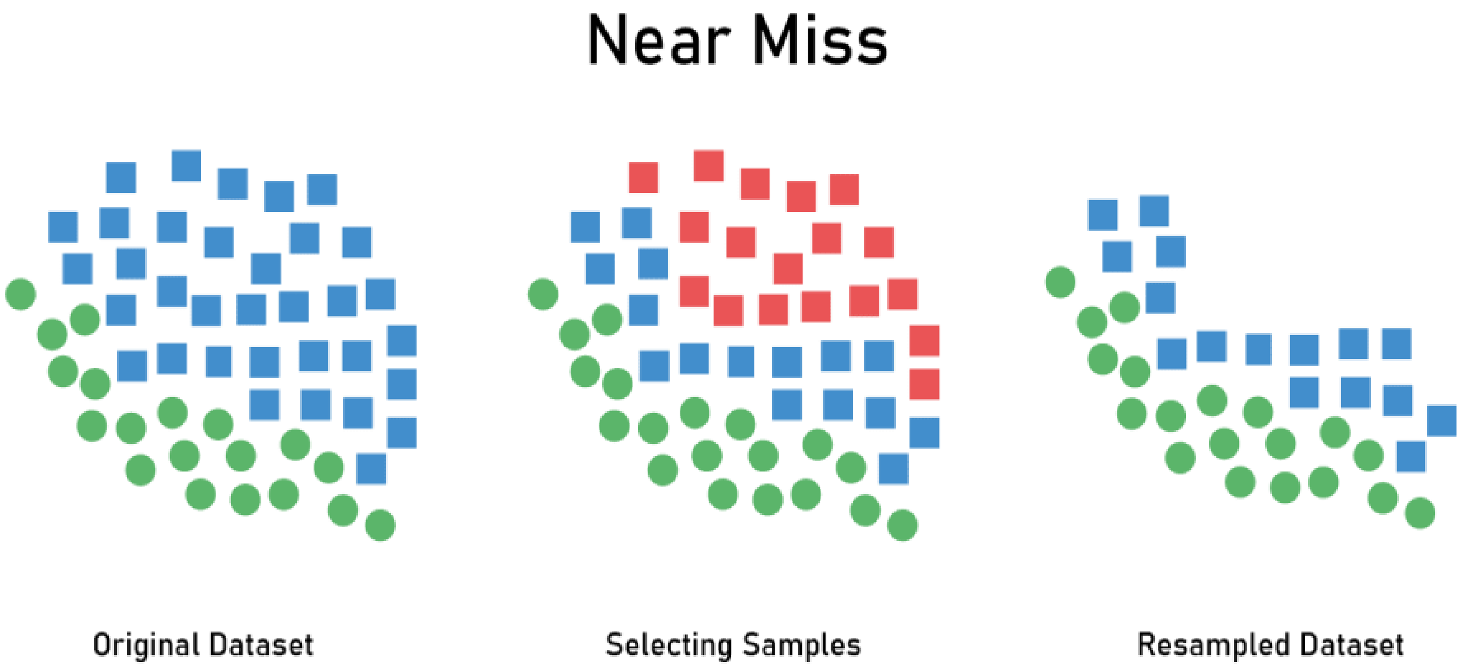
Tomek links are pairs of very close instances but of opposite classes. Removing the instances of the majority class of each pair increases the space between the two classes, facilitating the classification process.

Tomek's link exists if the two samples are the nearest neighbors of each other.



3-NearMiss

NearMiss is an under-sampling technique. Instead of resampling the Minority class, using a distance will make the majority class equal to the minority class.



4-Balanced Class Weight.

The under-sampling technique removes the majority class data points which results in data loss, whereas up-sampling creates artificial data points of the minority class. During the training of machine learning, one can use (class_weight) parameter to handle the imbalance in the dataset.

The whole purpose is to penalize the misclassification made by the minority class by setting a higher class weight and at the same time reducing weight for the majority class.

5- BalancedBaggingClassifier.

is the same as a sklearn classifier but with additional balancing. It includes an additional step to balance the training set at the time of fit for a given sampler. This classifier takes two special parameters “sampling_strategy” and “replacement”.

The sampling_strategy decides the type of resampling required (e.g. ‘majority’ – resample only the majority class, ‘all’ – resample all classes, etc) and replacement decides whether it is going to be a sample with replacement or not.

6- ENN (Edited Nearest Neighbours).

is an undersampling method technique that remove the majority class to match the minority class.

ENN works by removing samples whose class label differs from the class of the majority of their k nearest neighbors**.**

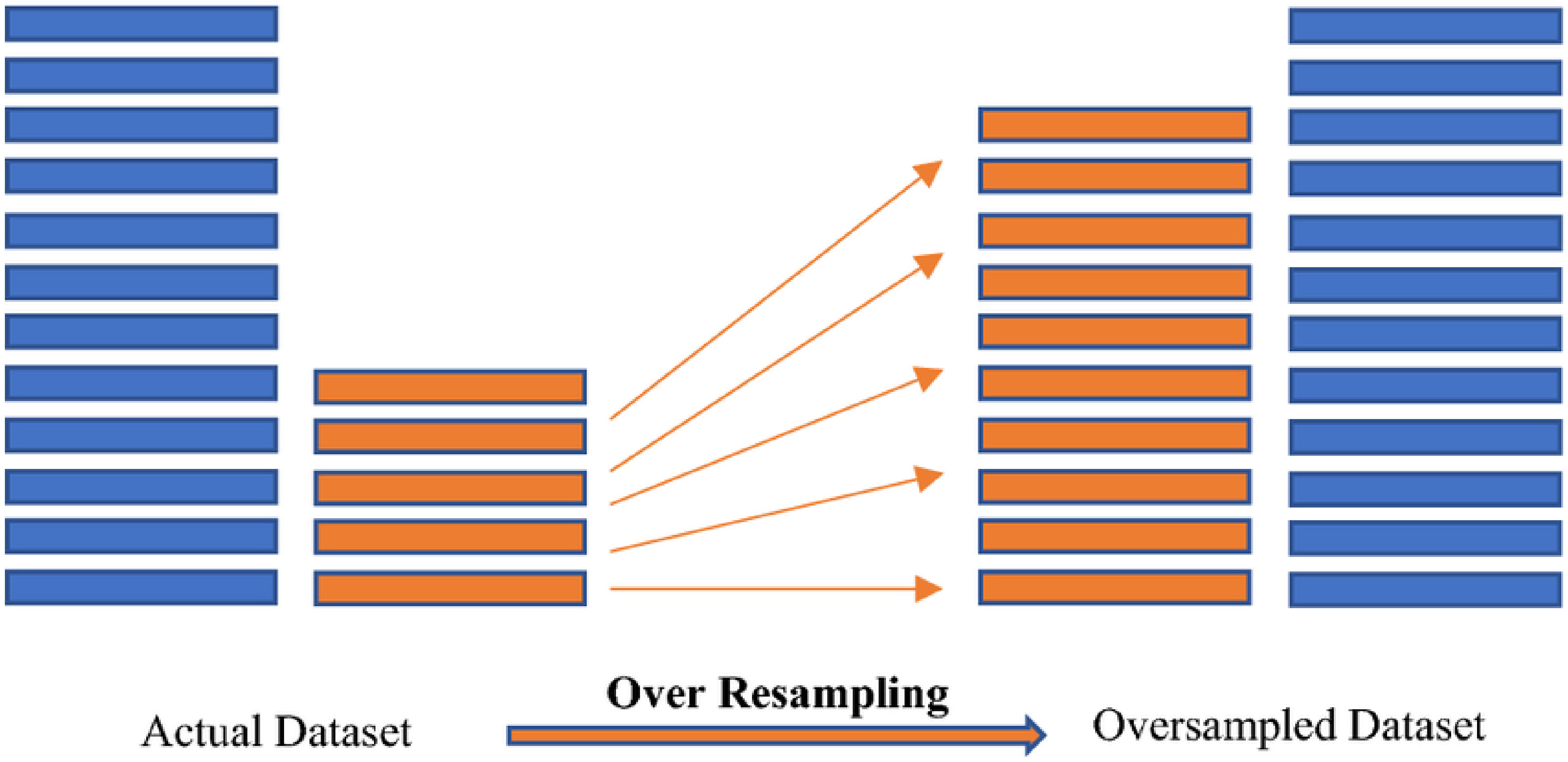
Over Sampling

Now we will show the possible methods using Python libraries to Handling Oversampling

1- Random Over Sampling

One way to fight imbalanced data is to generate new samples in the minority classes.

The most naive strategy is to generate new samples by random sampling with the replacement of the currently available samples.

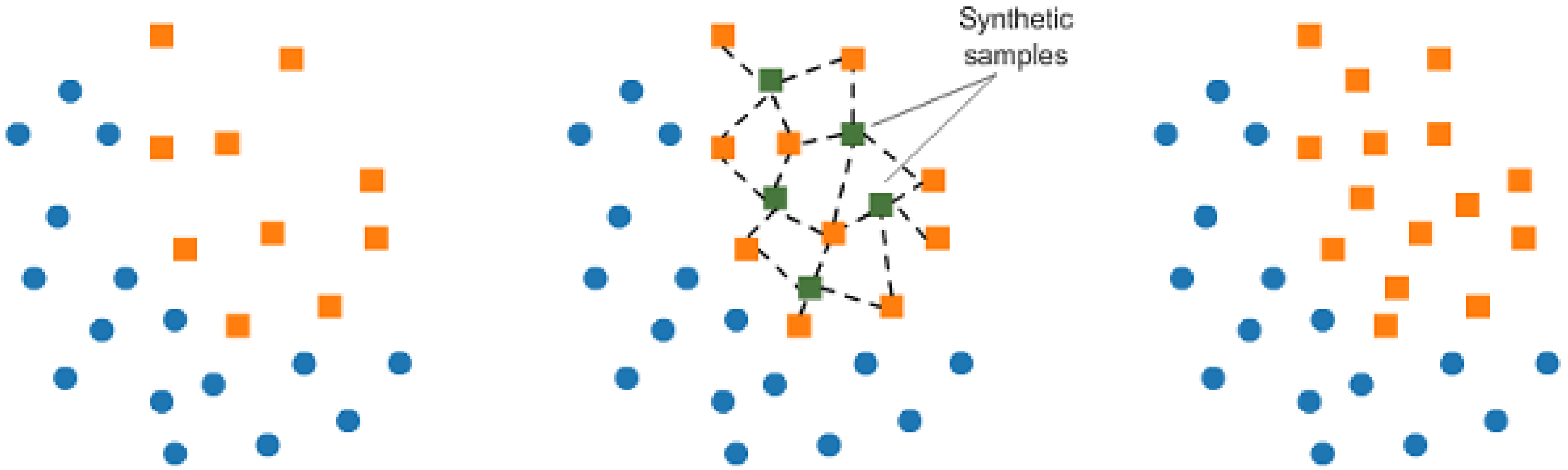


2- Synthetic Minority Oversampling Technique (SMOTE)

This technique generates synthetic data for the minority class. SMOTE (Synthetic Minority Oversampling Technique) works by randomly picking a point from the minority class and computing the k-nearest neighbors for this point. The synthetic points are added between the chosen point and its neighbors.

SMOTE algorithm works in 4 simple steps:

- Choose a minority class as the input vector.
- Find its k nearest neighbors (k_neighbors is specified as an argument in the SMOTE() function).
- Choose one of these neighbors and place a synthetic point anywhere on the line joining the point under consideration and its chosen neighbor.
- Repeat the steps until the data is balanced



Combine Oversampling and Undersampling Techniques.

Undersampling techniques is not recommended as it removes the majority class data points.

Oversampling techniques are often considered better than undersampling techniques.

The idea is to combine the under-sampling and oversampling techniques to create a robust balanced dataset fit for model training.

The idea is to **first** use an **oversampling** technique to create duplicate and artificial data points **and** use **undersampling** techniques to remove noise or unnecessary generated data points.

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

A combination of over-sampling the minority (abnormal) class and under-sampling the majority (normal) class can achieve better classifier performance than only under-sampling the majority class. This method was first introduced by Batista et al. (2003).

2- Smote-ENN: Smote (Oversampler) combined with ENN (Undersampler)

SMOTE + ENN is another hybrid technique where more no. of observations are removed from the sample space. Here, ENN is yet another undersampling technique where the nearest neighbors of each of the majority class is estimated

Other Methods

Feel free to read about these other methods:

1. Change the Performance Metric.
2. Penalize Algorithms (Cost-Sensitive Training).
3. Threshold Moving.

Dataset Overview

According to the Centers for Disease Control and Prevention (CDC), heart disease is one of the leading causes of death for people of most races in the US (African Americans, American Indians and Alaska Natives, and white people). About half of all Americans (47%) have at least 1 of 3 key risk factors for heart disease: high blood pressure, high cholesterol, and smoking.

Other key indicator include diabetic status, obesity (high BMI), not getting enough physical activity or drinking too much alcohol. Detecting and preventing the factors that have the greatest impact on heart disease is very important in healthcare.

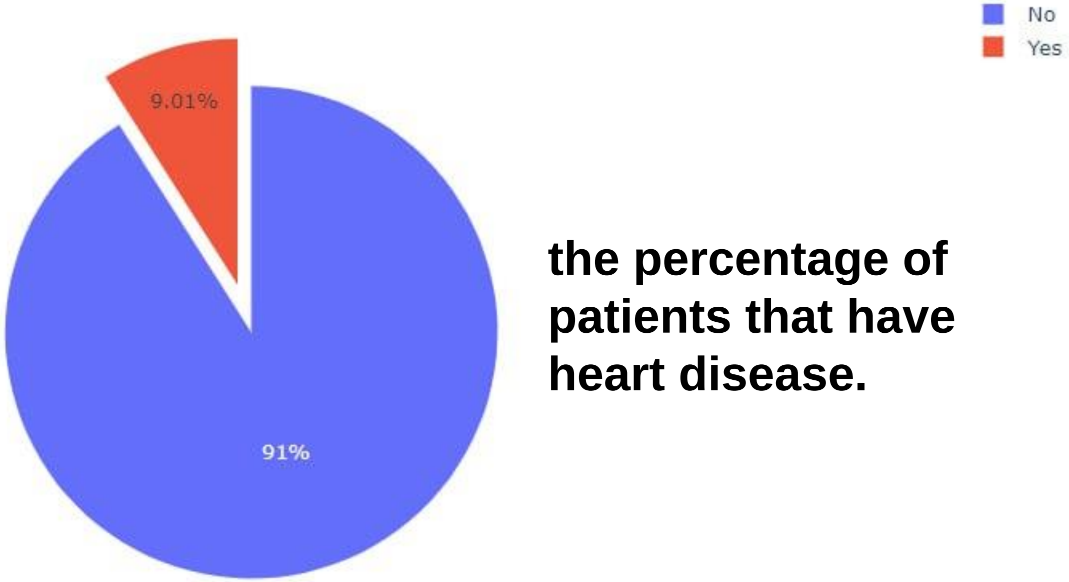
Description of Columns

Columns Name	Description
HeartDisease	*heart disease (CHD) or myocardial infarction (MI)*
BMI	Computed body mass index
Smoking	Smoked at Least 100 Cigarettes
AlcoholDrinking	Heavy Alcohol Consumption
Stroke	Ever Diagnosed with a Stroke
PhysicalHealth	Number of Days Physical Health Not Good
MentalHealth	Number of Days Mental Health Not Good
DiffWalking	*Do you have serious difficulty walking or climbing stairs?*
Sex	Are you male or female?

Columns Name	Description
AgeCategory	Reported age in five-year age categories
Race	Imputed race
Diabetic	had diabetes ?
PhysicalActivity	Exercise in Past 30 Days
GenHealth	General Health
SleepTime	On average, how many hours of sleep?
Asthma	Had Asthma?
KidneyDisease	Not including kidney stones, bladder infection or incontinence, were you ever told you had kidney disease?
SkinCancer	you had skin cancer?

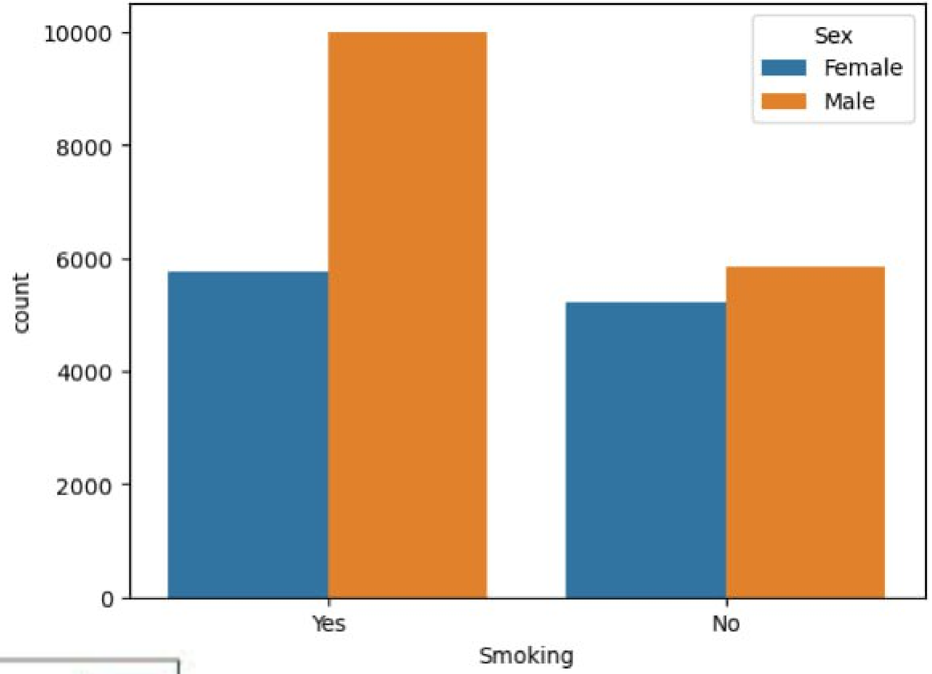
Exploratory Data Analysis

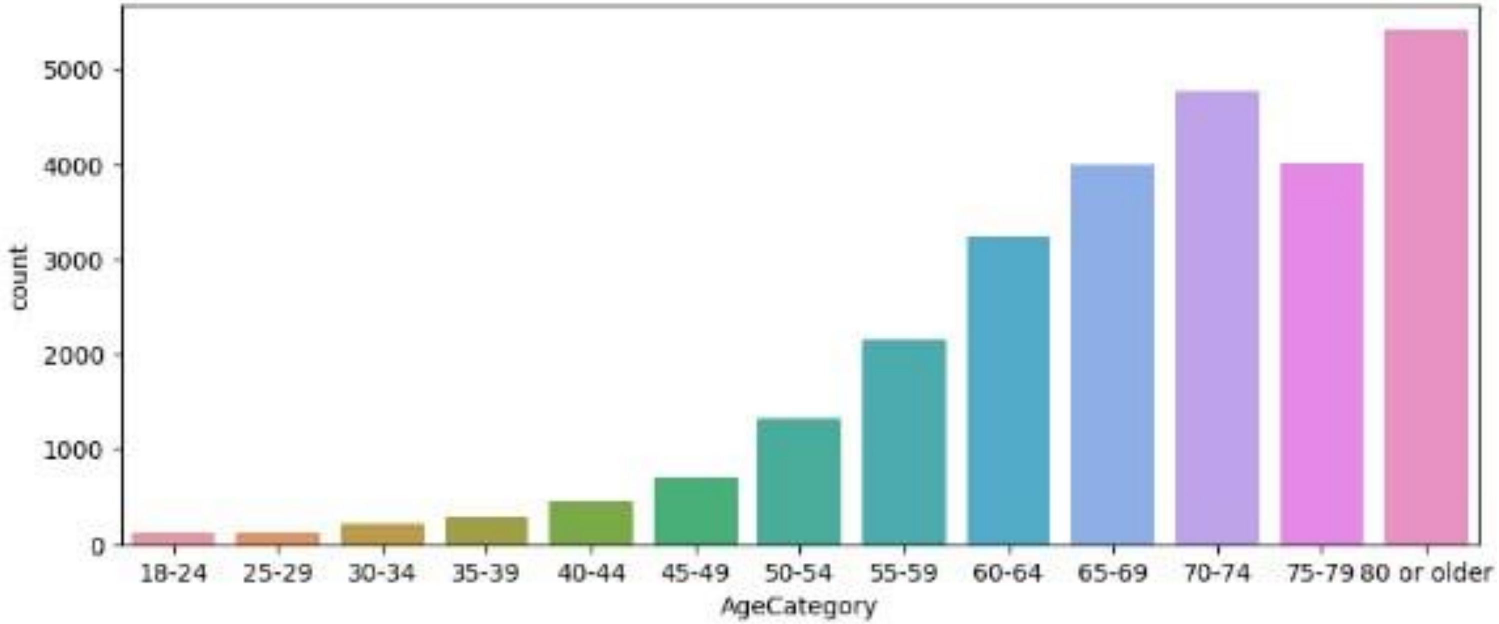
- What is the percentage of patients that have heart disease?
- What is the most people with heart disease?
- What is the most people without heart disease?
- How many patient that have heart disease are smoking ?
- What is general health of people that heart disease ?
- What is percentage of men and women with heart disease?



the percentage of patients that have heart disease.

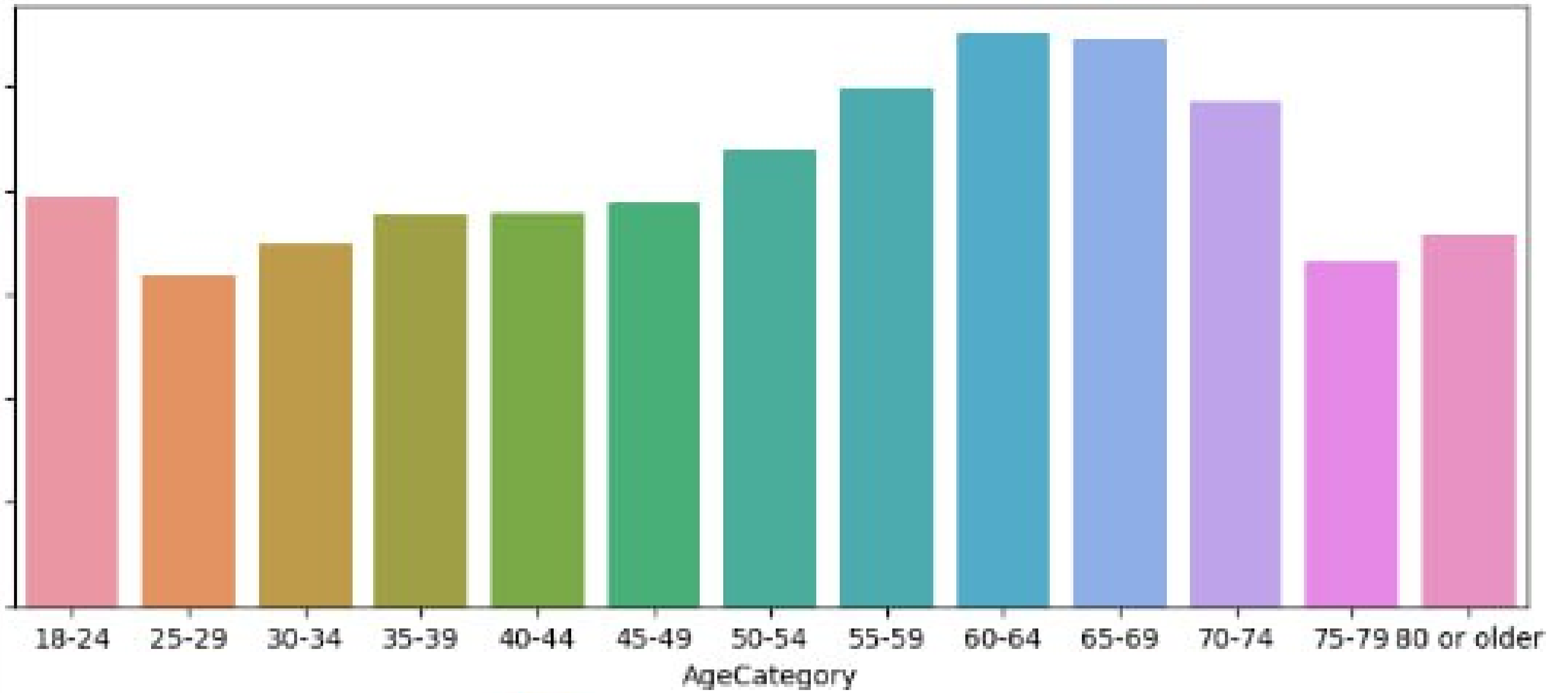
patient that have heart disease are smoking.

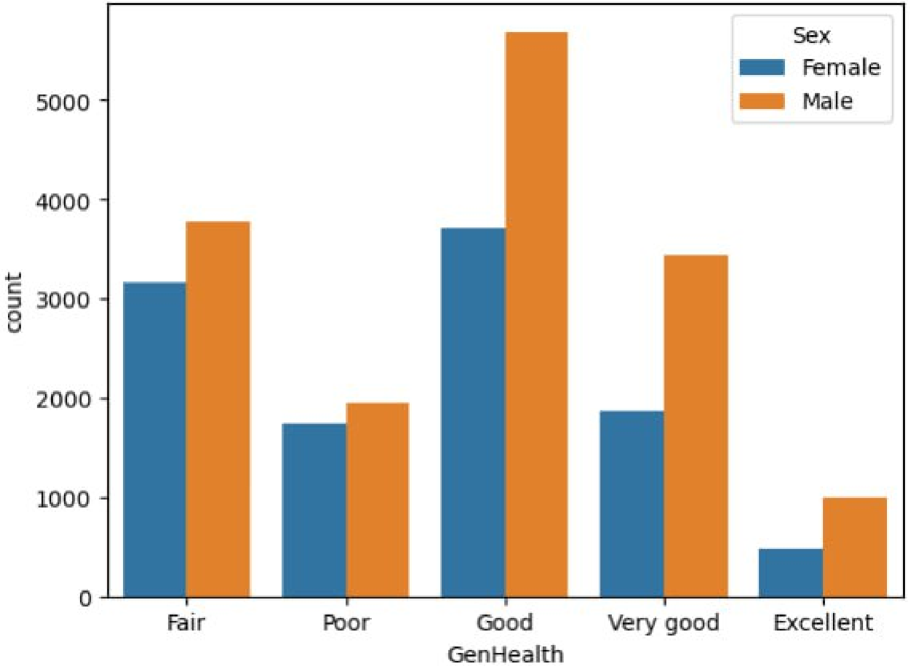




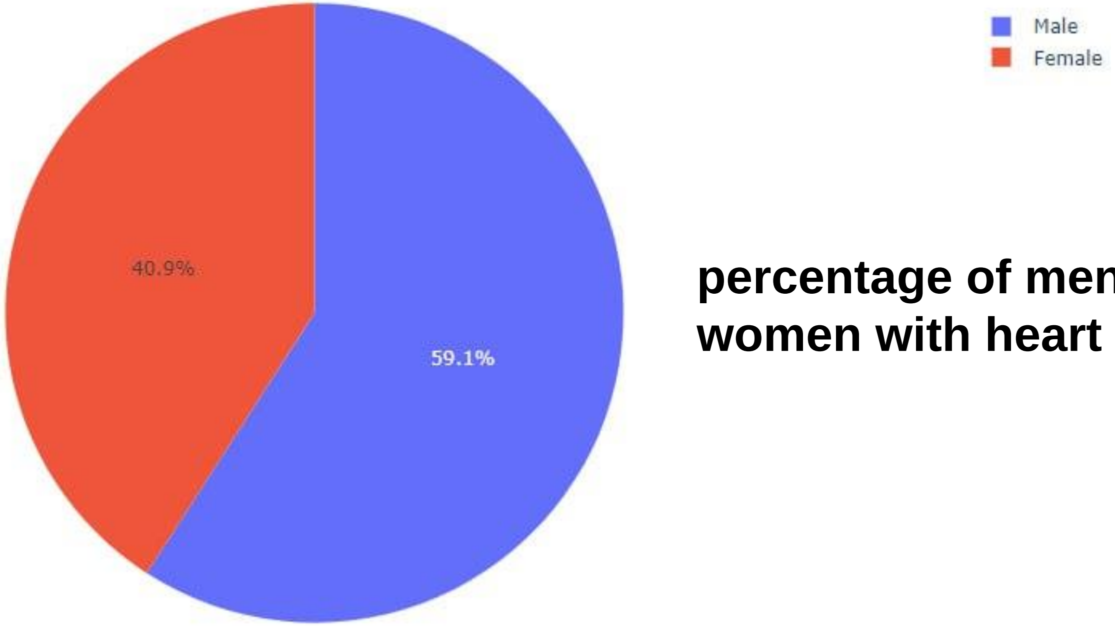
the most people with heart disease age are between 80 or older and least patients are 18 to 29

the most people without heart disease age are between 60 to 64 and least patient age are 25 to 29.





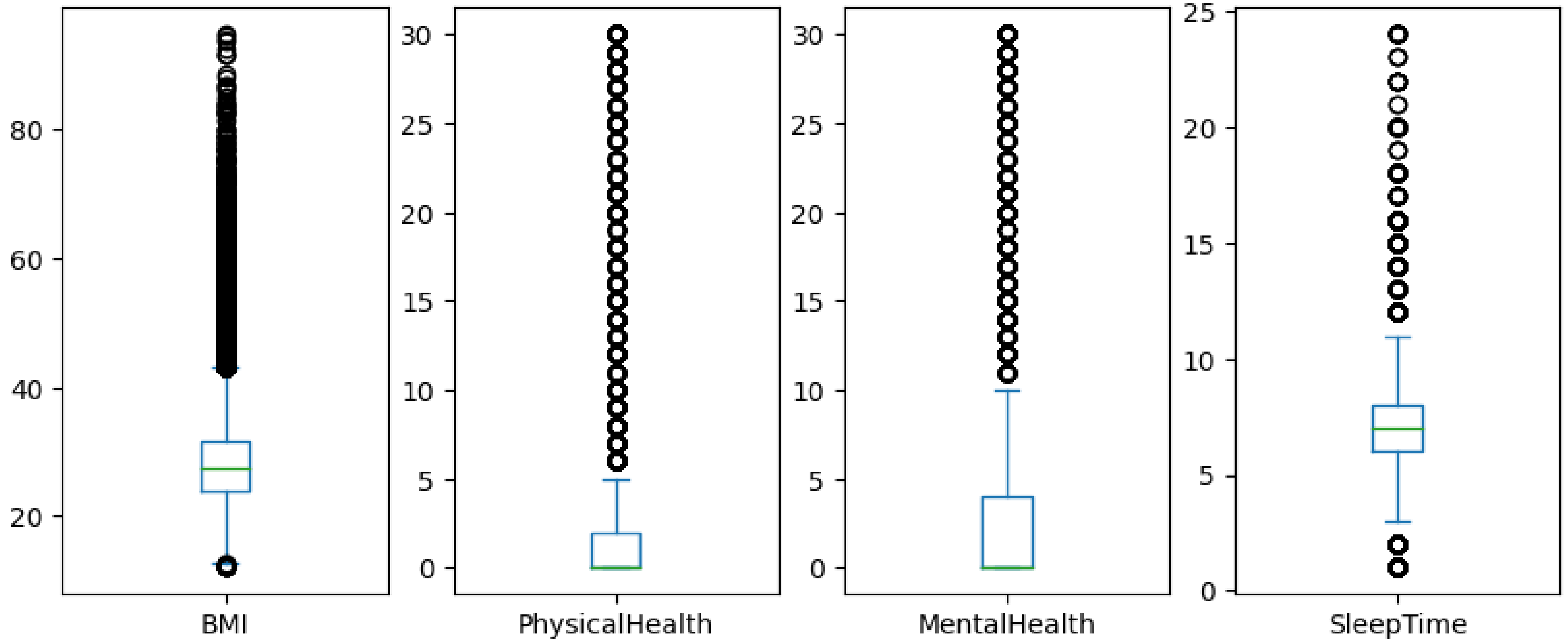
general health of people that heart disease



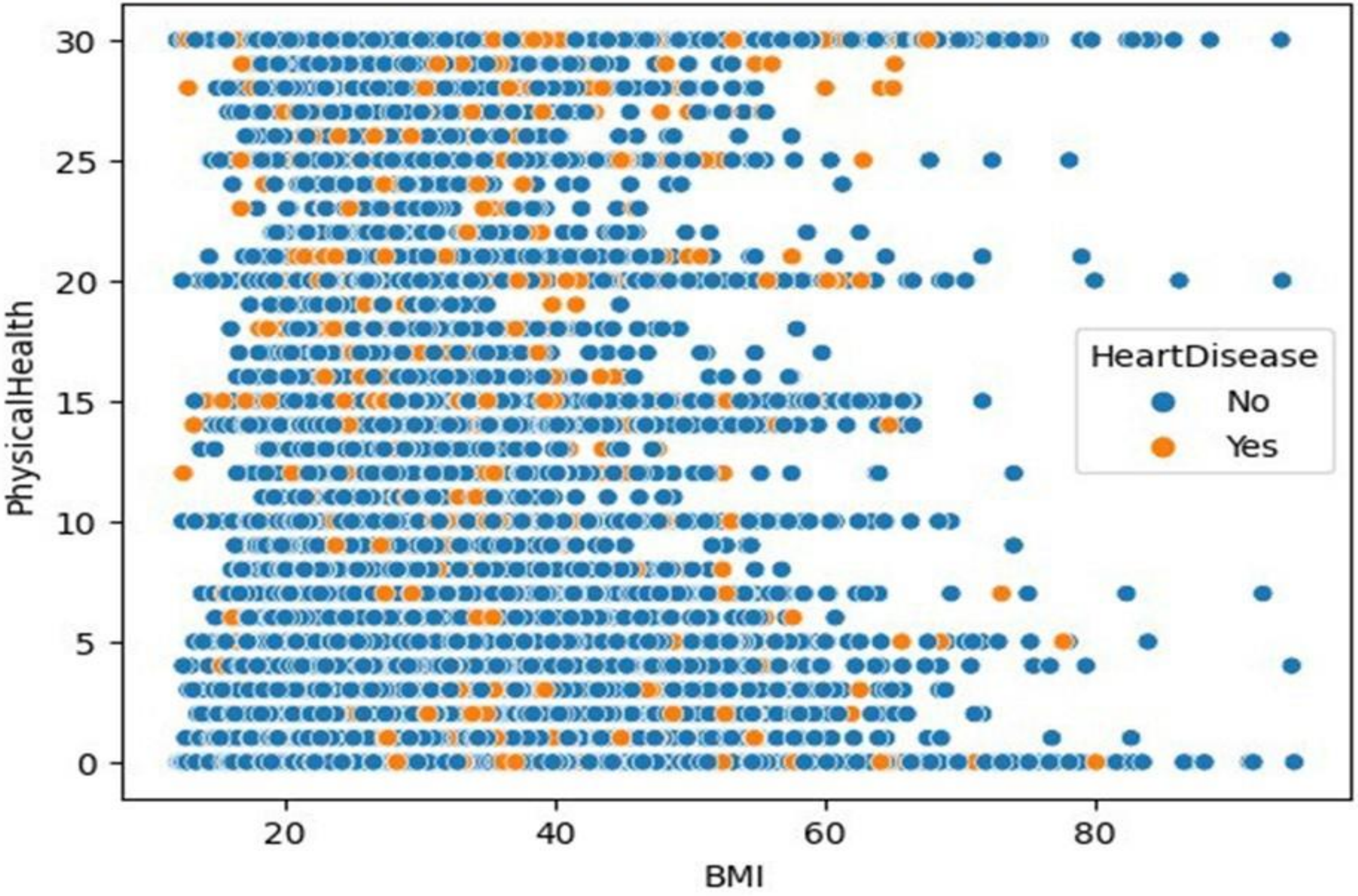
percentage of men and women with heart disease

Problems and Challenges

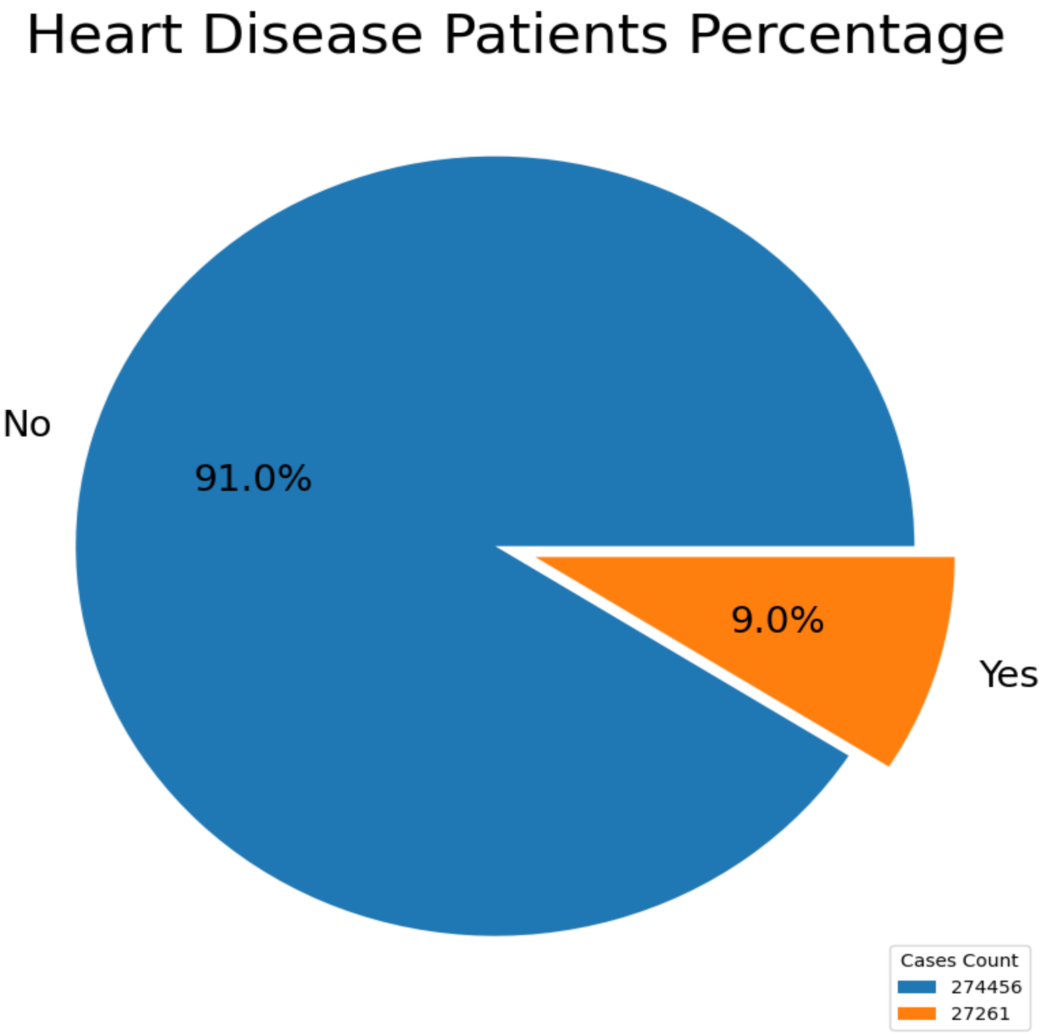
Outliers


80
60
40
20
BMI
30
25
20
15
10
5
0
PhysicalHealth
30
25
20
15
10
5
0
MentalHealth
25
20
15
10
5
0
SleepTime

Outliers


PhysicalHealth
0
5
10
15
20
25
30
BMI
20
40
60
80
HeartDisease
No
Yes

Imbalancing Problem

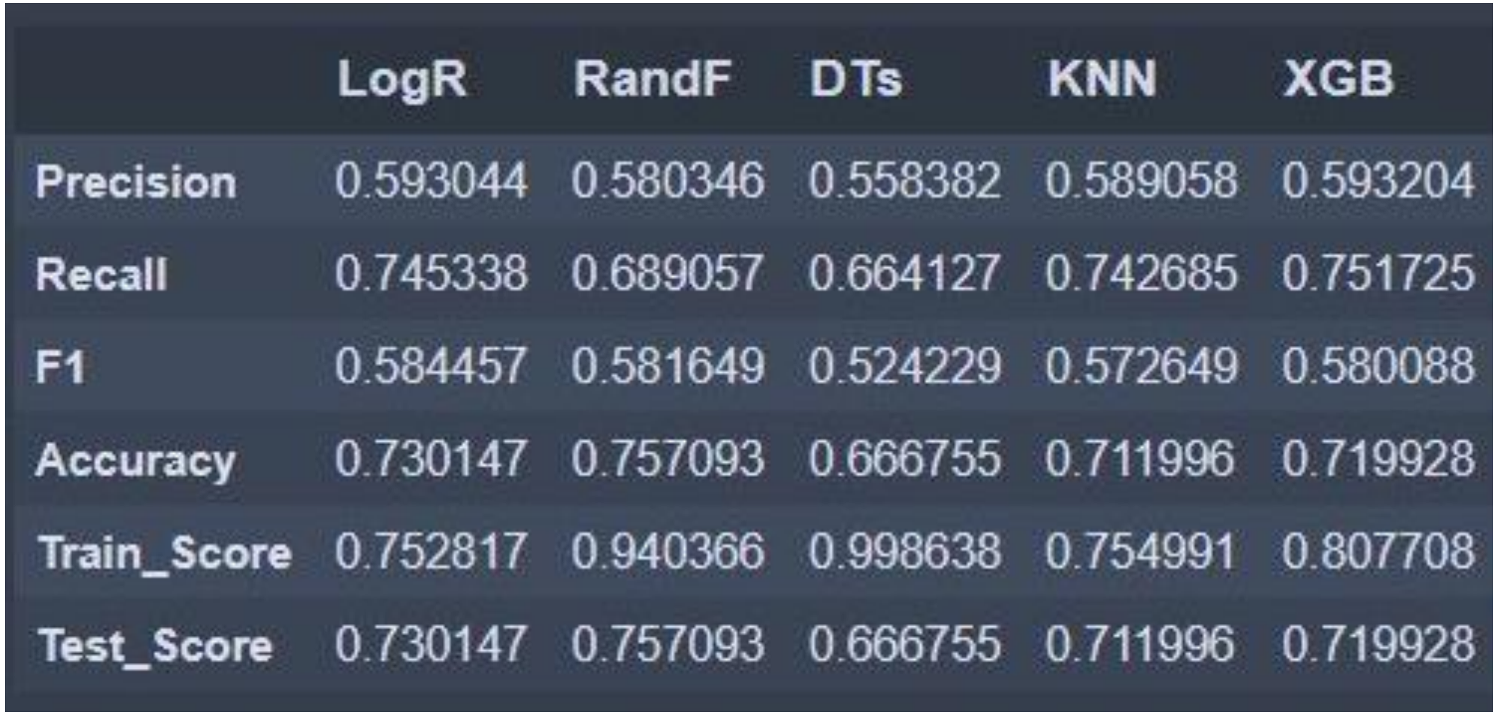


Modeling and Comparing Results

We used the following models:

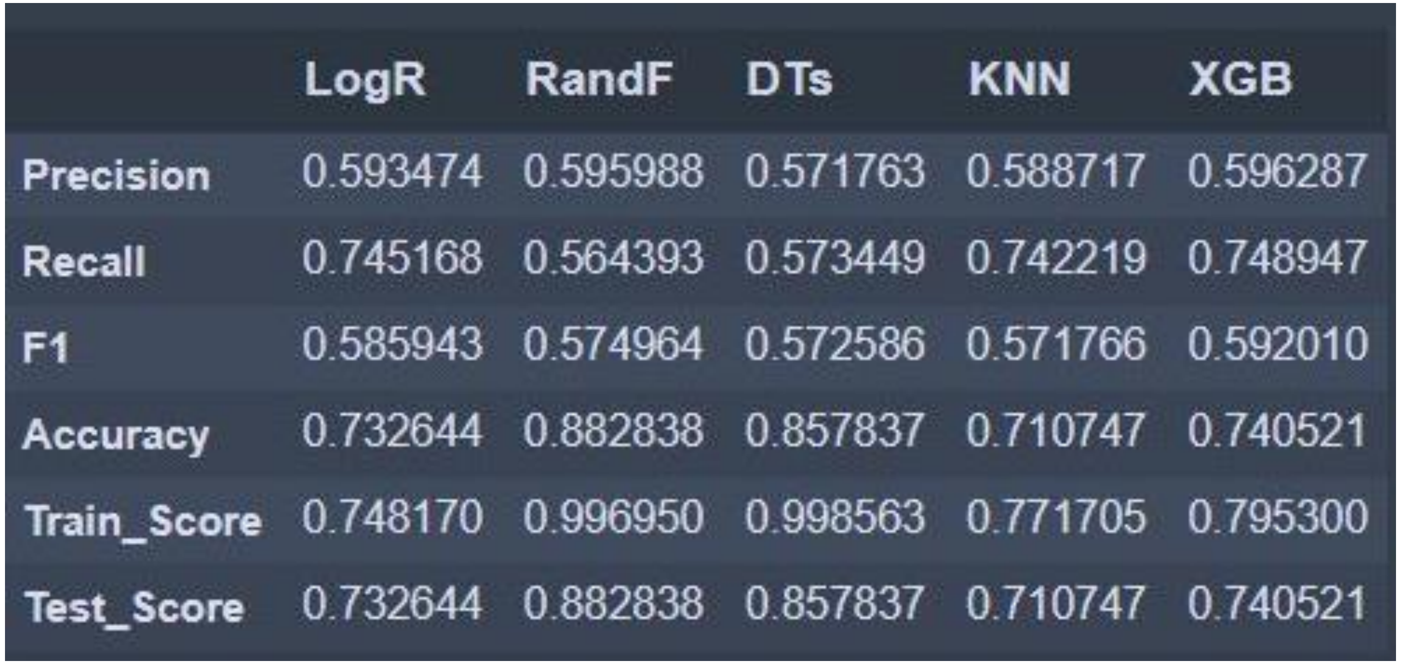
1. Logistic Regression
2. Random Forrest
3. Decision Trees
4. K-Nearest Neighbors
5. XGBoost Classifier

Under-Sampling Randomly

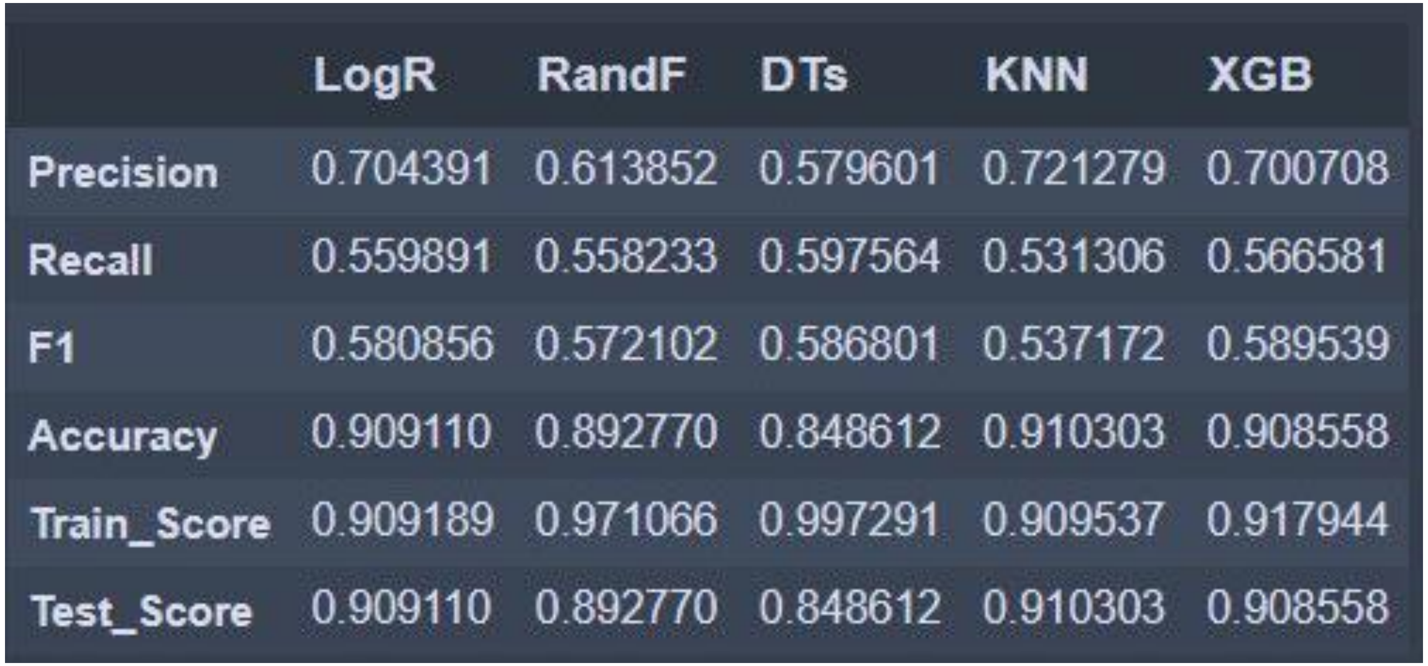


	LogR	RandF	DTs	KNN	XGB
Precision	0.593044	0.580346	0.558382	0.589058	0.593204
Recall	0.745338	0.689057	0.664127	0.742685	0.751725
F1	0.584457	0.581649	0.524229	0.572649	0.580088
Accuracy	0.730147	0.757093	0.666755	0.711996	0.719928
Train_Score	0.752817	0.940366	0.998638	0.754991	0.807708
Test_Score	0.730147	0.757093	0.666755	0.711996	0.719928

Over-Sampling Randomly



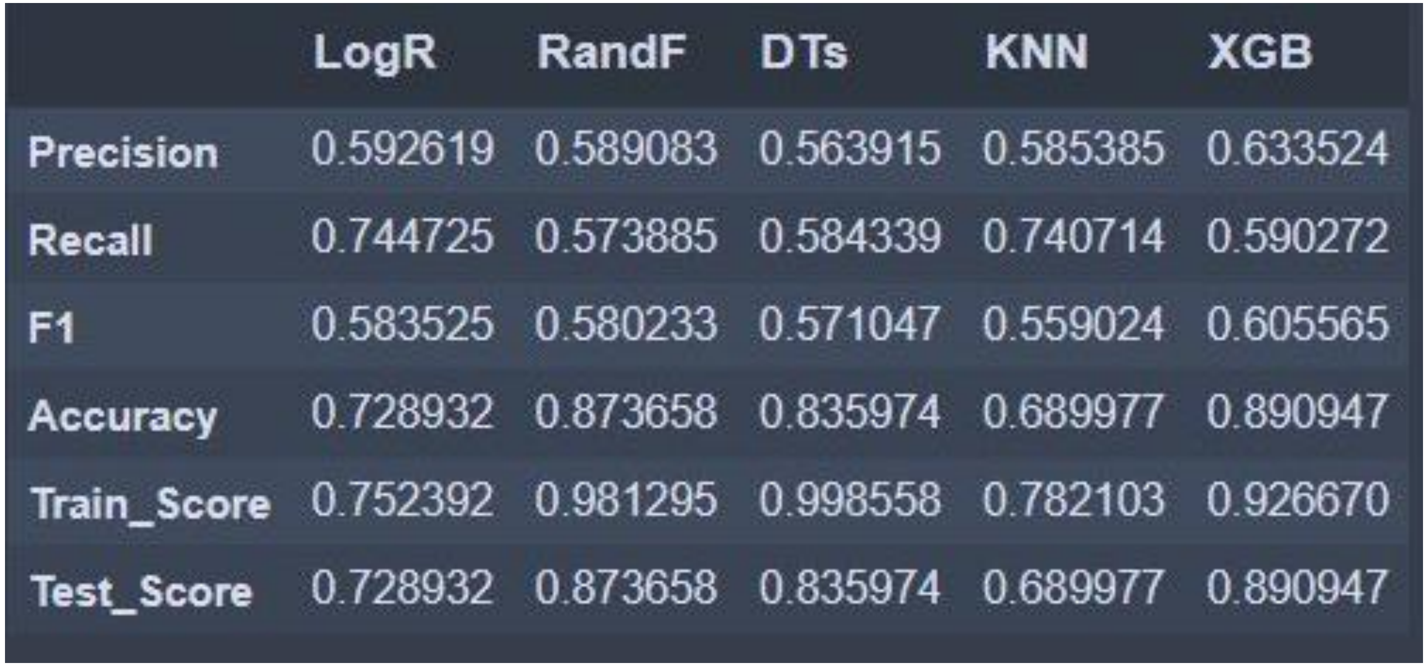
	LogR	RandF	DTs	KNN	XGB
Precision	0.593474	0.595988	0.571763	0.588717	0.596287
Recall	0.745168	0.564393	0.573449	0.742219	0.748947
F1	0.585943	0.574964	0.572586	0.571766	0.592010
Accuracy	0.732644	0.882838	0.857837	0.710747	0.740521
Train_Score	0.748170	0.996950	0.998563	0.771705	0.795300
Test_Score	0.732644	0.882838	0.857837	0.710747	0.740521



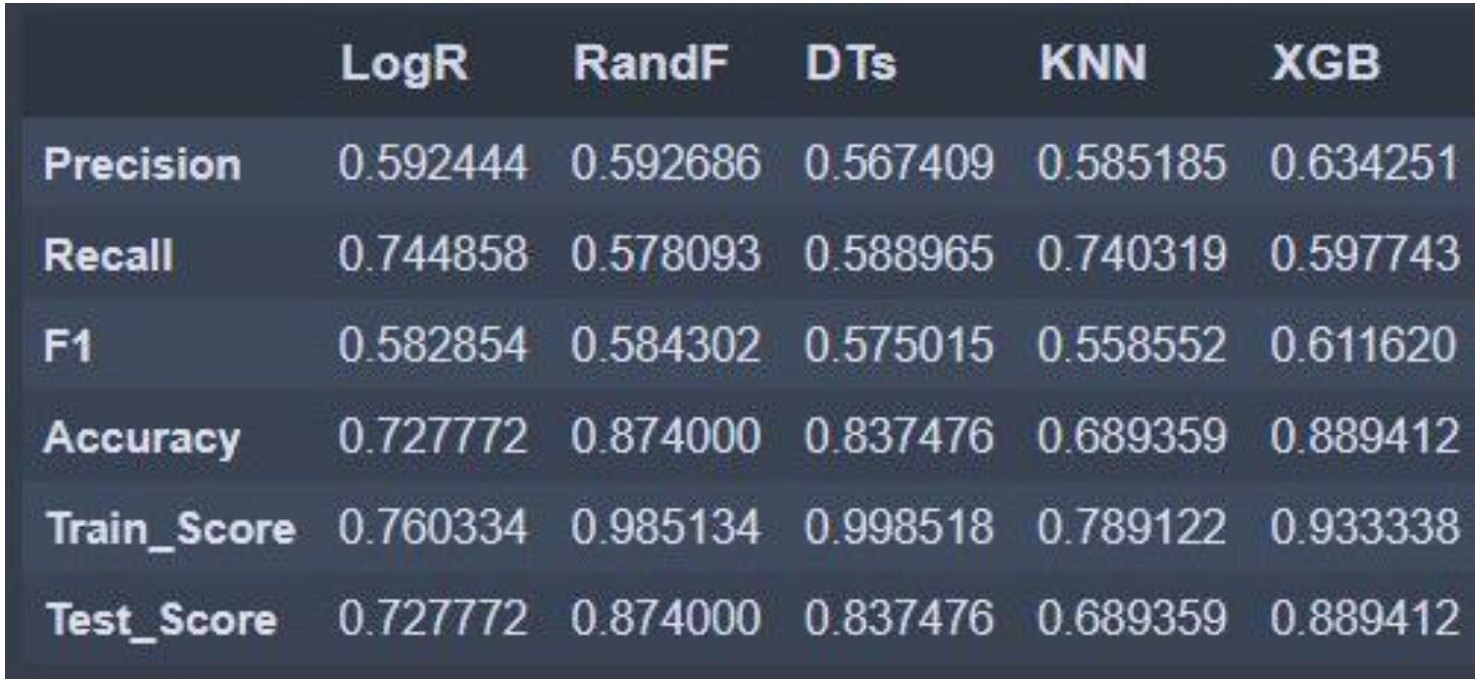
	LogR	RandF	DTs	KNN	XGB
Precision	0.704391	0.613852	0.579601	0.721279	0.700708
Recall	0.559891	0.558233	0.597564	0.531306	0.566581
F1	0.580856	0.572102	0.586801	0.537172	0.589539
Accuracy	0.909110	0.892770	0.848612	0.910303	0.908558
Train_Score	0.909189	0.971066	0.997291	0.909537	0.917944
Test_Score	0.909110	0.892770	0.848612	0.910303	0.908558

Under-Sampling using Tomeklinks

Over-Sampling using SMOTE



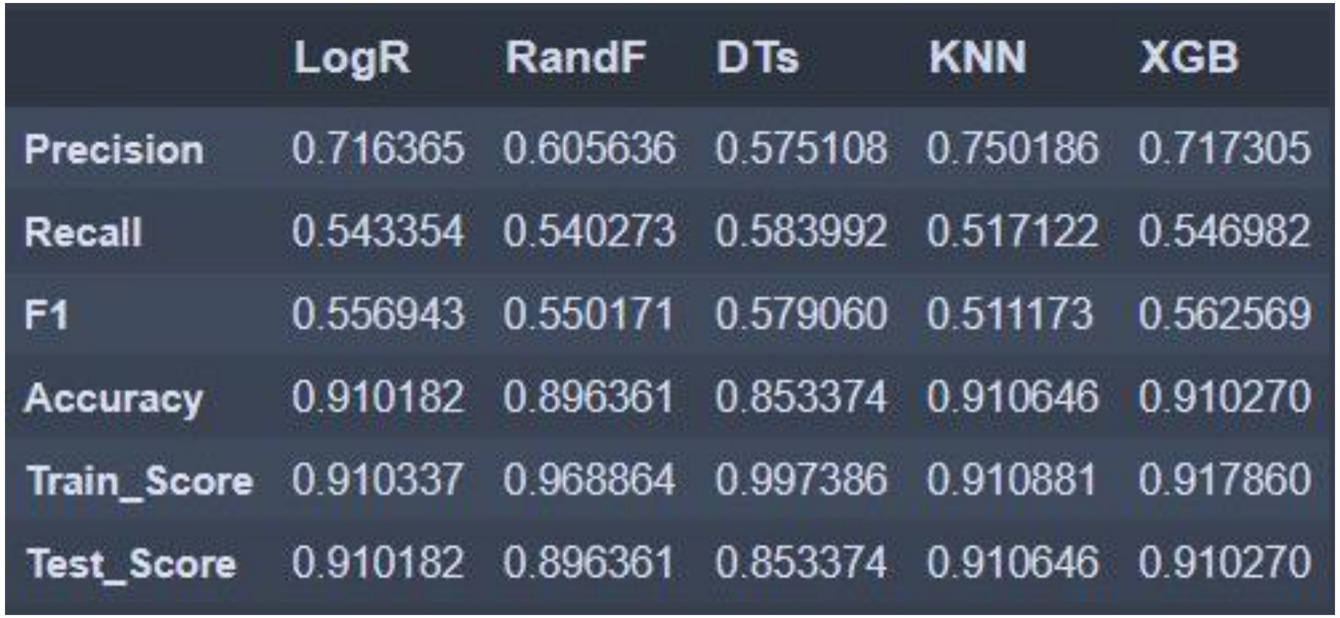
	LogR	RandF	DTs	KNN	XGB
Precision	0.592619	0.589083	0.563915	0.585385	0.633524
Recall	0.744725	0.573885	0.584339	0.740714	0.590272
F1	0.583525	0.580233	0.571047	0.559024	0.605565
Accuracy	0.728932	0.873658	0.835974	0.689977	0.890947
Train_Score	0.752392	0.981295	0.998558	0.782103	0.926670
Test_Score	0.728932	0.873658	0.835974	0.689977	0.890947



	LogR	RandF	DTs	KNN	XGB
Precision	0.592444	0.592686	0.567409	0.585185	0.634251
Recall	0.744858	0.578093	0.588965	0.740319	0.597743
F1	0.582854	0.584302	0.575015	0.558552	0.611620
Accuracy	0.727772	0.874000	0.837476	0.689359	0.889412
Train_Score	0.760334	0.985134	0.998518	0.789122	0.933338
Test_Score	0.727772	0.874000	0.837476	0.689359	0.889412

Combine both Over-Sampling and Under-Sampling Using SMOTETOMEK

Without any Resampling



	LogR	RandF	DTs	KNN	XGB
Precision	0.716365	0.605636	0.575108	0.750186	0.717305
Recall	0.543354	0.540273	0.583992	0.517122	0.546982
F1	0.556943	0.550171	0.579060	0.511173	0.562569
Accuracy	0.910182	0.896361	0.853374	0.910646	0.910270
Train_Score	0.910337	0.968864	0.997386	0.910881	0.917860
Test_Score	0.910182	0.896361	0.853374	0.910646	0.910270

Conclusion.

- We conducted that for our case, The SMOTE technique, and the combination technique (SMOTETETOMEK) were the best approaches.

- We found that both XGB and Logistic Regression had the best results, regarding both the accuracy and the recall (with about 90% accuracy and 75% recall).

- Our original data without any resampling, had the highest and most misleading accuracy as expected.

Thanks