“**Imbalanced Machine Learning to Predict Heart Disease”**

***\*\*Md Saiful Islam Saif\*\* Department of Mathematics and Statistics, Texas tech university, USA\*\****

**01.Abstract:**

Machine learning in classification has become popular in predicting any disease. However, Imbalanced situation in target variable can lead to poor prediction in machine learning. Also, statistical significance can have misleading conclusion due to imbalanced. Our goal is to solve imbalanced problem by using SMOTE technique, which improves accuracy in our model prediction. In addition, we can control threshold probability to improve accuracy in logistic regression. Specifically, we are dealing with heart disease, where we have a very small number of people who has heart disease. We found SMOTE technique is improving machine learning model to predict risk of heart disease.

**02. Data and Variables**

We have selected our dataset from Demographic Health Survey (DHS) data;

(<https://dhsprogram.com/data/dataset/India_Standard-DHS_2020.cfm?flag=0>)

*Dependent Variable*:

SM627E-Do you currently have Any heart disease? (0-No, 1-Yes)

# Independent Variable:

# MV012-Current age

# MV190-Wealth index combined (1-Poorest,2-Poorer, 3-Middle,4-Richer,5-Richest)

# MV463A-Smokes cigarettes (0-No, 1-Yes)

# SM619-Do you drink alcohol (0-No,1-Yes)

# MV025-Type of place of residence (1-Urban, 2-Rural)

# SM627B-Do you currently have hypertension (0-No, 1-Yes)

# MV106-Educational level (0-No education, 1-Primary, 2-Secondary,3-Higher)

# SM630D-Frequency of eating fruits (0-Never, 1-Daily, 2-Weekly, 3-Occassionally)

# SM630E-Frequency of eating eggs (0-Never, 1-Daily, 2-Weekly, 3-Occassionally)

**03. Machine Learning**:

Our objective in this research is to predict heart disease by using machine learning algorithm. In classification, we focused on five things to compare our models such as:

1.F1 Score. 2. Precision. 3. Recall 4. Kappa 5. AUC

**04**. **Imbalanced Technique**:

Imbalance data is a case where the classification dataset class has a skewed proportion. An imbalance class creates a bias where the machine learning model tends to predict the majority class. There are various hacks or techniques to handle an imbalanced class data sample prior to modeling including:

1) Oversampling Minority Class

2) Downsampling Majority Class

3) Generate Synthetic Data

4) Balanced Class Weight

5) Combine Oversampling & Downsampling Techniques

In our research, we will focus on oversample the data by creating synthetic data using the SMOTE technique. There are few variations of SMOTE, including: SMOTE, SMOTE-NC, Borderline-SMOTE, SVM-SMOTE, ADASYN.

we need to use SMOTE-NC when we have cases of mixed data. In our model, we have mostly categorical data along with a quantitative variable. The premise is simple, we denote which features are categorical, and SMOTE would resample the categorical data instead of creating synthetic data.

**05**. **Machine Learning Accuracy Before & After SMOTE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithms Before SMOTE** | **Accuracy** | **Recall** | **Precision** | **F1** | **Kappa** | **AUC** |
| Logistic Regression | 0.9924 | **0.000** | **0.0000** | 0.0000 | 0.0000 | 0.7467 |
| K Neighbors Classifier | 0.9924 | **0.0000** | **0.0000** | 0.0000 | 0.0000 | 0.5118 |
| Ridge Classifier | 0.9924 | **0.0000** | **0.0000** | 0.0000 | 0.0000 |  |
| Ada Boost Classifier | 0.9924 | **0.0000** | **0.0000** | 0.0000 | 0.0000 | 0.7416 |
| Gradient Boosting Classifier | 0.9924 | **0.0000** | **0.0000** | 0.0000 | -0.0001 | 0.7398 |
| Dummy Classifier | 0.9924 | **0.0000** | **0.0000** | 0.0000 | 0.0000 | 0.5000 |
| Light Gradient Boosting Machine | 0.9923 | **0.0000** | **0.0000** | 0.0000 | -0.0002 | 0.7120 |
| Extreme Gradient Boosting | 0.9922 | **0.0000** | **0.0000** | 0.0000 | -0.0004 | 0.6683 |
| SVM - Linear Kernel | 0.9916 | **0.0132** | **0.0097** | 0.0112 | 0.0105 | 0.0000 |
| Random Forest Classifier | 0.9912 | **0.0038** | **0.0211** | 0.0064 | 0.0042 | 0.6073 |
| Extra Trees Classifier | 0.99 | **0.0112** | **0.0373** | 0.0172 | 0.0134 | 0.5684 |
| Decision Tree Classifier | 0.98 | **0.0205** | **0.0343** | 0.0256 | 0.0202 | 0.5038 |
| Linear Discriminant Analysis | 0.9664 | **0.2675** | **0.0675** | 0.1078 | 0.0969 | 0.7447 |
| Naive Bayes | 0.9537 | **0.0512** | **0.0870** | 0.0870 | 0.1064 | 0.7282 |
| Quadratic Discriminant Analysis | 0.4933 | **0.5089** | **0.5248** | 0.0145 | -0.0004 | 0.5089 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithms After SMOTE** | **Accuracy** | **Recall** | **Precision** | **F1** | **Kappa** | **AUC** |
| Random Forest Classifier | 0.9596 | **0.9895** | **0.9338** | 0.9608 | 0.9193 | 0.9875 |
| Extra Trees Classifier | 0.9590 | **0.9891** | **0.9329** | 0.9602 | 0.9180 | 0.9864 |
| Decision Tree Classifier | 0.9570 | **0.9895** | **0.9292** | 0.9584 | 0.9141 | 0.9790 |
| CatBoost Classifier | 0.9399 | **0.9765** | **0.9099** | 0.9421 | 0.8799 | 0.9799 |
| K Neighbors Classifier | 0.9354 | **0.9793** | **0.9003** | 0.9382 | 0.8709 | 0.9699 |
| Extreme Gradient Boosting | 0.9341 | **0.9700** | **0.9050** | 0.9364 | 0.8682 | 0.9769 |
| Light Gradient Boosting Machine | 0.9050 | **0.9447** | **0.8752** | 0.9086 | 0.8100 | 0.9598 |
| Gradient Boosting Classifier | 0.8553 | **0.8832** | **0.8365** | 0.8592 | 0.7105 | 0.9235 |
| Logistic Regression | 0.8309 | **0.8259** | **0.8342** | 0.8300 | 0.6618 | 0.9060 |
| Linear Discriminant Analysis | 0.8302 | **0.8170** | **0.8391** | 0.8279 | 0.6604 | 0.9036 |
| Ridge Classifier | 0.8301 | **0.8169** | **0.8391** | 0.8279 | 0.6603 | 0.0000 |
| Ada Boost Classfier | 0.8247 | **0.8122** | **0.8331** | 0.8225 | 0.6495 | 0.9042 |
| Quadratic Discriminant Analysis | 0.8116 | **0.8155** | **0.8092** | 0.8123 | 0.6232 | 0.8817 |
| SVM - Linear Kernel | 0.8044 | **0.7621** | **0.8442** | 0.7864 | 0.6089 | 0.0000 |
| Naive Bayes | 0.7188 | **0.8168** | **0.6830** | 0.7439 | 0.4377 | 0.8052 |
| Dummy Classifier | 0.5000 | **0.000** | **0.0000** | 0.0000 | 0.0000 | 0.5000 |