Harnessing Machine Learning to Detect

Cardiovascular Disease:

A Comparative Study

Done by:

Karunya R V
125018039
CSBS

TABLE OF CONTENTS

ABSTRACT

Cardiovascular disease (CVD), commonly known as heart disease, is a leading global health concern, accounting for approximately 31% of all deaths worldwide. It encompasses a wide range of conditions that affect the heart and blood vessels, driven by risk factors such as smoking, high blood pressure, elevated cholesterol levels, obesity, diabetes, and a family history of heart-related issues. Symptoms often include chest pain, shortness of breath, fatigue, dizziness, and nausea. Effective management of CVD relies on lifestyle changes, medications, and, in severe cases, surgical interventions. Early detection of CVD plays a crucial role in reducing mortality and improving patient outcomes.

This project aims to harness the power of machine learning techniques to predict cardiovascular disease based on patient data, facilitating early identification of at-risk individuals. By analysing key risk factors and utilizing ML algorithms, this study seeks to enhance both the accuracy and interpretability of predictive models. The focus will be on developing robust models that can identify subtle patterns in patient data, which may be indicative of future cardiovascular events. The ultimate goal is to contribute to more effective prevention and treatment strategies by providing a data-driven approach to early diagnosis. Improved detection of CVD through machine learning could empower healthcare providers with actionable insights, thereby aiding in timely clinical decision-making and reducing the overall burden of cardiovascular disease globally.

INTRODUCTION

In this project, we aim to predict the presence of **Heart Disease** based on a comprehensive set of health and demographic attributes. The dataset comprises features that include health behaviours, pre-existing medical conditions, and demographic factors, all of which play a critical role in determining cardiovascular risk.

About the Dataset

This dataset consists of 319795 entries and 18 columns (17 features and 1 target). The datatype of the columns is a mix of numerical and categorical.

The dataset has the following features : HeartDisease, BMI, Smoking, AlcoholDrinking, Stroke, PhysicalHealth, MentalHealth, DiffWalking, Sex, AgeCategory, Race, Diabetic, PhysicalActivity, GenHealth, SleepTime, Asthma, KidneyDisease, SkinCancer.

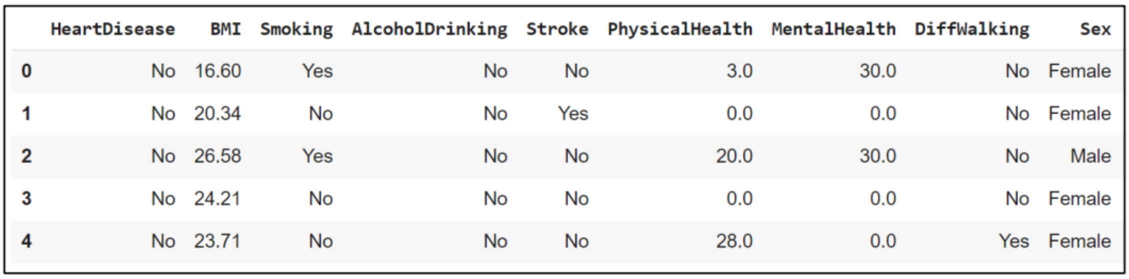
The target variable in this dataset is *HeartDisease*, which represents whether an individual has been diagnosed with cardiovascular disease. It is a binary classification where ‘yes’ indicates the presence of heart disease, and ‘no’ indicates its absence.

Dataset Source

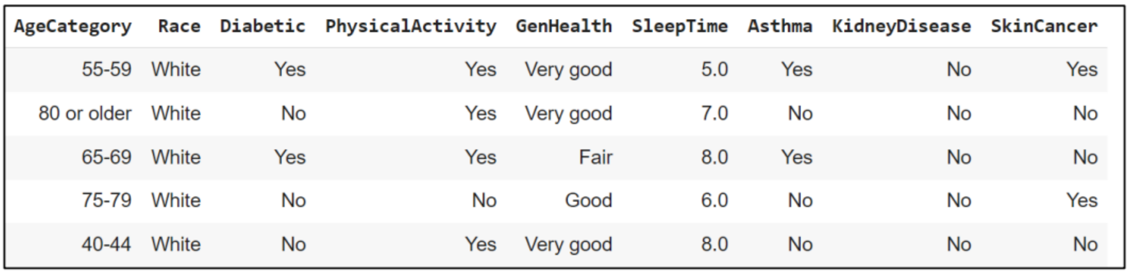
The dataset is obtained from the online Kaggle data repository.

Link : *https://www.kaggle.com/datasets/luyezhang/heart-2020-cleaned*

Dataset Description

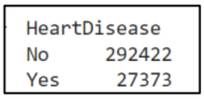


	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex
0	No	16.60	Yes	No	No	3.0	30.0	No	Female
1	No	20.34	No	No	Yes	0.0	0.0	No	Female
2	No	26.58	Yes	No	No	20.0	30.0	No	Male
3	No	24.21	No	No	No	0.0	0.0	No	Female
4	No	23.71	No	No	No	28.0	0.0	Yes	Female

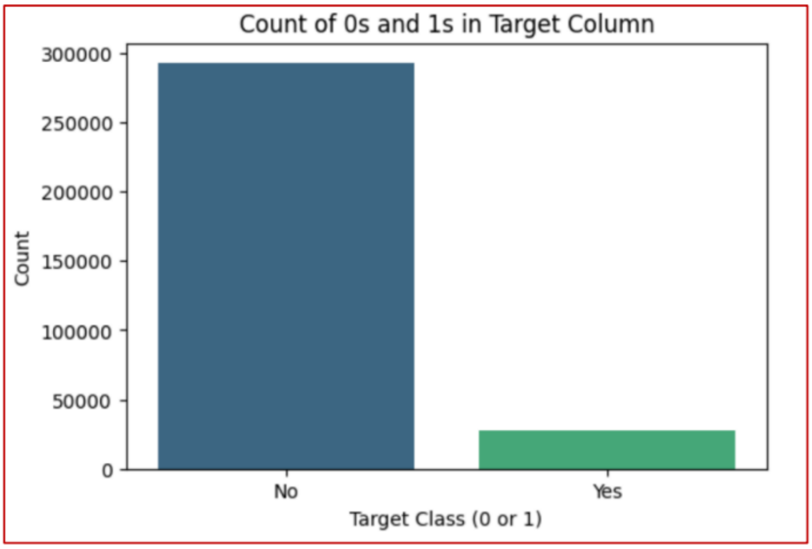


AgeCategory	Race	Diabetic	PhysicalActivity	GenHealth	SleepTime	Asthma	KidneyDisease	SkinCancer
55-59	White	Yes	Yes	Very good	5.0	Yes	No	Yes
80 or older	White	No	Yes	Very good	7.0	No	No	No
65-69	White	Yes	Yes	Fair	8.0	Yes	No	No
75-79	White	No	No	Good	6.0	No	No	Yes
40-44	White	No	Yes	Very good	8.0	No	No	No

Target Column Description



```
HeartDisease
No      292422
Yes      27373
```



What we are trying to accomplish (<T,P,E> format)

Task: The task here is to detect Cardiovascular Disease using machine learning techniques. This involves building a predictive model that can classify whether an individual is at risk of heart disease based on various features such as age, cholesterol levels, blood pressure, and other health metrics.

- *Input Data*: Patient data, which may include numerical and categorical features like age, gender, cholesterol, blood pressure, etc.
- *Output*: A binary classification (1 = Disease, 0 = No disease).

Performance Metrics: The metrics for this task include accuracy, precision, recall and F1-score. The F1-score balances precision and recall, making it useful when false positives and false negatives are of equal concern.

Experience: Applying machine learning to cardiovascular disease detection involves iterating through different models like Logistic Regression, Random Forest, and XGBoost, while tuning hyperparameters to improve performance. Handling data imbalance plays a key role in optimizing model performance. Experimenting with different algorithms and tuning them provided better predictive power in our models.

Results

The average accuracy across models was around 91% before balancing the dataset. However, after balancing the accuracy dropped significantly for most models, averaging 81.4%.

RELATED WORK

Dataset Link : https://www.kaggle.com/datasets/luyezhang/heart-2020-cleaned

References :

- ChatGPT
- https://www.kaggle.com/code/youssefislamelrefaie/heart-disease-prediction-ann

MODELS USED

1. Logistic Regression

Logistic Regression is a supervised learning algorithm used for **binary classification** tasks, though it can be extended to multiclass problems. Unlike linear regression, which predicts continuous values, logistic regression predicts the **probability** that a given input belongs to a particular class.

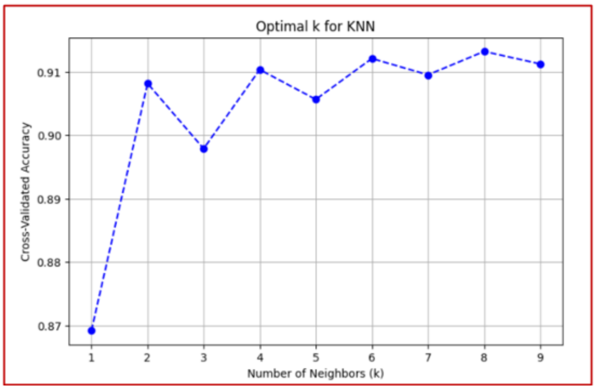
Principle : Uses a linear combination of input features, passes it through a sigmoid function to output a value between 0 and 1.

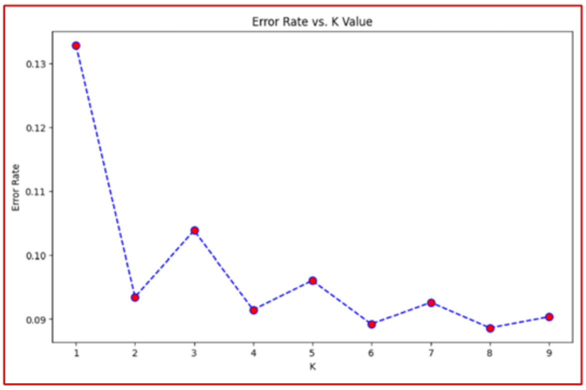
Parameter Tuning :
In this project, the **L2 penalty**, also known as Ridge regularization is applied to prevent overfitting. This helps to distribute weights evenly across all features. it also ensures that no feature is entirely excluded, as the weights are shrunk toward zero but never exactly zero. Hence, the model generalizes better to unseen data.

2. KNN

K-Nearest Neighbours (KNN) is a supervised learning algorithm that classifies a data point based on the majority class of its **k closest neighbours** in the feature space. It works by computing the **distance** between points and assigning the label of the most common class among the nearest neighbours.

Principle : Data points that are closer to each other are similar and belong to the same class.





From the graph of **Accuracy vs. K Value**, the optimal k value for KNN is the value of k where the cross-validated accuracy is the highest. In the graph, the peak accuracy is around **k = 6 or k = 8** where the accuracy is about 0.91.

From the graph of **Error Rate vs. K Value**, the goal is to find the k value that minimizes the error rate. It is observed that the error rate decreases when **k = 6** and **k = 8**.

Since both k values of 6 and 8 give similar results, the smaller value of **k = 6** is chosen to avoid overfitting.

Distance metric used : Euclidean distance

3. SVM

A support vector machine (SVM) is a supervised machine learning algorithm that classifies data by finding an optimal **hyperplane** that maximizes the distance between each class in the data.

Principle : Finds the optimal line that maximizes the margin between the closest data points of opposite classes.

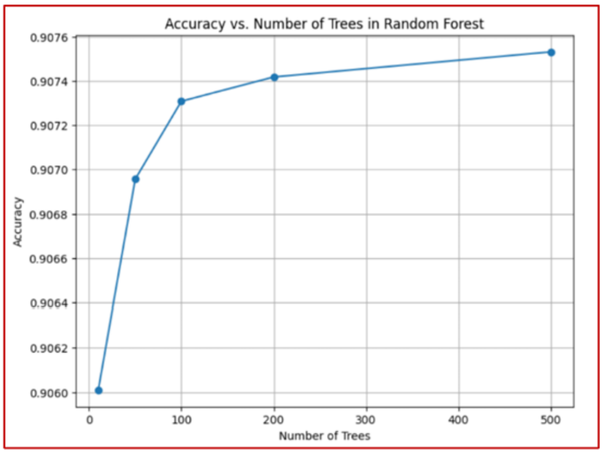
Parameters used :

- Kernel – ‘linear’
- C (Regularization parameter) – ‘1.0’
- Gamma – ‘scale’

4. Random Forest

Random Forest is an **ensemble learning** method that builds multiple decision trees during training and merges their predictions to improve accuracy and prevent overfitting.

Principle : Creates multiple decision trees on different subsets of the data and combines their outputs to reach a single result.

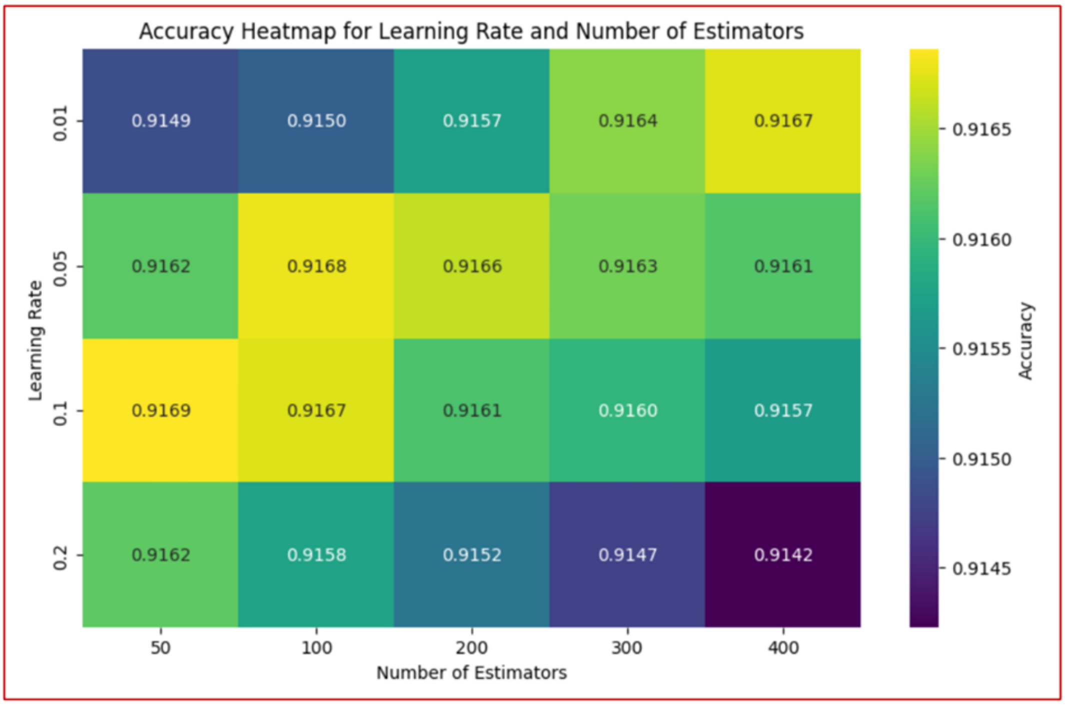


This is a plot of the model’s performance against the number of trees. This plot helps visualize how the model’s accuracy changes as more trees are added.

From this plot, it is seen that the accuracy increases significantly up to around **100 trees**, after which the improvement becomes marginal. The curve starts to plateau after 100 trees, due to which adding more trees beyond this point doesn’t lead to significant accuracy improvement.

Hence, the number of trees chosen is 100.

5. XGBoost



This heatmap helps visualize the interaction between the learning rate and the number of estimators (trees) in terms of model accuracy. From the plot, it can be observed that the accuracy generally improves as the number of estimators increases up to a certain point. The number of estimators around 100 to 200 provides good results across multiple learning rates. With a learning rate of 0.05 and 100 estimators, the accuracy is **0.9168**, one of the best values on the map.

Hence, the values of the parameters are chosen based on this heatmap.

6. ANN

An Artificial Neural Network (ANN) is a computational model inspired by the structure and function of the human brain, consisting of interconnected layers of nodes (neurons). ANNs are used for tasks such as classification, regression, and pattern recognition, leveraging their ability to learn from data through backpropagation and adjust weights to minimize errors.

Parameters used :

Total number of layers used in ANN in this project is 9.

i. 5 Dense layers (including the output layer)
ii. 4 Batch Normalization layers
iii. 4 Dropout layers

Layers and Architecture:

- 1st Dense layer: 256 units, ReLU activation
- 2nd Dense layer: 128 units, ReLU activation
- 3rd Dense layer: 64 units, ReLU activation

- 4th Dense layer: 32 units, ReLU activation
- Output layer: 1 unit, Sigmoid activation (for binary classification)

Additional Layers:

- **BatchNormalization**: Applied after each dense layer to normalize activations.
- **Dropout**: 0.5 dropout rate after each dense layer to reduce overfitting.

Optimizer:

- **Adam optimizer** with a learning rate of 0.001.

Loss Function:

- **Binary Crossentropy**: Used since this is a binary classification problem.

Metrics:

- **Accuracy**: Used to evaluate model performance.

Training Parameters:

- **EarlyStopping**: Monitors validation loss (val_loss), with patience of 5 epochs and restores best weights.
- **Epochs**: 15.
- **Batch size**: 64
- **Validation split**: 20% of the training data is used for validation.

PREPROCESSING TECHNIQUES

1) **Missing Values**

   The dataset has no null values.

2) **Feature Encoding**

   14 out of 18 attributes in the dataset are categorical. So, **label encoding** has been performed for those specific features, thereby converting them into numerical values without increasing the total no. of columns.

3) **Scaling**

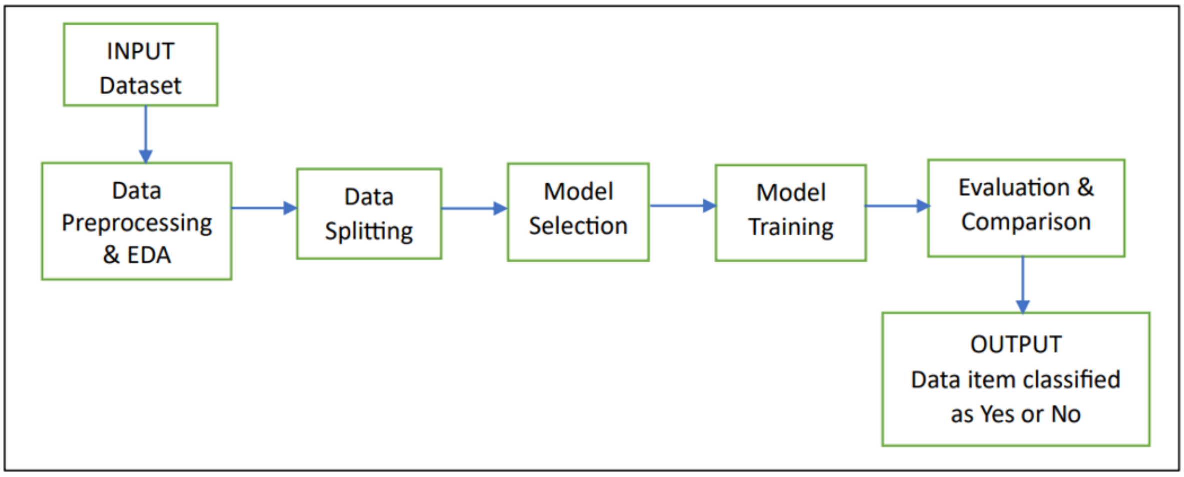
   Standard scaling is performed to normalize feature values so they have a mean of 0 and a standard deviation of 1, ensuring that all features contribute equally to the model and improving the performance of algorithms sensitive to scale, like KNN and SVM.

4) **Balancing**

   SMOTE (**Synthetic Minority Over-sampling Technique**) is the method used in this project to handle class imbalance. It works by selecting data points from the minority class and creating new **synthetic points** along the line segments between these selected points and their nearest neighbours. This approach helps generate new, similar instances for the minority class to balance the dataset.

METHODOLOGY

Overall Workflow



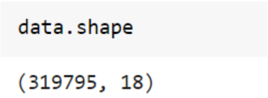
Tools used

- Google Colab
- MS Word

Libraries used

- pandas
- sklearn
- seaborn
- imblearn
- matplotlib
- xgboost
- tensorflow

Dataset Description



```
data.shape
```

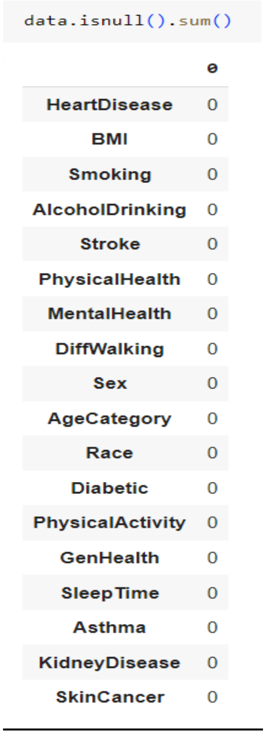
```
(319795, 18)
```

The dataset has a total of 3,19,795 rows and 18 columns.

Out of the 18 columns, 14 columns including the Target variable are of type ‘object’ and the remaining 4 are of type ‘float’.

Results of Data Preprocessing

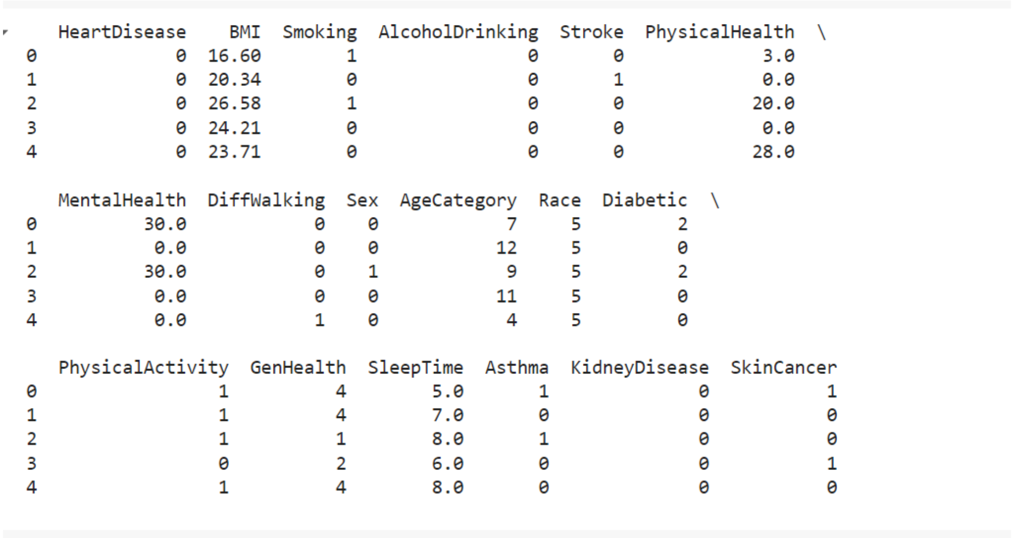
Checking for **NULL** values :



```
data.isnull().sum()
```

	0
HeartDisease	0
BMI	0
Smoking	0
AlcoholDrinking	0
Stroke	0
PhysicalHealth	0
MentalHealth	0
DiffWalking	0
Sex	0
AgeCategory	0
Race	0
Diabetic	0
PhysicalActivity	0
GenHealth	0
SleepTime	0
Asthma	0
KidneyDisease	0
SkinCancer	0

After performing **Label Encoding**, all categorical features were transformed into numeric values, with the no. of columns remaining unchanged**.**

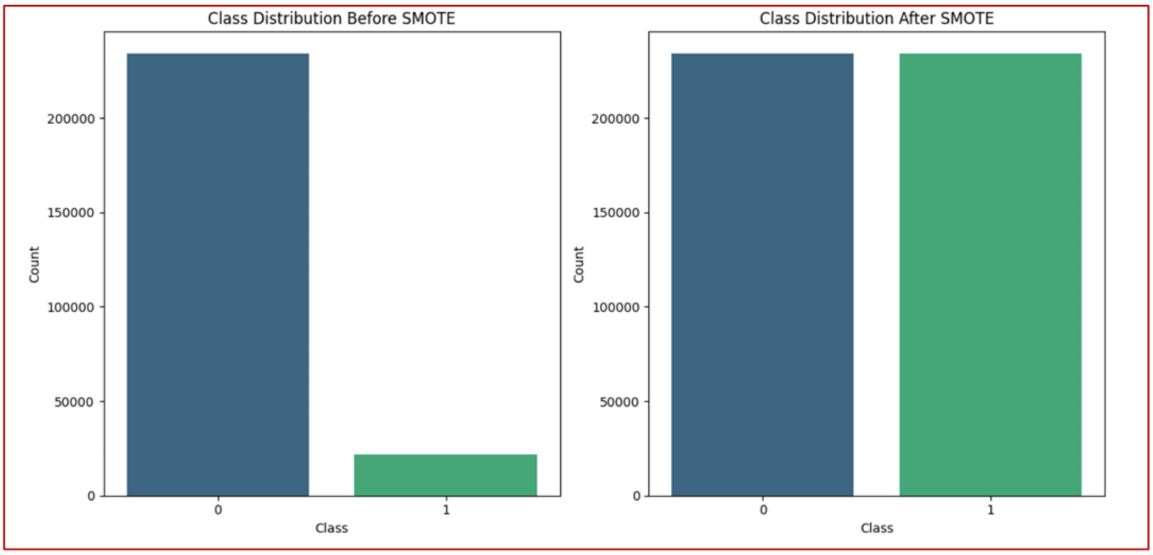


```
   HeartDisease    BMI  Smoking  AlcoholDrinking  Stroke  PhysicalHealth  \
0             0  16.60        1                0       0             3.0   
1             0  20.34        0                0       1             0.0   
2             0  26.58        1                0       0            20.0   
3             0  24.21        0                0       0             0.0   
4             0  23.71        0                0       0            28.0   

   MentalHealth  DiffWalking  Sex  AgeCategory  Race  Diabetic  \
0          30.0            0    0            7     5         2   
1           0.0            0    0           12     5         0   
2          30.0            0    1            9     5         2   
3           0.0            0    0           11     5         0   
4           0.0            1    0            4     5         0   

   PhysicalActivity  GenHealth  SleepTime  Asthma  KidneyDisease  SkinCancer  
0                 1          4        5.0       1              0           1  
1                 1          4        7.0       0              0           0  
2                 1          1        8.0       1              0           0  
3                 0          2        6.0       0              0           1  
4                 1          4        8.0       0              0           0  
```

After performing **SMOTE** on the data, the count of minority class entries also increased, thereby creating a more balanced dataset.



RESULTS

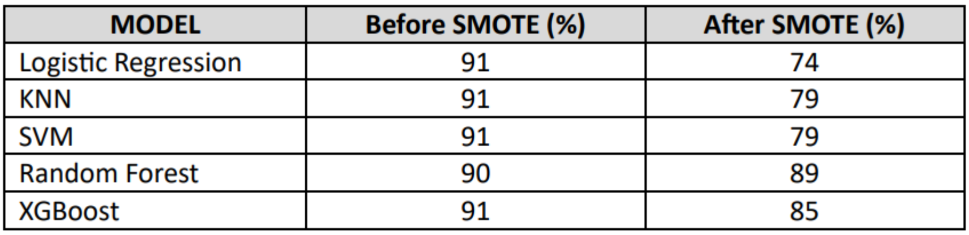
1. Accuracy

MODEL	Before SMOTE (%)	After SMOTE (%)
Logistic Regression	91	74
KNN	91	79
SVM	91	80
Random Forest	90	89
XGBoost	91	85

2. Weighted Average Precision

MODEL	Before SMOTE (%)	After SMOTE (%)
Logistic Regression	88	91
KNN	88	88
SVM	83	85
Random Forest	87	87
XGBoost	89	89

3. Weighted Average Recall



MODEL	Before SMOTE (%)	After SMOTE (%)
Logistic Regression	91	74
KNN	91	79
SVM	91	79
Random Forest	90	89
XGBoost	91	85

4. Results of ANN (in %)

Test Accuracy	Validation Accuracy	Validation Loss
91.38	91.55	22.74

Inference:

- SMOTE negatively impacts accuracy for most models, especially Logistic Regression and KNN.
- Weighted Average Precision improves for Logistic Regression, SVM, and remains stable for others.
- SMOTE generally causes a reduction in Weighted Average Recall for most models, particularly for Logistic Regression and KNN.
- Random Forest and XGBoost demonstrate some robustness, experiencing smaller drops in recall.

LEARNING OUTCOME

Links

Google Colab Page (Code):

https://colab.research.google.com/drive/1lVh7udRQfA875VW1K85bG-UoDfWOTUDp#scrollTo=cZIY5ifZleth

Github Repository :

https://github.com/Karu08/ML-Project/upload

Skills used & Learnings from this project

1. Supervised Learning Algorithms: Good understanding of various machine learning models such as Logistic Regression, K-Nearest Neighbours (KNN), Support Vector Machines (SVM), Random Forest, XGBoost, and ANN for binary classification tasks**.**

2. Model Tuning and Optimization: Skills in hyperparameter tuning, such as choosing optimal values for regularization (L2), number of neighbours (K), kernel functions, regularization parameters (C), number of trees, learning rates, and neural network layer configurations**,** and how adjusting these can significantly affect model performance.

3. Data Preprocessing: Expertise in handling categorical data through label encoding, normalizing features with standard scaling, and addressing class imbalances using SMOTE.

4. Regularization and Generalization: Implemented L2 penalty, BatchNormalization, and Dropout layers to control overfitting, ensuring the models generalize better to unseen data.

5. Evaluation Metrics: Ability to evaluate models using metrics like accuracy, precision, and validation loss, and using techniques like EarlyStopping to prevent overfitting during training.

6. Ensemble Learning: Gained practical experience with ensemble learning methods like Random Forest and XGBoost for improving predictive performance.

7. ANN Design and Training: Skill in designing multi-layer ANNs with ReLU activation for hidden layers, Sigmoid for binary classification, and optimizing with Adam optimizer for model training.

Hence, doing this project demonstrate a strong understanding of different machine learning algorithms, data preprocessing techniques, choosing model parameters and model optimization.

CONCLUSION

(a) Concluding Remarks of the Work:

- Successfully implemented and compared multiple machine learning models for cardiovascular disease detection.
- The project demonstrated that balancing the dataset using SMOTE improves precision but may reduce accuracy for some models.
- The ANN model performed well with optimized layers, showcasing the benefits of deep learning for classification tasks.
- The project highlighted the importance of hyperparameter tuning, as different algorithms required careful adjustment to achieve optimal performance, demonstrating that no single model fits all datasets without customization.

(b) Did You Accomplish the TPE?

- **Task (T)**: Yes, the task of classifying cardiovascular disease was effectively accomplished using multiple supervised learning algorithms and ANN.

- **Performance Metric (P)**: Models were evaluated using accuracy and weighted average precision and recall successfully achieving performance benchmarks, with ANN and Random Forest showing robust results.

- **Experience (E)**: Gained valuable experience in applying machine learning and deep learning techniques, tuning parameters, handling imbalanced datasets, and leveraging regularization for improved model generalization.

(c) Advantages of the project:

- **Comprehensive Model Comparison**: Evaluated and compared the performance of several machine learning models, providing insights into which techniques perform best for cardiovascular disease detection.
- **Data Balancing with SMOTE**: Improved model precision and helped balance the class distribution, addressing the imbalanced dataset issue.
- **Regularization Techniques**: Implemented methods like L2 regularization, BatchNormalization, and Dropout to prevent overfitting and enhance model generalization.
- **Scalability**: The ANN model, due to its complexity, can be further tuned or extended for larger datasets and more complex problems.

(d) Limitations of the project:

- **Accuracy Trade-off**: While SMOTE improved precision, it led to a decrease in accuracy for some models like Logistic Regression, KNN, and SVM.
- **Computation Time**: Some models like Random Forest and XGBoost can be computationally expensive, especially with a large number of trees or estimators.
- **Dependence on Preprocessing**: The performance of certain algorithms (KNN, SVM) was highly dependent on proper scaling and encoding, which requires additional preprocessing steps.
- **Overfitting Risk in Deep Learning**: Despite regularization, deep learning models like ANN can still overfit, especially with limited data, requiring careful tuning of parameters.

FUTURE SCOPE

- ➢ User Interface (UI) Development – using react and Django, or tkinter, or streamlit
- ➢ Incorporation of additional health data, such as genetic data
- ➢ Cross-Disease Prediction

~~ ** ~~