CS 6375 - Machine Learning
Personal Key Indicators of Heart Disease

Shreeprasad Anant Sonar, Abhishek Chauhan, Aniket Kulkarni

May 8, 2023

Abstract

In this report, we present the findings of our machine learning project using the 2020 annual CDC survey data of 400K adults related to their health status. Our goal was to create models that, using a collection of relevant data, can forecast the chance of developing heart disease. The dataset was cleaned, normalized, and the "HeartDisease" variable was changed to a binary class as part of the preprocessing procedure. Next, we used a variety of machine learning techniques to forecast the chance of developing heart disease, such as logistic regression, Naive Bayes, Decision Tree, XGBoost & K Nearest Neighbors.

Project Topic

The goal of the project is to predict heart disease using data from 400,000 US individuals included in the 2020 annual CDC survey. The dataset includes important markers such as high blood pressure, high cholesterol, smoking, diabetes, obesity, physical activity, and alcohol use. Heart disease is a main cause of mortality in the US. The number of variables was reduced to 20 through cleaning and feature selection so that the dataset can be used for exploratory data analysis and machine learning techniques.

Motivation

Heart disease is a major global health problem, and identifying and reducing its risk factors is essential to enhancing medical care. Large datasets like those from the CDC survey may be analyzed using machine learning techniques to get insights into the trends and causes of heart disease. We may be able to prevent deaths and enhance lifestyle standards for people at risk by utilizing this dataset to anticipate heart disease.

About the Data Set

The dataset comes from the 2020 CDC survey on the health status of 400K individuals, which focused on heart disease risk factors such as high blood pressure, high cholesterol, smoking, diabetes status, obesity, physical activity, and alcohol use. The Behavioral Risk Factor Surveillance System (BRFSS), which conducts yearly telephone surveys to collect information on Americans' health state, provided the dataset. The variable "HeartDisease" is treated as binary, and classes are not balanced.

Exploratory Data Analysis

The adult health status dataset from the 2020 CDC survey is imbalanced, which means that the number of respondents with heart disease is much lower than those who do not have heart disease. Although the total dataset is imbalanced, the distribution of the "Sex" variable is rather balanced for the "No" label, with 54% of respondents being female and 46% being male, and for the "Yes" label, there are 41% of female respondents and 59% of male respondents.

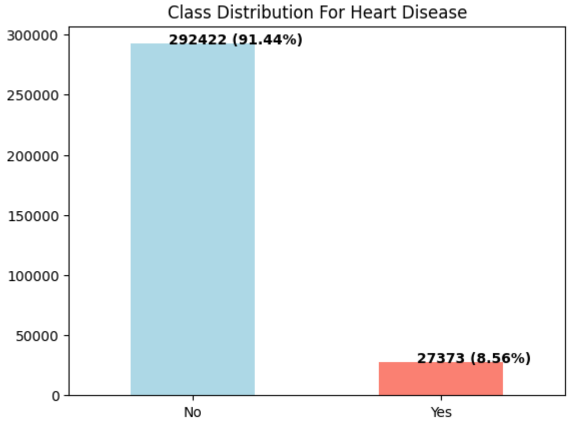

Figure 1: Class Distribution For HeartDisease

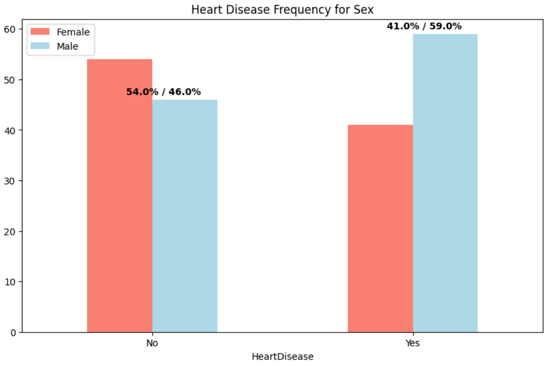

Figure 2: Heart Disease Frequency for Sex

A correlation matrix is plotted to investigate the linear correlations between variables. The correlation matrix is used to determine the strongest correlations between variables, and graphs are then plotted to depict the links between variables with the highest correlation values in the matrix.

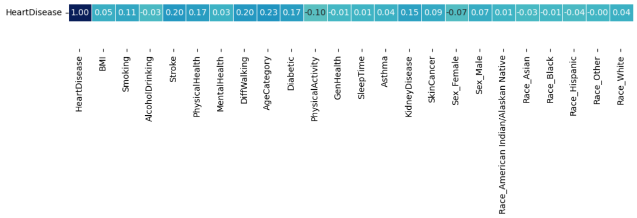

Figure 3: Correlation Matrix for Label and Features

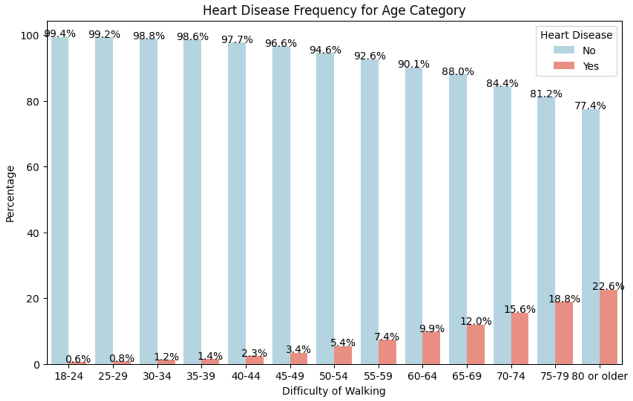

Figure 4: Heart Disease Frequency for AgeCategory

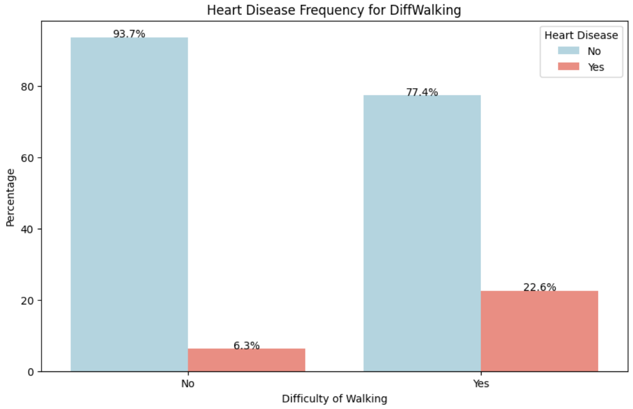

Figure 5: Heart Disease Frequency for DiffWalking

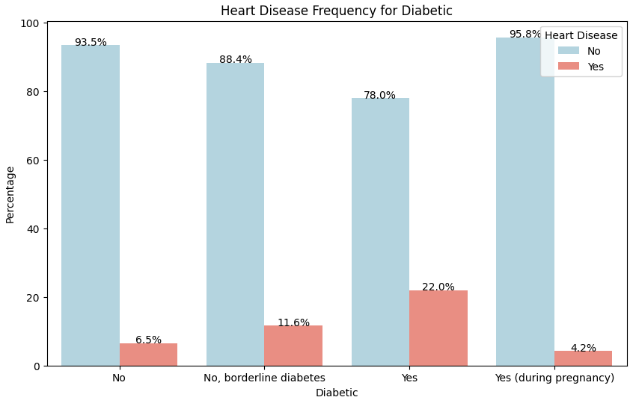

Figure 6: Heart Disease Frequency for Diabetic

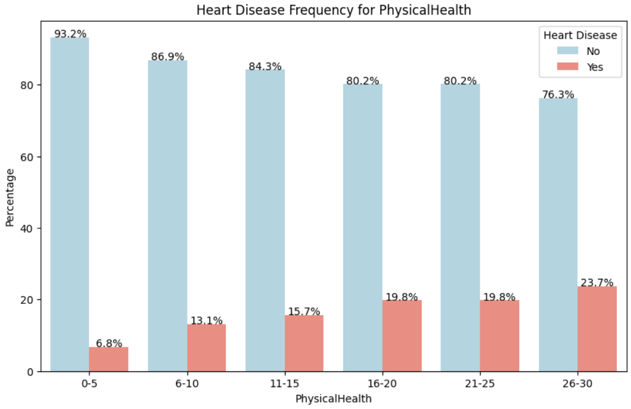

Figure 7: Heart Disease Frequency for PhysicalHealth

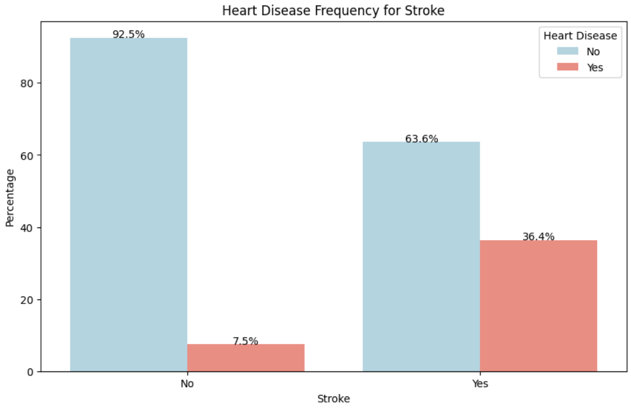

Figure 8: Heart Disease Frequency for Stroke

Methods

We compared custom implementation of the four classifiers i.e. Logistic Regression, Naive Bayes, Decision Tree along with Bagging, AdaBoost and XGBoost with the corresponding Scikit-learn models under 10 Fold cross validation and sampling the data in ratios from 1:1 to 9:1 to analyze the accuracies, precision, recall and F1 scores.

Logistic Regression

Given its widespread use for binary classification issues in machine learning, logistic regression is a strong choice for predicting the risk of heart disease based on a person's key indicators. In medical applications where false positives and false negatives might have major repercussions, it can handle imbalanced data by altering the threshold for predicting the outcome. Additionally, it is interpretable, able to reveal patterns in the relationship among predictors and results. In general, logistic regression is a straightforward yet powerful method that can offer insightful information for anticipating the risk of heart disease based on individual key markers.

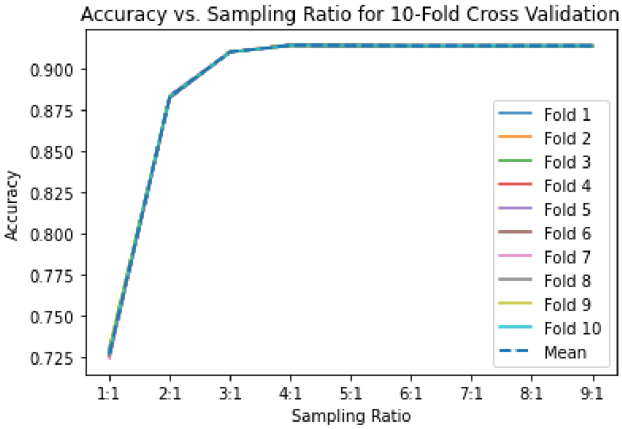

Figure 9: Accuracy on Testing Data using Custom LR

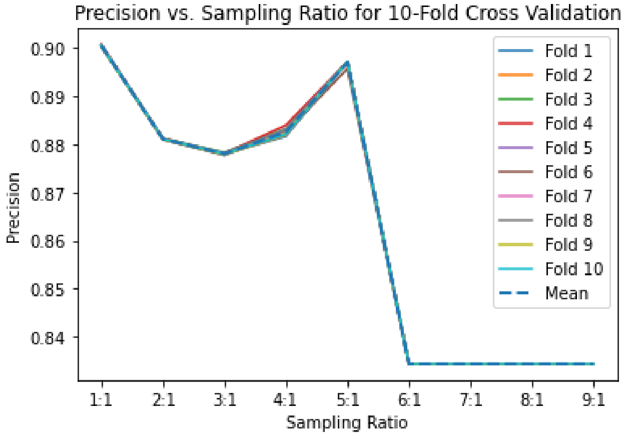

Figure 10: Precision on Testing Data using Custom LR

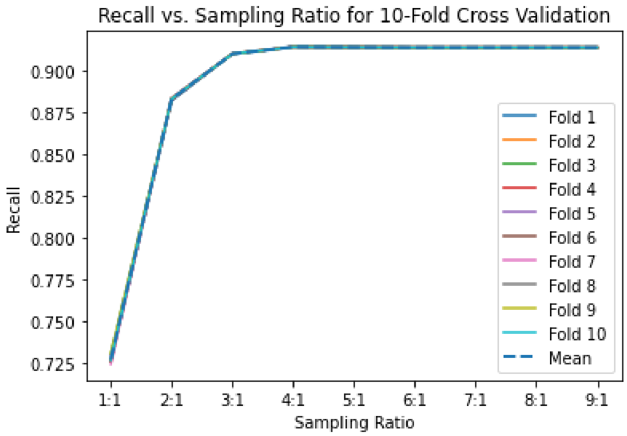

Figure 11: Recall on Testing Data using Custom LR

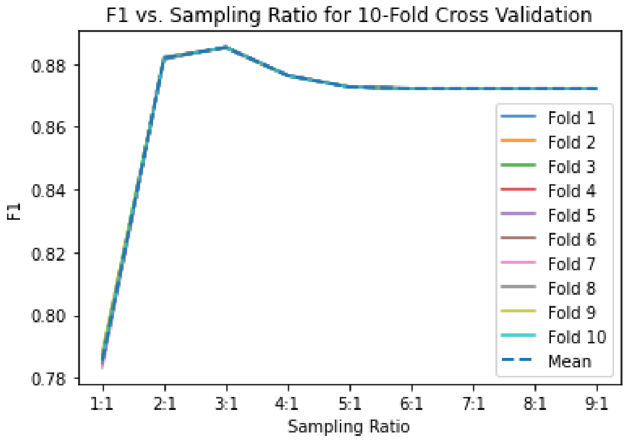

Figure 12: F1 on Testing Data using Custom LR

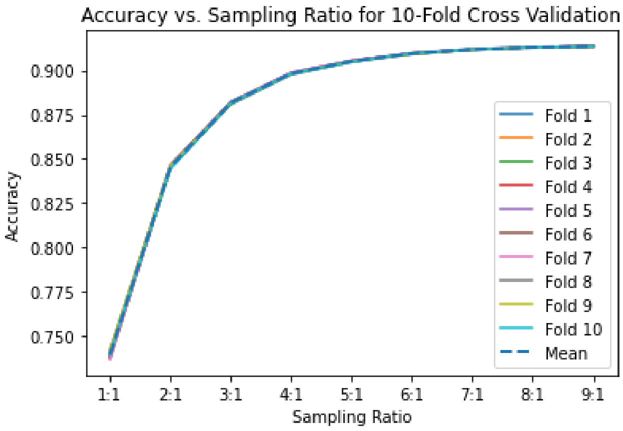

Figure 13: Accuracy on Testing Data using Sklearn LR

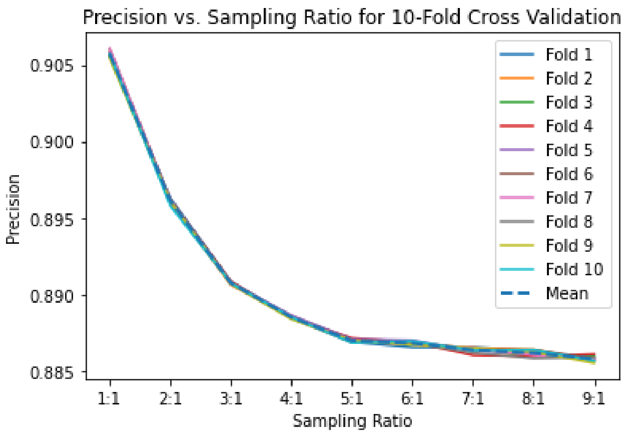

Figure 14: Precision on Testing Data using Sklearn LR

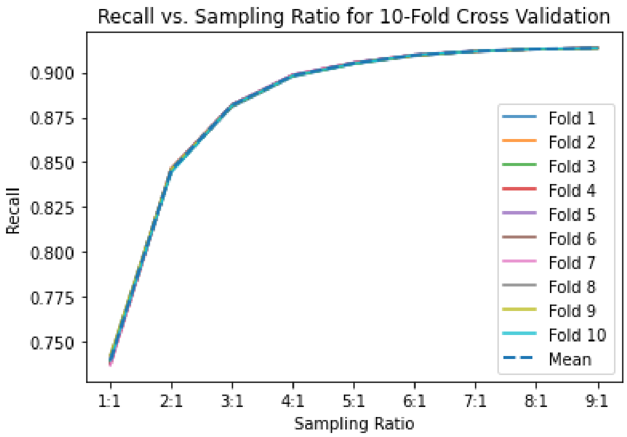

Figure 15: Recall on Testing Data using Sklearn LR

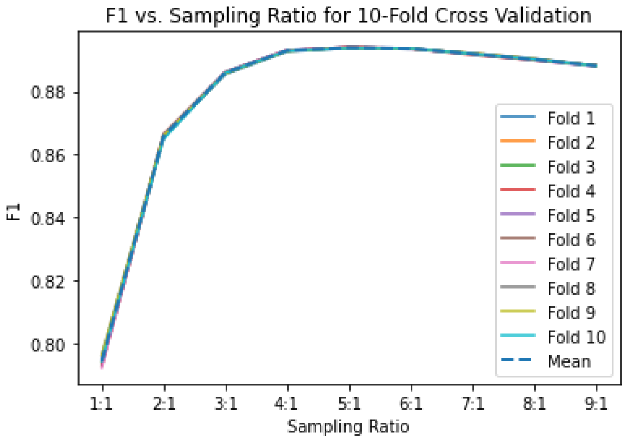

Figure 16: F1 on Testing Data using Sklearn LR

Naive Bayes

An approach that works well with imbalanced datasets is Naive Bayes, which is probabilistic in nature. Naive Bayes can efficiently model the conditional probabilities of the characteristics given the class labels in the situation of personal key indicators of heart disease, when the disease's likelihood is low relative to its likelihood of not occurring. Although this reduction makes computation more efficient and lowers the possibility of overfitting, it makes the assumption that the features are independent, which may not be completely true in practice. Naive Bayes is a suitable option for handling medical data since it can work with high-dimensional data and is comparatively insensitive to irrelevant aspects. Medical data may contain a large number of potential predictors. In conclusion, Naive Bayes can be a valuable method for modeling imbalanced datasets.

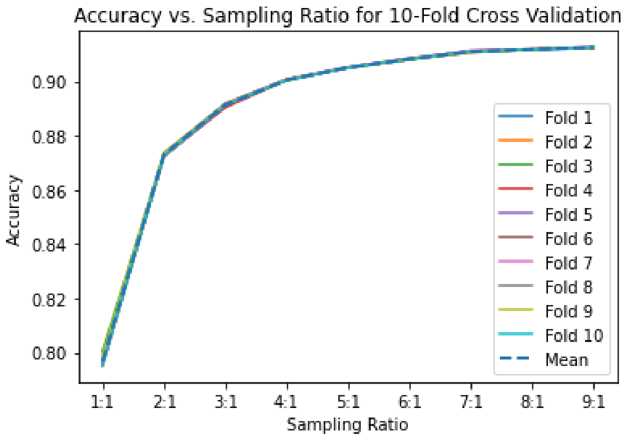

Figure 17: Accuracy on Testing Data using Custom NB

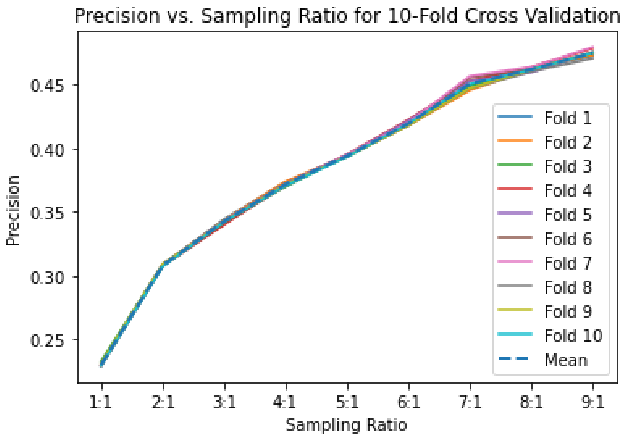

Figure 18: Precision on Testing Data using Custom NB

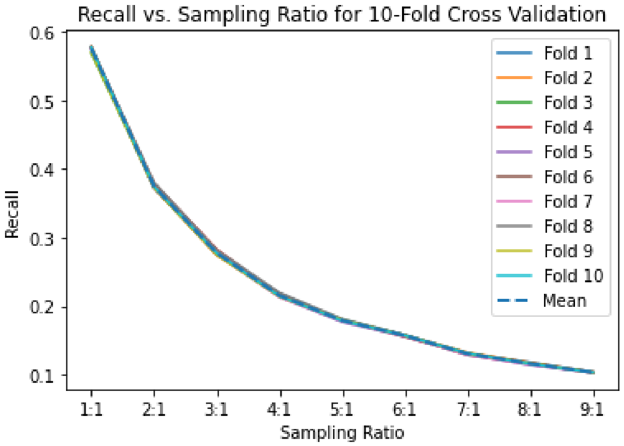

Figure 19: Recall on Testing Data using Custom NB

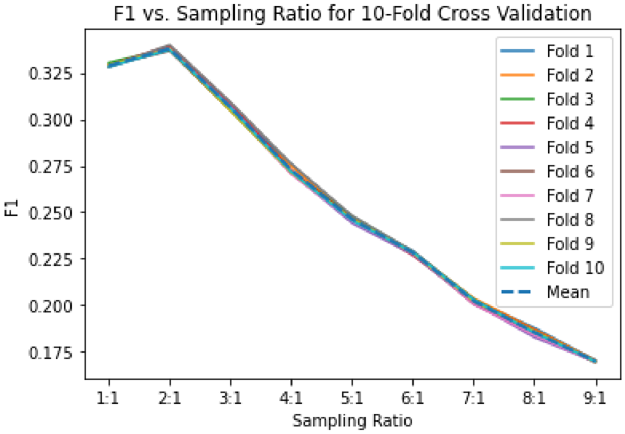

Figure 20: F1 on Testing Data using Custom NB

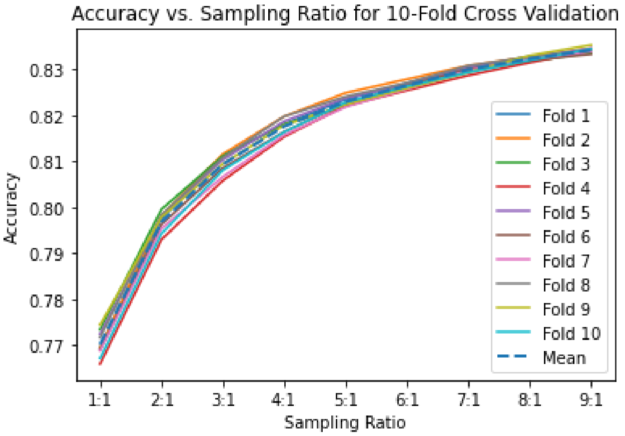

Figure 21: Accuracy on Testing Data using Sklearn NB

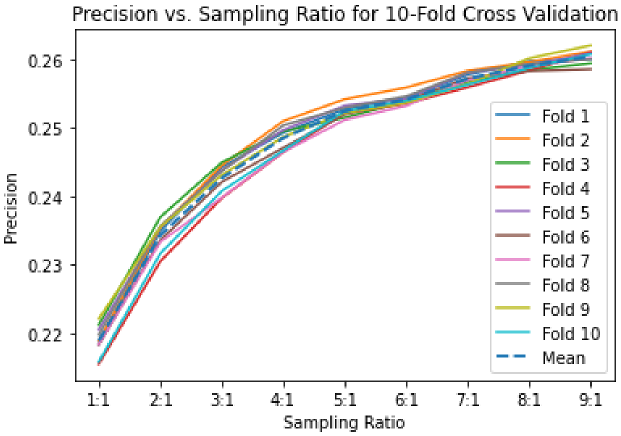

Figure 22: Precision on Testing Data using Sklearn NB

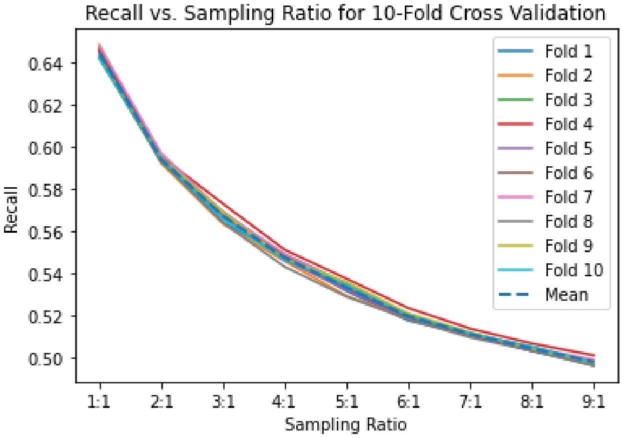

Figure 23: Recall on Testing Data using Sklearn NB

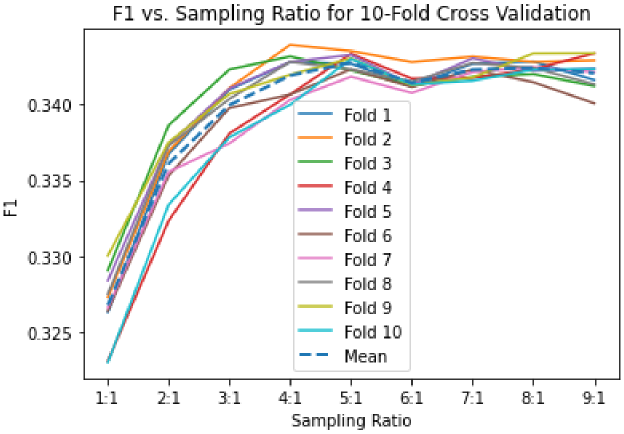

Figure 24: F1 on Testing Data using Sklearn NB

Decision Trees with Bagging and Boosting

When predicting a person's critical heart disease markers from unbalanced data, bagging and boosting on decision trees are useful strategies. When bagging, many decision trees are created using different random subsets of the training data, and the results are then combined to provide a prediction. This method can lessen overfitting and increase the precision of predictions made using unbalanced data. Boosting, on the other hand, entails creating a series of decision trees, each one intended to fix the flaws of the one before it. By giving the minority class greater weight during training, this strategy can be very useful in resolving the issue of class imbalance. At the end of the day, bagging and boosting on decision trees can both be effective strategies for handling unbalanced data which helps predicting personal key indicators of heart disease.

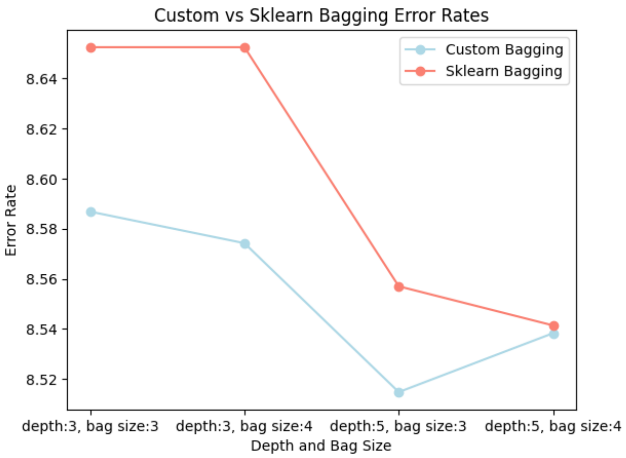

Figure 25: Custom vs Sklearn bagging

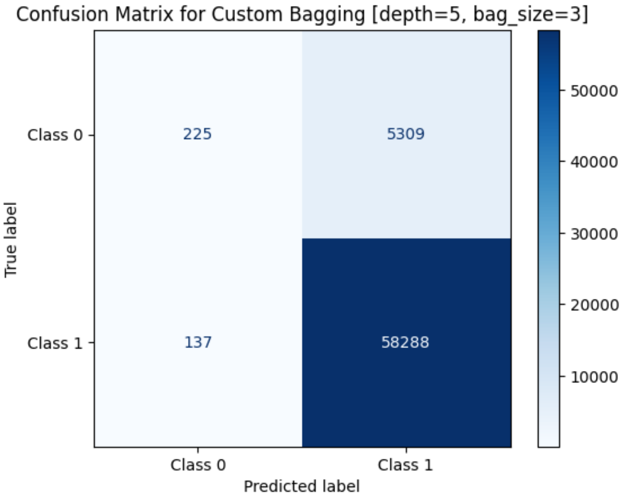

Figure 26: Confusion Matrix for Custom Bagging with optimum parameters

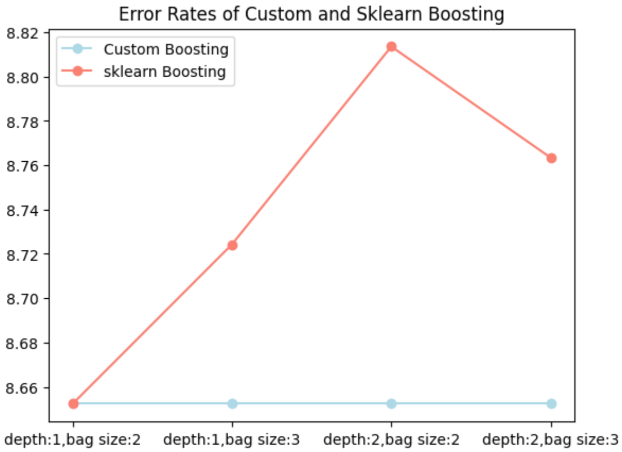

Figure 27: Custom vs Sklearn boosting

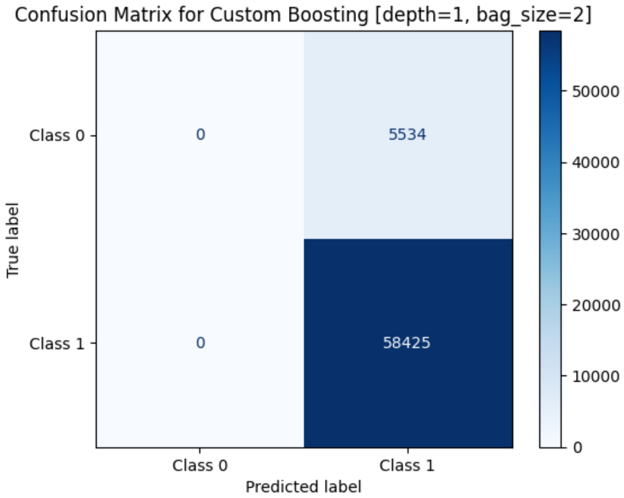

Figure 28: Confusion Matrix for Custom Boosting with optimum parameters

XGBoost

Due to its ability to handle both linear and nonlinear correlations between features and target variables, XGBoost is a popular technique for unbalanced data categorization. It can lower the danger of overfitting and is particularly useful when dealing with high-dimensional data. Additionally, XGBoost can deal with missing data, which is a typical problem in datasets used in medicine.

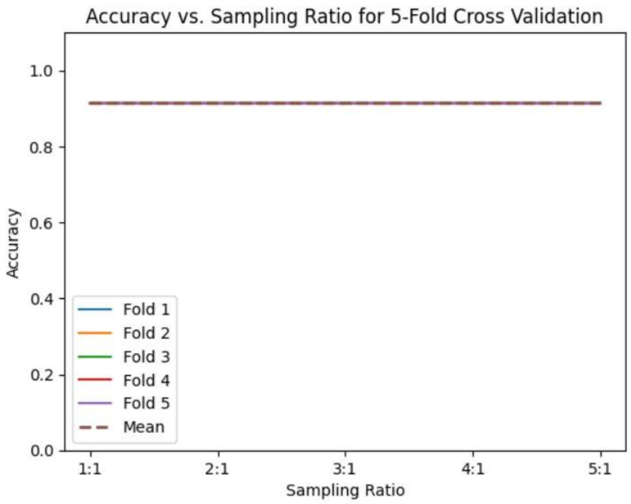

Figure 29: Accuracy on Testing Data using Custom XGB

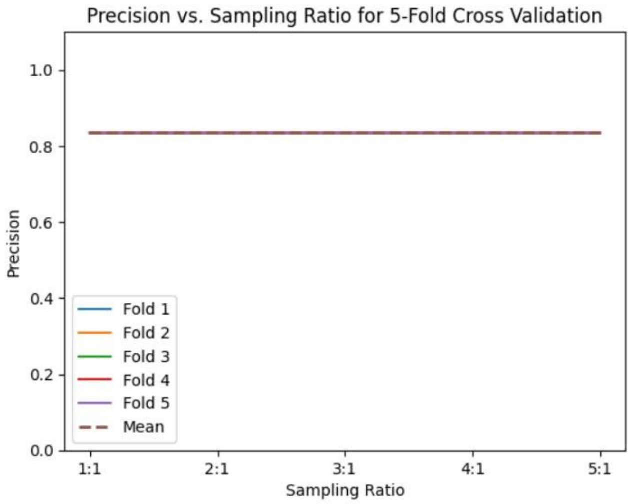

Figure 30: Precision on Testing Data using Custom XGB

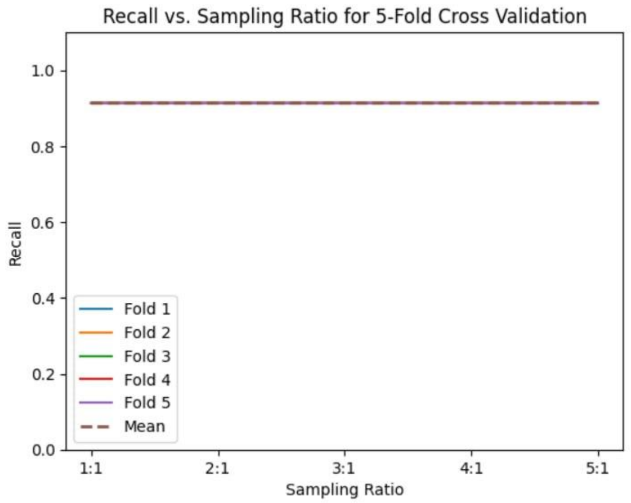

Figure 31: Recall on Testing Data using Custom XGB

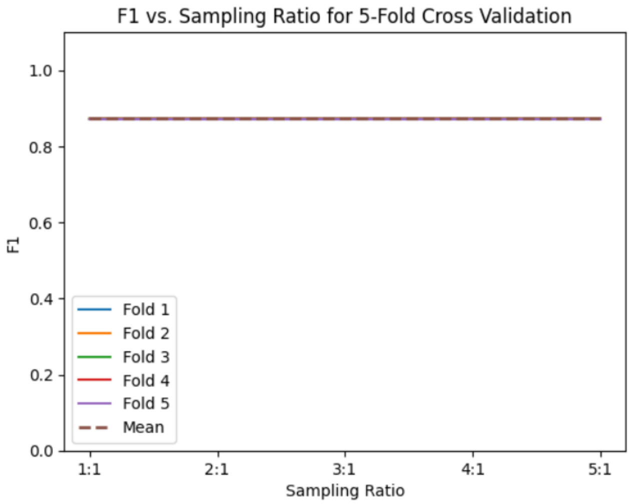

Figure 32: F1 on Testing Data using Custom XGB

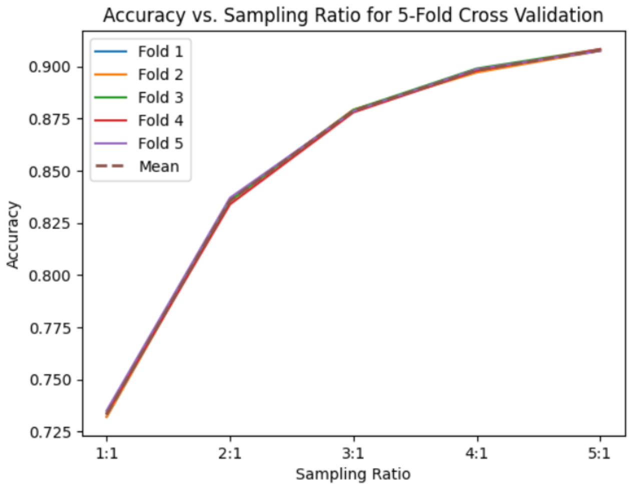

Figure 33: Accuracy on Testing Data using Sklearn XGB

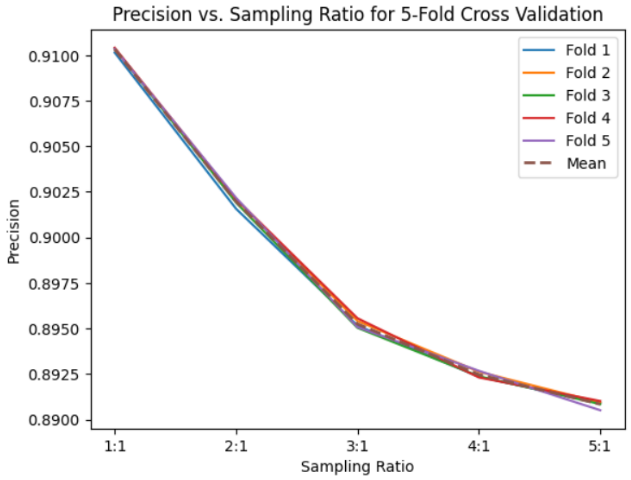

Figure 34: Precision on Testing Data using Sklearn XGB

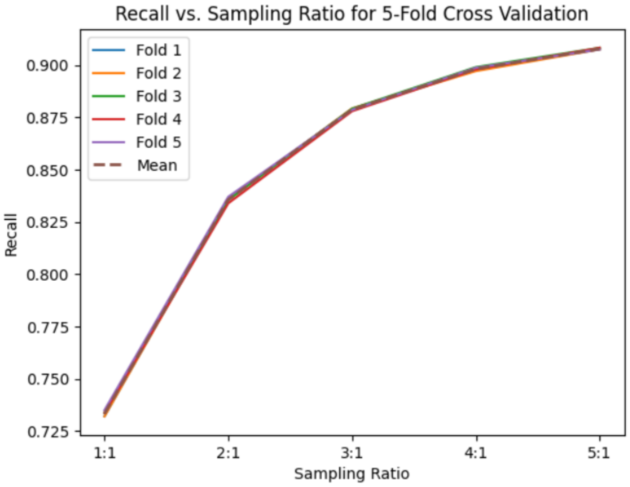

Figure 35: Recall on Testing Data using Sklearn XGB

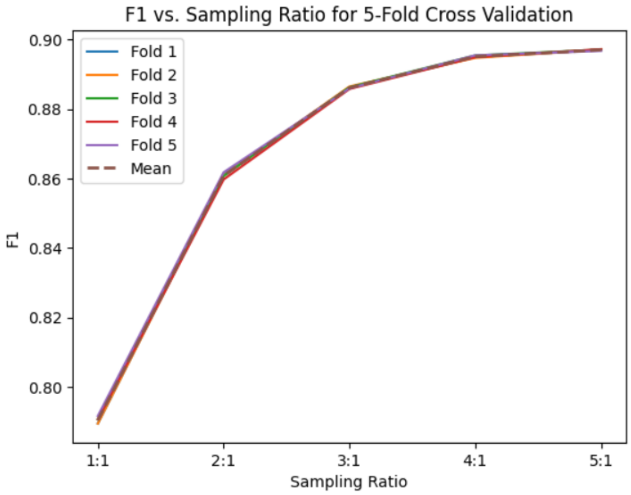

Figure 36: F1 on Testing Data using Sklearn XGB

AUC/ROC Curve

We constructed an AUC/ROC curve and got the following graph to assess the performance of different Scikit Learn Classification algorithms. According on our observations, XGBoost performs best, followed by Logistic Regression.

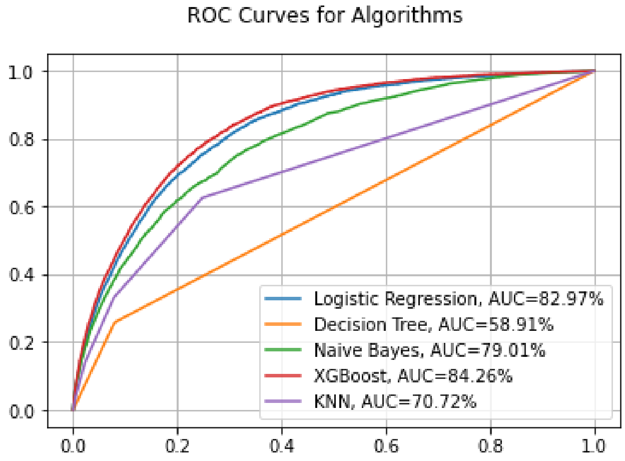

Figure 37: AUC/ROC of Sklearn Classifiers

Experiment and Results

We ran our implementation of Logistic Regression, Naive Bayes, Decision Tree with bagging and boosting & XgBoost and compared the same with Scikit-learn's on the dataset and calculated the accuracy, precision, recall and F1 score. Since the data is highly imbalanced, we performed 10 fold cross validation as well as undersampled the majority class. For Logistic Regression, accuracy and recall initially increase and remain constant thereafter, but precision goes down as we increase the sampling. For Naive Bayes, the accuracy goes up as the sampling ratio increases, but the recall goes down drastically, whereas the precision increases. We compared our implementation of bagging and scikit-learn bagging for depths (3, 5) & bag sizes (3, 4) and our Adaboost and scikit-learn Adaboost for depths (1, 2) and bag sizes (2, 3). We can see that our bagging with depth 5 and bag size 3 gives the lowest error rate of 8.51%. We can see that our Adaboost and scikit-learn Adaboost with depth 1 and bag size 2 give the same error rate of 8.65% on the test dataset. Training our own implementation of XgBoost with 5-Folds and undersampling ratio from 1:1 to 5:1 and then testing on the test dataset revealed accuracy, precision, recall and F1 to have almost the same value throughout with only minute differences. Python Library's XgBoost Classifier showed an increasing trend in accuracy and recall while the precision was decreasing as we increased the sampling ratio.

Discussion

The project's first stage was data analysis. The dataset was found to be imbalanced during the preprocessing step, with the "Yes" class label (8.56%) and the "No" class label (91.44%). Correlation matrix showed that areas with strong correlation values included stroke, physical health, difficulty walking, age category, diabetes, and kidney disease. Then, we examined how these values affected the output label by plotting the frequency of heart disease against the fields with the highest correlation values.

Since the data was imbalanced, we performed 10 fold cross validation and undersampled the majority class and performed experiments with various models, but did not observe any significant increase in the performance metrics.

After analyzing the correlation matrix in figure 3, we removed 15 features with low values ([-0.8, 0.8]) like ‘MentalHealth’, ‘GenHealth’, etc. We kept important features like ‘Smoking’, ‘Stroke’, ‘PhysicalHealth’, ‘DiffWalking’,‘AgeCategory’,‘Diabetic’,‘PhysicalActivity’ , ‘KidneyDisease’, ‘SkinCancer’ in the dataset. We can observe a significant increase in accuracy for Sklearn Decision Tree and Sklearn Naive Bayes after proper feature selection in the below figure 38.

We tried hyperparameter tuning using Scikit-learn’s RandomizedSearchCV to do a randomized search for hyperparameters for Logistic Regression and Random Forest Classifier, but did not observe any significant improvement in the metrics.

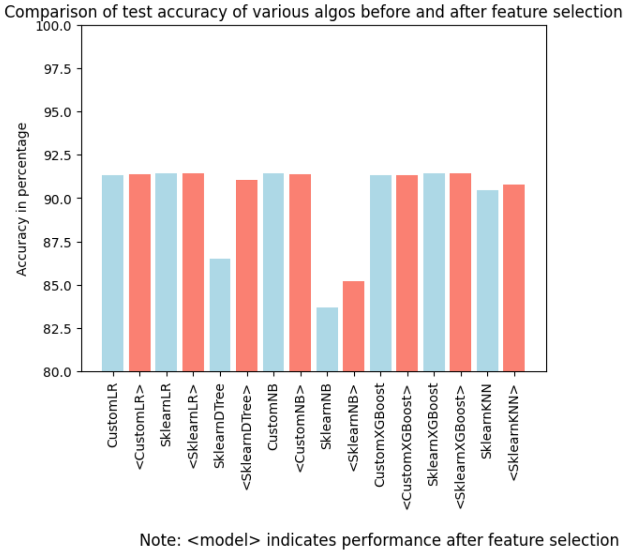

Figure 38: Comparison of test accuracy before and after feature selection

Conclusion

While recall is a crucial component to take into account in medical datasets, it is not always the only or the most crucial one. It is only one of several assessment criteria frequently used to gauge how well machine learning models perform on medical data.

Having said that, it is important to note that in this specific instance, Logistic Regression, Decision Trees with Bagging, and XGBoost outperformed other models in terms of recall values. This implies that these algorithms are very effective at locating genuine positive cases in medical data. However, while assessing the general performance of these models, it is still crucial to take into account additional metrics like precision, accuracy, and F1 score.

References

[1] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

[2] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: ACM.

[3] Patel B, Sengupta P. Machine learning for predicting cardiac events: what does the future hold? Expert Rev Cardiovasc Ther. 2020;18(2):77–84.

[4] Shah D, Patel S, Bharti SK. Heart Disease Prediction using Machine Learning Techniques. SN Computer Sci. 2020;1:345–6.

[5] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020).

[6] J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[7] McKinney, Wes. "pandas: a foundational Python library for data analysis and statistics." Python for high performance and scientific computing 14.9 (2011): 1-9.