Md. Mahfuzur Rahman Mahim

ML intern task

Contents

Introduction **1**

Background Studies **2**

0.1 Exploratory Data Analysis (EDA) 2

0.2 Performance Matrics 2

0.3 Supervised Learning Models 2

0.3.1 Logistic Regression: 2

0.3.2 Decision Tree: 3

0.3.3 Support Vector Machine: 3

0.3.4 Unsupervised Learning: 4

0.3.5 Ensemble Learning: 4

Methodology **5**

0.4 Overview: 5

0.5 Dataset & EDA: 5

0.6 Model Description: 7

Experimental Setup **12**

Result Analysis **13**

0.7 MIT-BIH dataset: 13

0.8 PTB Dataset: 14

Conclusion **15**

References **16**

Introduction

This task report presents a comprehensive analysis of a hands-on project on machine learning that utilized the ECG Heartbeat Categorization Dataset [8] from Kaggle. The dataset is mainly published on physionet [13]. The project aimed to explore the potential of state-of-the-art machine learning approaches in heartbeat classification using this dataset. The project employed 5 supervised learning approaches, a stack-based ensemble learning method, and an unsupervised learning approach to analyze the dataset. The dataset comprises a total of 109446 heartbeat signals that are classified into 5 categories. The main objective of the project was to showcase the performance of different machine learning approaches on the ECG Heartbeat Categorization Dataset. This report provides a detailed comparison of the performance of different models on different datasets and highlights how each model contributes to the final performance results. The report also discusses the implications of these findings for future research in the field of machine learning and ECG heartbeat classification. Overall, this report provides a valuable resource for anyone interested in the application of machine learning to ECG heartbeat classification.

Background Studies

0.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) [23] [15] is a crucial step in comprehending the structure and variables of a given dataset. To carry out EDA successfully, it is necessary to follow certain steps such as data visualization, descriptive statistics, data cleaning, grouping and aggregating, and correlation analysis. These steps help in comprehending the data's characteristics, identifying patterns, and discovering relationships between variables, thereby aiding in making informed decisions and drawing meaningful insights.

0.2 Performance Matrics

The performance matrices [11] evaluates classification accuracy using Accuracy, Recall, Precision, F-1 score, and AUC-ROC. Accuracy is the percentage of correct predictions, Recall is the prediction among actual positive values, F-1 score balances high precision with low recall, and AUC-ROC measures binary classification model performance.

0.3 Supervised Learning Models

We know that Supervised learning[24] is a category of machine learning that uses labeled datasets to train algorithms to predict outcomes and recognize patterns.

0.3.1 Logistic Regression:

Logistic regression is a classification algorithm that can be used for both binary and multiclass classification. It is a statistical method used to model the probability of a certain class. The maximum likelihood estimation (MLE) is used to estimate the coefficients (weights and biases)

of the logistic regression model.

0.3.2 Decision Tree:

Decision tree [6] is a classification algorithm that can work on both classification and regression tasks. It has a tree structure where each node represents the test data and the corresponding represents the solution from it. The tree splitting of decision tree happens based on gini [21] or by using entropy formula. However, it has a huge time complexity. Decision tree splits its tree based on information gain.

$$\textbf{Gini Index} = (1 - \sum \text{probability}^2)$$

$$\textbf{Gain}(S, A) = \textbf{H}(S) - \sum \left| \frac{S_v}{S} \right| \textbf{H}(S_v)$$

0.3.3 Support Vector Machine:

Linear SVC:

Linear SVC [19] is a classification tool that uses a linear kernel to divide different groups. It has a regularization parameter and a loss function called 'squared hinge'. It's useful for handling large amounts of data and can be used for both dense and sparse input. It's implemented in scikit-learn.

SVM:

It's a machine learning model that is used for both classification and regression tasks [1]. Based on data points, which are called support vectors, it creates a hyperplane (decision boundary) for classification or regression tasks. SVM converts low-dimensional data to high-dimensional data for classification tasks. The SVM kernels are chosen on the basis of hyperparameter tuning.

K-nearest:

K-nearest neighbor [18] is a machine learning classification algorithm that is commonly used for non-linear datasets. The algorithm works by classifying data based on the distance between data points. This distance is usually calculated using Euclidean distance or Manhattan distance. K-nearest neighbor is a highly interpretable algorithm and can be easily implemented. However,

one of the main drawbacks of this algorithm is the impact of outliers. Since the algorithm is heavily reliant on the distance between data points, outliers can significantly affect the accuracy of the model.

0.3.4 Unsupervised Learning:

Unsupervised learning is performed on an unlabeled dataset where data points are gathered in clusters.

K-means:

K-means [17] [5] clustering is an effective unsupervised learning algorithm used to group observations into k clusters based on their closest mean value to the cluster centroid. To determine the appropriate number of clusters for a given task, it's best to consider the specific needs and goals of the project. For instance, a T-shirt business may decide on the number of clusters based on the sizes they offer. If they only sell shirts in S, M, and L sizes, they can divide them into k=3 clusters, but if they also offer XL and XXL, they may need k=5 clusters. Another commonly used method for determining the optimal number of clusters is the elbow method.

0.3.5 Ensemble Learning:

Ensemble Learning is a powerful machine learning technique that leverages the strength of multiple models to deliver superior results.

Ensemble Stacking:

Ensemble Stacking is a specific approach within this technique that involves stacking models in two levels: the base model and the meta-model. The base model utilizes multiple models that work together to complement one another. The meta-model takes in the output of the base model, including labels and predictions, to generate a final prediction. However, it's important to note that stacking can lead to overfitting, which can be avoided by performing cross-validation at the outset.

Methodology

0.4 Overview:

The result of this task is an outcome that is based on thorough study. I utilized a combination of Supervised Learning methods, as well as an unsupervised approach after dropping the labels, and ultimately applied stacking Ensemble approaches. Before that, I conducted an EDA to gain insights into the dataset and determine the most appropriate methods to apply. To showcase the effectiveness of these approaches, I provided detailed visualizations and reports, including classification reports and confusion matrices. This task can rightfully be considered an outcome-based task.

0.5 Dataset & EDA:

The dataset is ECG Heartbeat Categorization Dataset [8], available in Kaggle but originally from the author Kauchuee et.el.[8]. The dataset comprises data from two databases named, the MIT-BIH Arrhythmia Dataset[10] and the PTB Diagnostic ECG Database[3]. The dataset contains a total of 123996 samples where MIT-BIH contains 109446 samples and PTB database contains 14550 samples. The dataset is a time series tabular dataset and it's min-max normalized. The samples (ECG signals) are preprocessed and segmented. The MIT-BIH dataset comprises 5 Classes: ['N': 0, 'S': 1, 'V': 2, 'F': 3, 'Q': 4]. Here, N: Normal beat, S: Supraventricular premature beat, V: Premature ventricular contraction, F: Fusion of ventricular and normal beat, Q: Unclassifiable beat. The MIT-BIH dataset is imbalanced [4] as most of the samples (72471) reside under the category Normal beats (N: '0') and the categories correlation was visualized also by using correlation matrix. The PTB database has only two classes, normal heart rates(10506 samples) and abnormal heart rates(4046 samples).

To visualize the datasets, I utilized the matplotlib [9] and seaborn libraries [16]. I employed the circular pie chart of matplotlib to display the distribution of categorical data and utilized seaborn library to observe the scatter of each column (feature). I also used the pandas library [12] to obtain an overall description of the datasets. It was clear from the description that the dataset was min-max normalized. Furthermore, I calculated the data correlation matrix and visualized it to see how the dataset was correlated with each feature and to observe the relationship between categories and features. Ultimately, the pie chart and data description proved to be extremely useful.

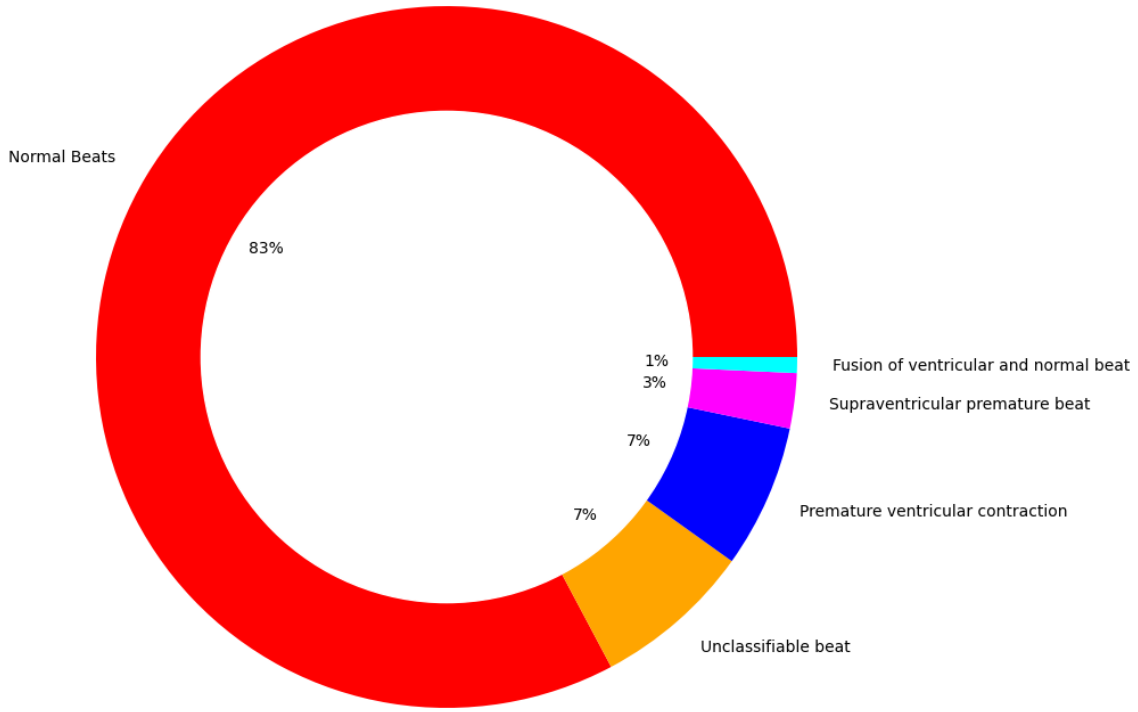

Figure 1: Donut chart on MIT-BIH Arrhythmia Dataset.

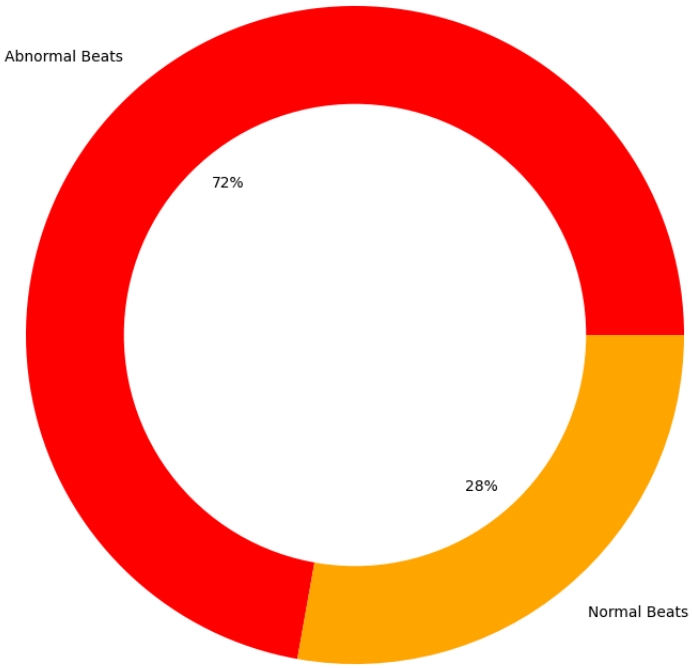

Figure 2: Donut chart on the PTB Diagnostic ECG Database

0.6 Model Description:

I had to perform sampling of the whole dataset before applying supervised learning algorithm. I performed oversampling with SMOTE [2], (undersampling) with Tomek [20] and finally performed SMOTE-Tomek [22] on the dataset as SMOTE-Tomek does both SMOTE on minority class and Tomek on majority class. For PTB dataset the dataset had two separate files of normal and abnormal heartbeats, so I combined them and then performed the application of model and data engineering on them. Then I first performed Logistic regression on the datasets and for MIT-BIH the model underperformed on both oversampled data and SMOTE-Tomek applied data. The model did good on undersampled data but the problem was, that it gets biased to a certain category (which is N: '0'). Also, all the features of the dataset are important so performing undersampling isn't a suitable way. So for the next ML algorithms, I used the balanced data from applying SMOTE-Tomek. I applied Decision Tree, Linear SVC, Support Vector Machine (SVM), and K-Nearest as supervised methods. From applying these models and from their results It was seen that k-nearest performed the best for both MIT-BIH dataset (multiclass

classification) and PTB dataset with an accuracy of respectively 99.09% and 94.67%. Now for MIT-BIH dataset SVM performed 85.02% which decreased to 82.8% while testing the model with the unseen data. Decision Tree performed second best in both the datasets with accuracy respectively, 98.12% for MIT-BIH and 93.37% for PTB dataset. But in the case of MIT-BIH dataset the model performed poorly while testing on unseen data because It performed well on identifying categories '0', '2', and '4' but performed poorly while trying to identify categories '1' and '3'. Linear SVC performed the worst for MIT-BIH on testing with an accuracy of 65.52%. For PTB dataset both logistic regression and Linear SVC performed the same. Then I tried applying Ensemble methods to the datasets. As the methods are ensemble methods I didn't use the sampled data. I just basically split the datasets without sampling and performed ensemble learning on them.

For ensemble Learning, I used Random Forest, Gradient Boost (which is a stacking-based ensemble method), and XGboost [14]. As stacking-based boosting algorithms are prone to outliers I used Stratified K-Fold(SK-Fold) to perform cross-validation. Now when applied Random forest the same problem remained as the model couldn't do better for all categories. Then applied SK-Fold on logistic regression and Random Forest and both of the models did exceptionally well as the accuracy for each fold was above 90%. Now finally applied XGBoost and it gave a rather well-performed result despite using original imbalanced data. But still, it performed rather poorly in categories '1' and '3' comparing the other categories. To remedy this I assigned weights to the classes where I penalized the categories having majority class with less weights and categories with minority class with higher weights. Then I performed XGBoost again and this time my model was able to predict above 80% for all the classes in MIT-BIH dataset.

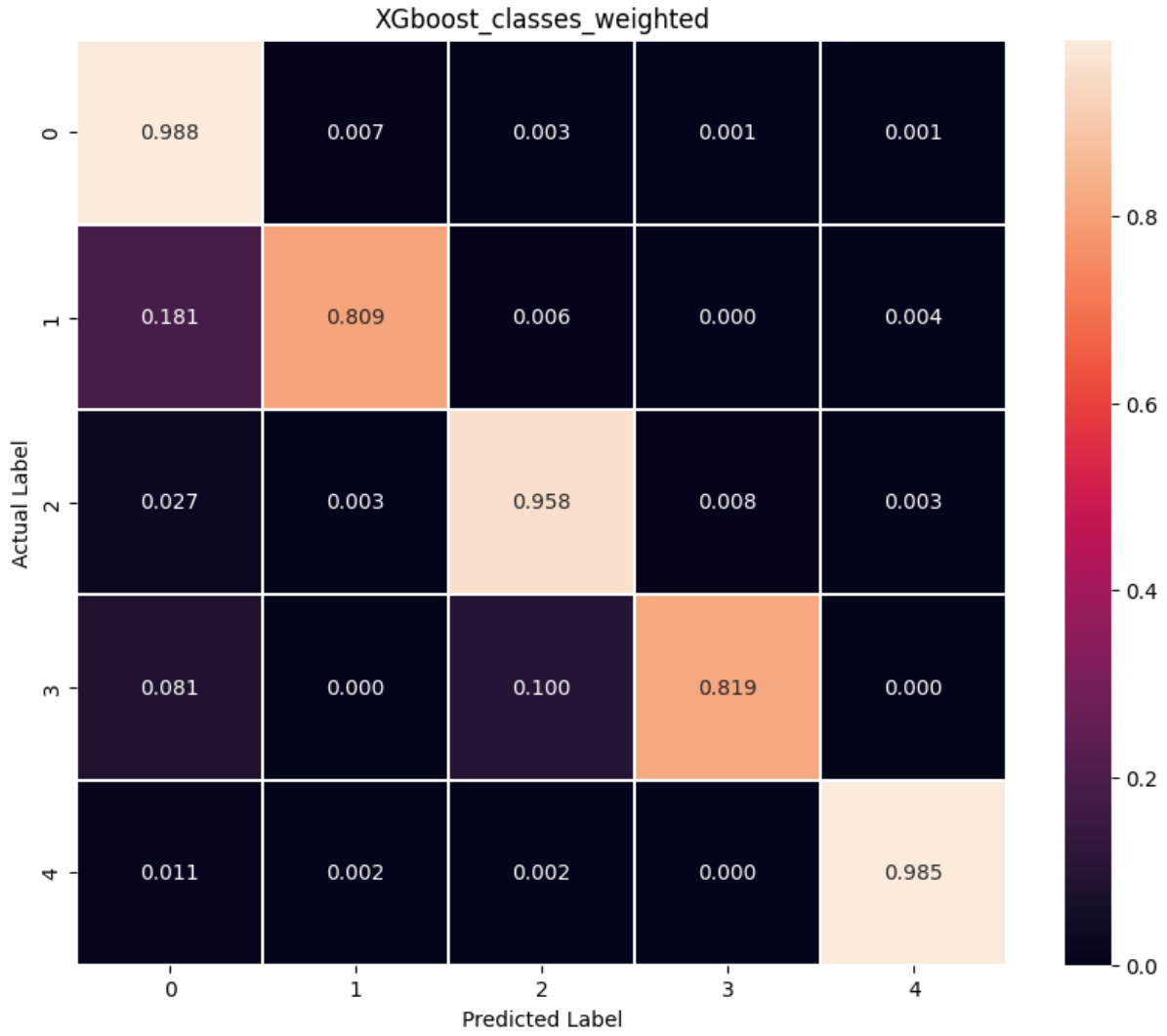

Figure 3: Performance of XGboost for weighted classes on MIT-BIH dataset.

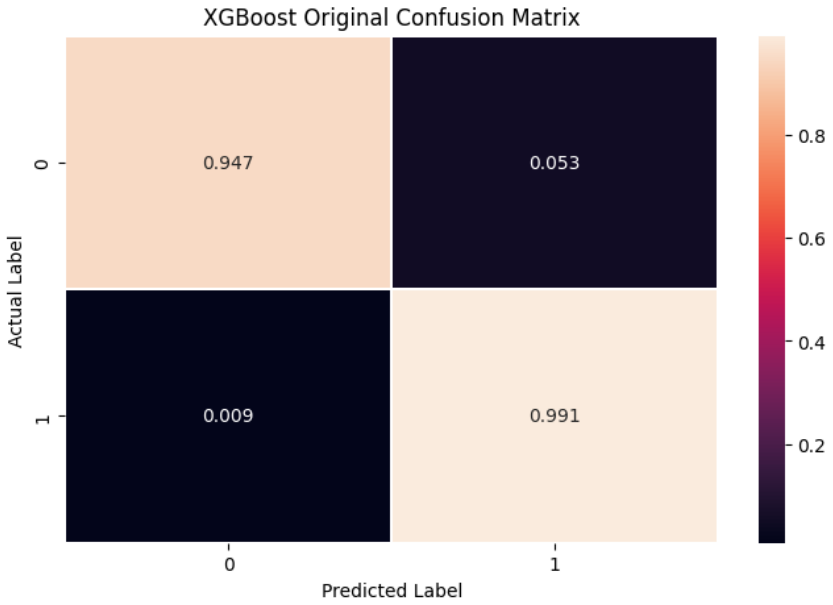

Figure 4: Performance of XGboost on PTB dataset.

For PTB dataset as the data has only two classes all the models worked well on it when using the balanced sampled data from applying SMOTE-Tomek. Even logistic regression performed above 80 percent for this dataset.

Now, Finally, I dropped the classes of the datasets to perform K-means clustering on both datasets. I performed PCA (Principal Component Analysis) so that I could visualize the clusters. I used silhouette score [7] to see how well the clustering algorithm performed on the datasets and for both datasets, the score was very low. For MIT-BIH the silhouette score was 0.12 and for PTB database the score was 0.296 which says the clusters are overlapping with each other on their decision boundary. The graph figure also expresses the same.

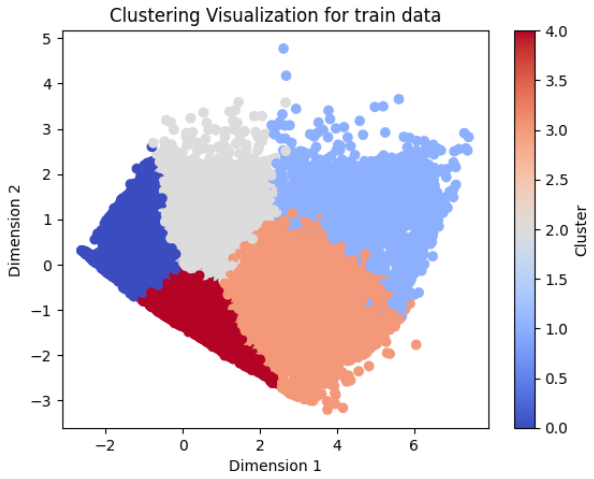

Figure 5: Clustering on MIT-BIH train data

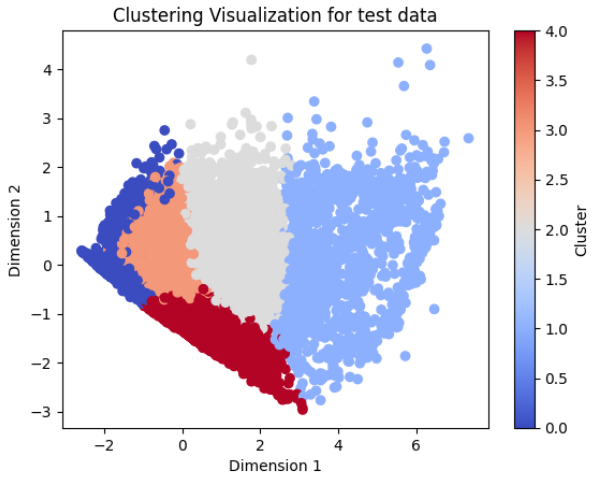

Figure 6: Clustering on MIT-BIH test data

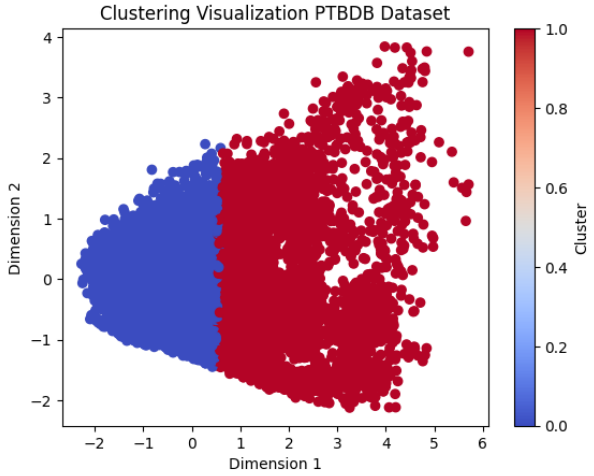

Figure 7: Clustering on PTB dataset

Experimental Setup

To perform this task I created a virtual environment where I installed and downloaded all the dependencies. I controlled and performed the whole process from here. As for the Virtual Environment, I created it using Python in VS-code. Version control was managed using Git to create a repository on the cloud, which was regularly updated. The git link of the project is given here,gitlink

Result Analysis

0.7 MIT-BIH dataset:

The model was biased to the majority class when applied to both undersampled data and balanced sampled data (SMOTE-Tomek). Decision Tree performed well on training data but poorly on predicting class '1' and '3' during testing. Linear SVC had an accuracy of 78.25% and performed poorly on individual class scores, except for classes '3' and '4'. SVM performed better than Logistic regression and Linear SVC, but not better than Decision Tree. K-nearest worked best on the dataset with an average accuracy of 99.09% on training and 95.09% on testing and performed well on predicting each class. XGBoost performed the best in identifying individual classes using ensemble methods on an imbalanced dataset. Initially, it didn't work well, so I assigned less weight to the majority classes. After this, it performed well with a lowest predicting score of 80.7% for class '3' and 82.2% for class '1', while the rest had a performance above 94%.

Categories	Logistic Regression(Balanced)	Linear SVC(Balanced)	Decision Tree(Balanced)	SVM(Balanced)	k-Nearest(Balanced)
0	63.0%	61.7%	94.5%	82.2%	95.8%
1	66.5%	68.5%	70.3%	67.3%	80.6%
2	70.9%	72.7%	88.5%	85.4%	94%
3	87.0%	89.1%	71.0%	90.1%	84.6%
4	92.2%	92.9%	95.2%	91.9%	97.8%

Figure 8: Supervised methods' performance on MIT-BIH dataset in a comparison table.

0.8 PTB Dataset:

The PTB dataset comprises only two classes and all the models except for Logistic Regression and Linear SVC performed well on it. These two models had relatively lower accuracy rates of 80.25% and 80.35%, respectively. On the other hand, the SVM, Decision Tree, and ensemble methods demonstrated exceptional performance, with accuracy rates exceeding 92%. The clustering performance of the entire dataset was low, with a silhouette score of 0.12 for MIT-BIH and 0.296 for PTB-db.

Categories	Logistic Regression(Balanced)	Linear SVC(Balanced)	Decision Tree(Balanced)	SVM(Balanced)	k-Nearest(Balanced)
0	84.4%	84.4%	89.0%	94.2%	96.7%
1	76.9%	77.3%	93.5%	88.1%	89.3%

Figure 9: Supervised methods' performance on PTB dataset in a comparison table.

Conclusion

I engaged in a learning task where I delved deep to gain knowledge about handling datasets and performing learning algorithms on them. My primary motivation throughout the task was to learn as much as possible to improve my future performance. I sought assistance from various individuals, sources, and articles, which enabled me to achieve the current state of the task. Although I completed most of the tasks, I still have some remaining tasks, such as obtaining a classification report for each fold when I applied SK-Fold for cross-validation on ensemble learning. In the future, I plan to implement this and create a comparison graph to make it easier to visualize the models' performance.

References

[1] *1.4. Support Vector Machines — scikit-learn.org.* https://scikit-learn.org/stable/modules/svm.html. [Accessed 24-03-2024].

[2] Ansh Bordia. *Handling Imbalanced Data by Oversampling with SMOTE and its Variants — medium.com.* https://medium.com/analytics-vidhya/handling-imbalanced-data-by-oversampling-with-smote-and-its-variants-23a4bf188eaf. [Accessed 25-03-2024].

[3] R. Bousseljot, D. Kreiseler, and A. Schnabel. In: *Biomedical Engineering / Biomedizinische Technik* 40.s1 (1995), pp. 317–318. DOI: doi:10.1515/bmte.1995.40.s1.317. URL: https://doi.org/10.1515/bmte.1995.40.s1.317.

[4] Kartik Chaudhary. *How to deal with Imbalanced data in classification? — medium.com.* https://medium.com/game-of-bits/how-to-deal-with-imbalanced-data-in-classification-bd03cfc66066. [Accessed 25-03-2024].

[5] Kasun Dissanayake. *Machine Learning Algorithms(14)—K-Means Clustering and Hierarchical Clustering — medium.com.* https://medium.com/towardsdev/machine-learning-algorithms-14-k-means-clustering-and-hierarchical-clustering-46acc005057d. [Accessed 25-03-2024].

[6] Mohtadi Ben Fraj. *InDepth: Parameter tuning for Decision Tree — mohtedibf.* https://medium.com/@mohtedibf/indepth-parameter-tuning-for-decision-tree-6753118a03c3. [Accessed 25-03-2024].

[7] Hazal Gültekin. *What is Silhouette Score? — hazallgultekin.* https://medium.com/@hazallgultekin/what-is-silhouette-score-f428fb39bf9a. [Accessed 25-03-2024].

[8] Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. "ECG Heartbeat Classification: A Deep Transferable Representation". In: *CoRR* abs/1805.00794 (2018). arXiv: 1805.00794. URL: http://arxiv.org/abs/1805.00794.

[9] *Matplotlib x2014; Visualization with Python — matplotlib.org*. https://matplotlib.org/. [Accessed 27-03-2024].

[10] G.B. Moody and R.G. Mark. "The impact of the MIT-BIH Arrhythmia Database". In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), pp. 45–50. DOI: 10.1109/51.932724.

[11] Riddhi Nisar. *(Visually) Interpreting the confusion-matrix: — medium.com*. https://medium.com/analytics-vidhya/visually-interpreting-the-confusion-matrix-787a70b65678. [Accessed 25-03-2024].

[12] *pandas documentation x2014; pandas 2.2.1 documentation — pandas.pydata.org*. https://pandas.pydata.org/docs/index.html. [Accessed 27-03-2024].

[13] *PhysioNet — physionet.org*. https://physionet.org/. [Accessed 24-03-2024].

[14] RITHP. *XGBoost and imbalanced datasets: Strategies for handling class imbalance — rithpansanga*. https://medium.com/@rithpansanga/xgboost-and-imbalanced-datasets-strategies-for-handling-class-imbalance-cdd810b3905c. [Accessed 25-03-2024].

[15] Navami S. *Complete Exploratory Data Analysis(EDA) using Python — navamisunil174*. https://medium.com/@navamisunil174/exploratory-data-analysis-of-breast-cancer-survival-prediction-dataset-c423e4137e38. [Accessed 25-03-2024].

[16] *seaborn: statistical data visualization x2014; seaborn 0.13.2 documentation — seaborn.pydata.org*. https://seaborn.pydata.org/. [Accessed 27-03-2024].

[17] *sklearn.cluster.KMeans — scikit-learn.org*. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. [Accessed 25-03-2024].

[18] *sklearn.neighbors.KNeighborsClassifier — scikit-learn.org*. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html. [Accessed 24-03-2024].

[19] *sklearn.svm.LinearSVC — scikit-learn.org.* https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html. [Accessed 24-03-2024].

[20] *TomekLinks x2014; Version 0.12.0 — imbalanced-learn.org.* https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.TomekLinks.html. [Accessed 25-03-2024].

[21] Neelam Tyagi. *Understanding the Gini Index and Information Gain in Decision Trees — medium.com.* https://medium.com/analytics-steps/understanding-the-gini-index-and-information-gain-in-decision-trees-ab4720518ba8. [Accessed 24-03-2024].

[22] Raden Aurelius Andhika Viadinugroho. *Imbalanced Classification in Python: SMOTE-Tomek Links Method — towardsdatascience.com.* https://towardsdatascience.com/imbalanced-classification-in-python-smote-tomek-links-method-6e48dfe69bbc. [Accessed 25-03-2024].

[23] Richard Warepam. *How to Conduct an Effective Exploratory Data Analysis (EDA) — medium.com.* https://medium.com/illumination/how-to-conduct-an-effective-exploratory-data-analysis-eda-fa4e65ab7735. [Accessed 24-03-2024].

[24] *What is Supervised Learning? — Google Cloud — cloud.google.com.* https://cloud.google.com/discover/what-is-supervised-learning. [Accessed 24-03-2024].