Machine Learning Report

Siddharth Bahekar
College of Engineering
Northeastern University
Toronto,ON
bahekar.si@northeastern.edu

Abstract

In this report, I have applied and explored the performance of supervised learning algorithm on two datasets. This report outlines the exploratory data analysis (EDA) and modeling process undertaken to train the Artificial Neural Network (ANN) algorithm. The two datasets included are **Heart Attack Analysis & Prediction** dataset and **Wine Quality** dataset. The learning curve, model complexity and model training has been performed and analyzed on both the datasets.

1. Dataset

The **heart attack analysis** dataset provides a comprehensive set of attributes for heart disease prediction, allowing for in-depth analysis of various factors contributing to heart health. The heart disease dataset contains 303 rows and 14 columns. The problem at hand is to classify whether a patient has heart disease or not based on various medical attributes. This is a binary classification problem using a dataset containing features and other relevant medical information.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trtbps    303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalachh  303 non-null    int64
 8   exng      303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64
 11  caa       303 non-null    int64
 12  thall     303 non-null    int64
 13  output    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

The **wine quality** dataset contains observations related to the physicochemical properties of red wine, including features and the target variable: wine quality. The dataset used in this analysis comprises 1,599 rows and 12 columns. Each row represents a unique wine sample with its respective measurements. The target variable, "quality," is an ordinal feature ranging from 3 to 8, indicating the wine's quality rating. This dataset provides a comprehensive view of various chemical attributes that potentially influence wine quality, allowing for in-depth exploratory data analysis and predictive modeling.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

2. Null Values

```
# Check for missing values
print(data.isnull().sum())
```

```
age         0
sex         0
cp          0
trtbps      0
chol        0
fbs         0
restecg     0
thalachh    0
exng        0
oldpeak     0
slp         0
caa         0
thall       0
output      0
dtype: int64
```

```
# Check for missing values
df.isnull().sum()
```

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

Both the datasets do not contain any null or missing values, so there is no need to clean the data.

3. Dataset Description

Heart Attack dataset

```
# Display summary statistics
data.describe()
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000

sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

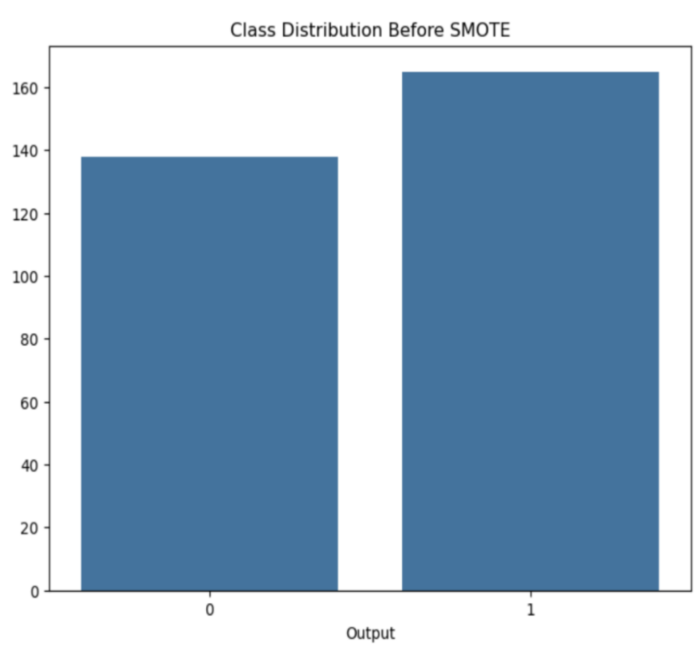
Wine Quality dataset

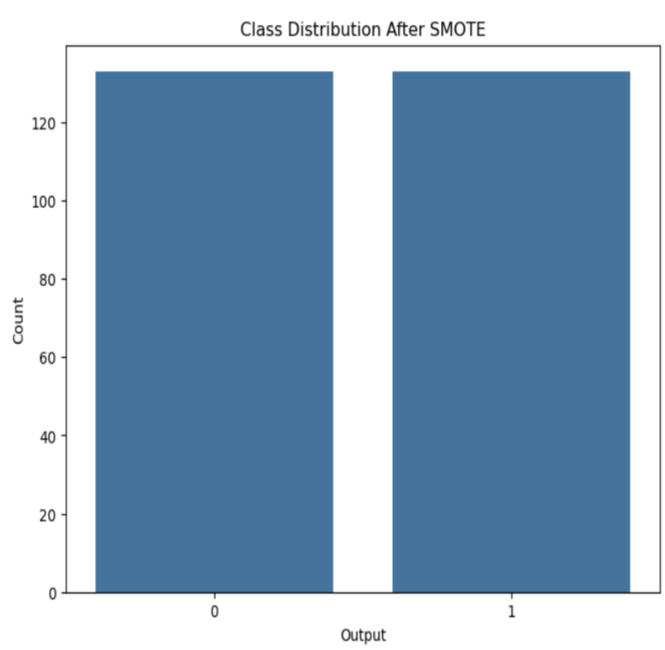
```python
# Display summary statistics
df.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

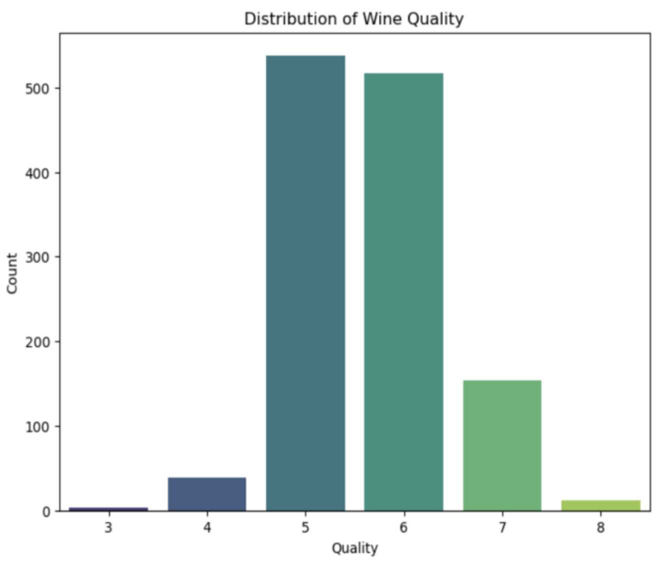
4. Dataset Imbalance

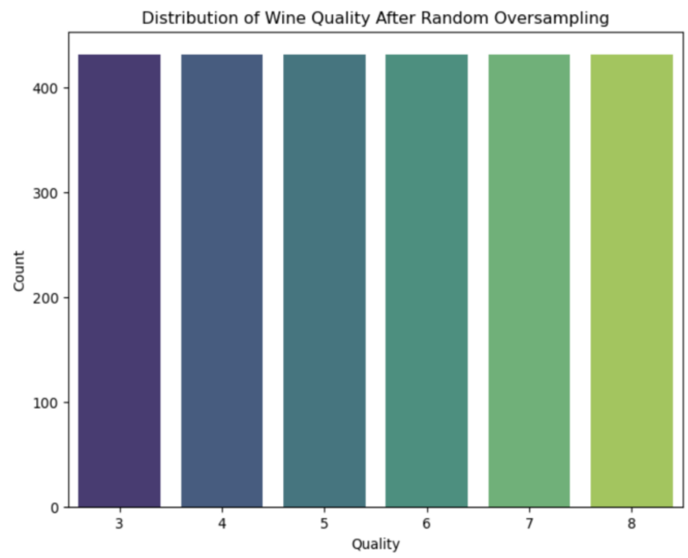
In this analysis of the heart disease dataset, I encountered a significant class imbalance issue. Upon examination of the target variable distribution, I found that the number of patients without heart disease (class 0) was considerably higher than those with heart disease (class 1). To address this issue, I implemented the **Synthetic Minority Over-sampling Technique (SMOTE)**. SMOTE is an oversampling method that creates synthetic examples of the minority class to balance the dataset. It effectively addressed the class imbalance, I was cautious to avoid potential overfitting and ensured a more equitable treatment of both classes, which is particularly important in medical diagnostics where false negatives can have serious consequences.





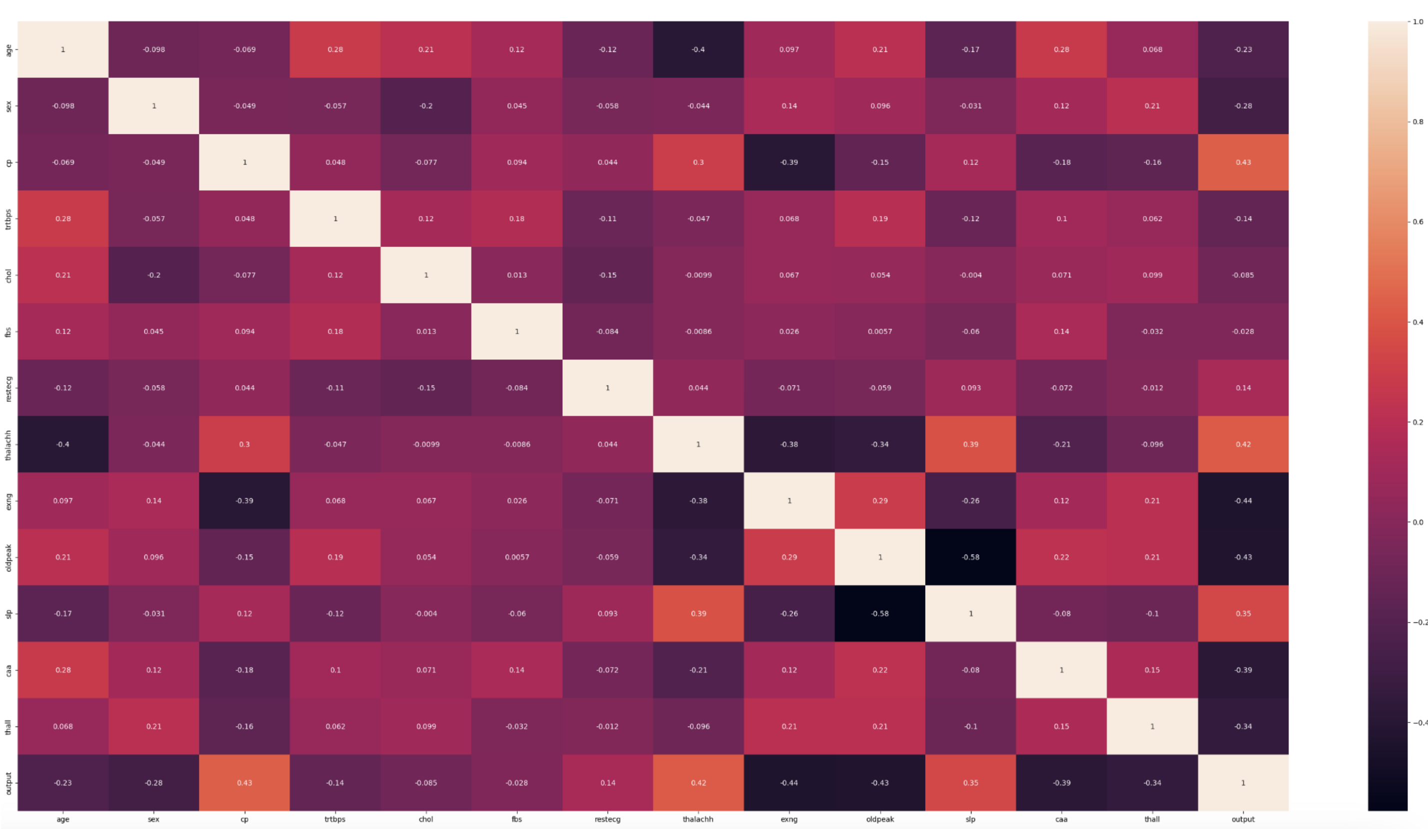
The wine quality dataset exhibited a significant class imbalance, as illustrated in the distribution chart of wine quality. Most samples were concentrated around quality scores of 5 and 6, with fewer instances of other ratings like 3, 4, 7, and 8. This imbalance can skew model training, causing it to favor the majority classes and potentially overlook the minority ones. To address this issue, I employed **Random Oversampling**. This technique duplicates samples from the minority classes to create a more balanced dataset. By doing so, I ensured that each class was represented more equally during model training, which helps improve the model's ability to generalize across all quality ratings.



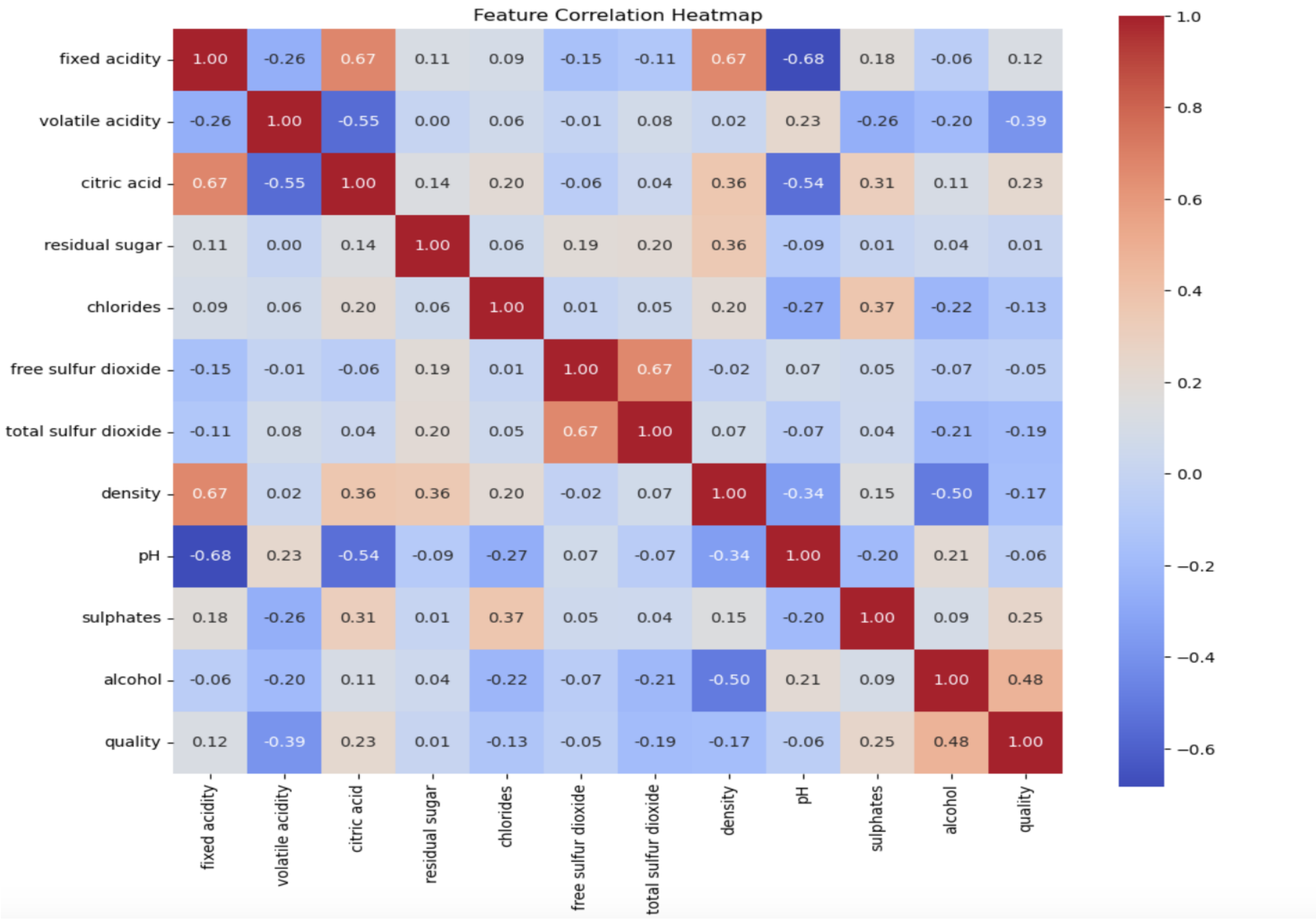


5. Correlation Matrices

I generated a correlation heatmap to visualize the relationships between different features in our heart disease dataset. A strong positive correlation was observed between 'chest pain' and the target variable, indicating that certain types of chest pain are closely associated with the presence of heart disease. 'Max heart rate' showed a moderate negative correlation with the target variable, suggesting that lower maximum heart rates might be indicative of heart disease. 'ST depression' exhibited a positive correlation with the target, implying that higher ST depression values are associated with an increased likelihood of heart disease. These correlations provided valuable insights into which features might be most predictive of heart disease, guiding our feature selection process and helping us understand the underlying patterns in the data.

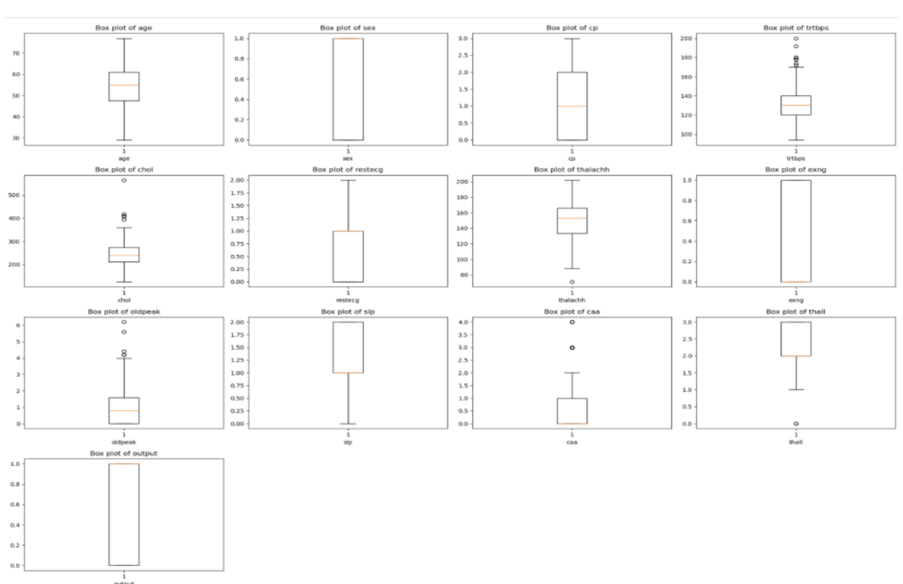


The correlation heatmap provides a visual representation of the relationships between different features in the dataset. It highlights how each feature correlates with wine quality and with each other. Notably, alcohol content shows a strong positive correlation with quality, suggesting its importance in predicting higher quality wines. Volatile acidity has a negative correlation, indicating that higher acidity might detract from perceived quality. Features like residual sugar and chlorides exhibit low correlation with quality, suggesting they may have less impact on the model's predictive power.

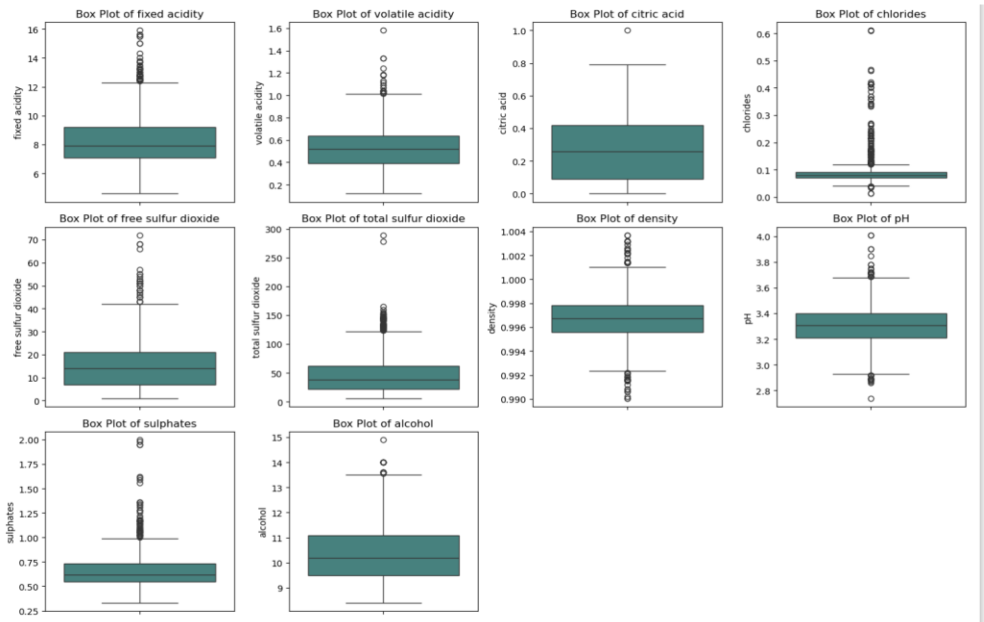


6. Outliers

To identify potential outliers in heart disease dataset, I employed box plots for each numerical feature. This analysis revealed the 'cholesterol' feature contained several high outliers, with some values significantly above the upper quartile. 'Age' and 'resting blood pressure' also showed a few outliers, but these were less extreme compared to cholesterol. Given the medical nature of our data, I decided to retain these outliers as they could represent genuine, rare, cases that are important for our model to learn from. I made note of these outliers to monitor their potential impact on our model's performance.



Outlier detection was performed using box plots for wine quality dataset, revealing extreme values in features such as residual sugar and sulphates. These outliers can skew analysis and model performance if not addressed. By identifying and potentially removing these outliers, I aimed to improve model robustness. I have used Inter Quartile Range to detect the outliers. I will keep the outliers within 1.5*(lower bound of IQR) - 1*5(upper bound of IQR) range test.

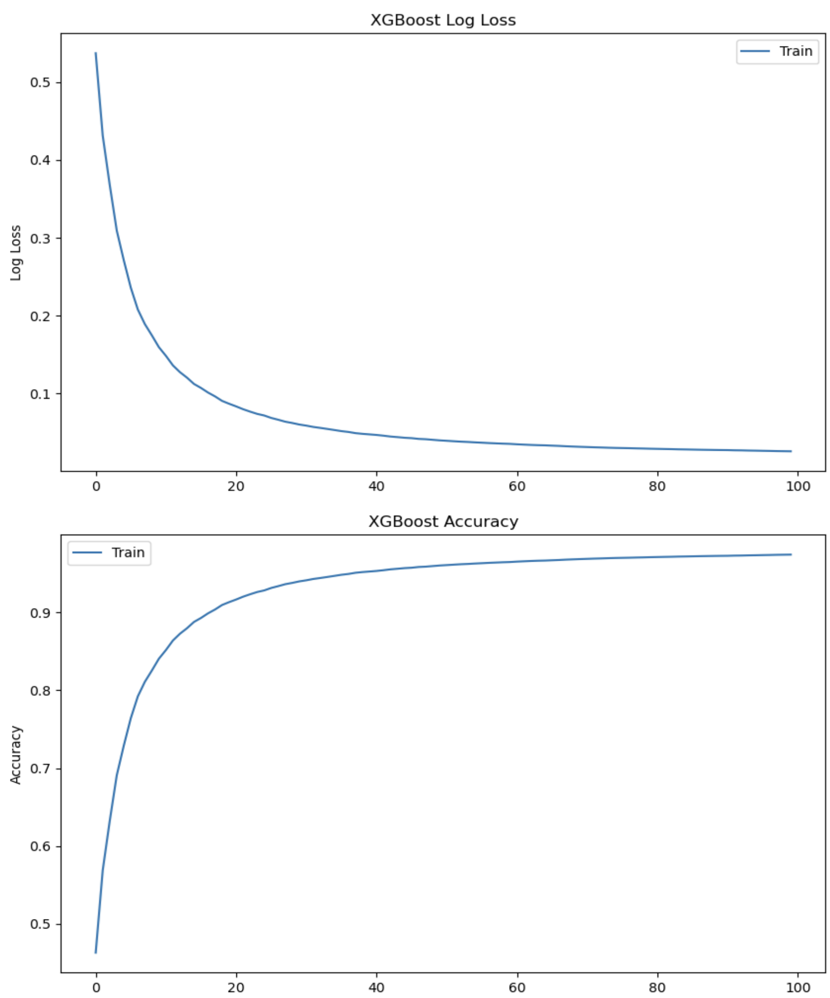


7. Why are these datasets interesting?

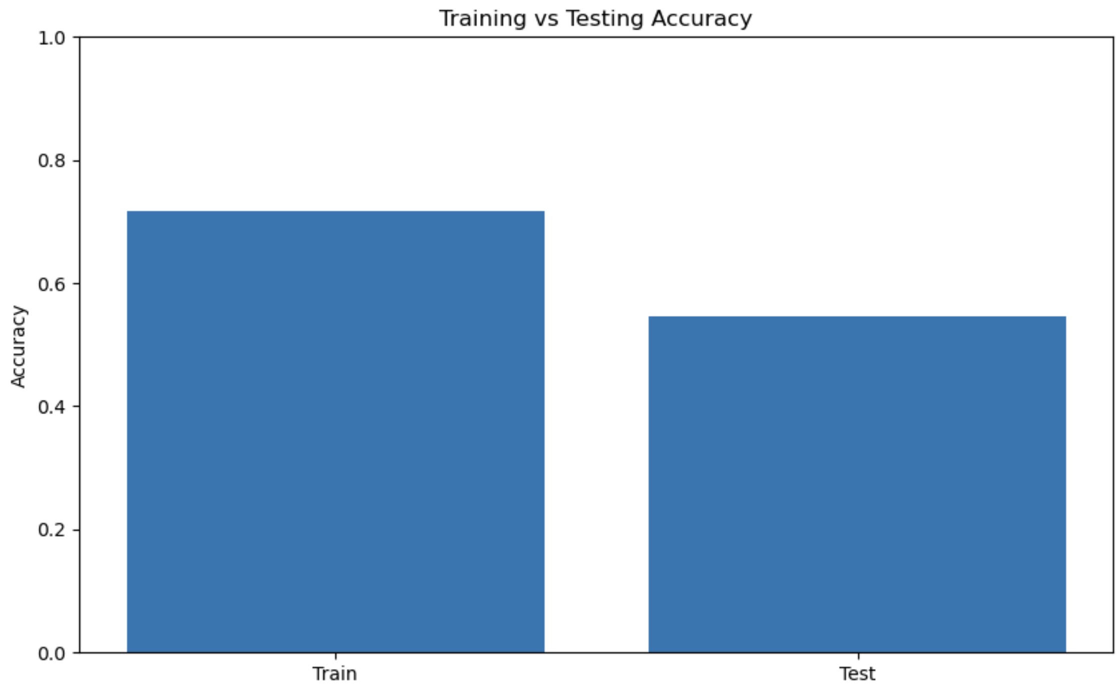
The dataset I have analyzed presents a compelling case study that bridges the gap between medical science and data analytics. Its real-world implications in potentially aiding early diagnosis of a leading cause of mortality worldwide make it particularly significant. The wine quality dataset is particularly interesting due to its practical application in the wine industry and its potential to reveal insights into the factors influencing wine quality. By analyzing physicochemical properties such as acidity, sugar content, and alcohol levels, I can identify key attributes that contribute to higher quality ratings. The challenge of addressing class imbalance and outliers provides a rich opportunity for applying advanced data preprocessing and machine learning techniques. This makes the dataset a valuable resource for both academic research and industry applications.

8. Boosting (XGBoost)

In heart dataset, XGBoost classifier is implemented using the XGBClassifier class from the XGBoost library. The classifier is configured with a random seed for reproducibility (random_state=42) and is set to evaluate its performance using the 'logloss' metric. The use_label_encoder=False parameter disables the use of a label encoder, which is deprecated in newer versions of XGBoost. The training accuracy is 1.0000, indicating perfect prediction on the training data, while the testing accuracy is 0.8033, suggesting that approximately 80.33% of test instances are correctly classified. The results show that the model achieves perfect scores across all metrics on the training data but performs less optimally on the test data. This discrepancy suggests potential overfitting, where the model learns the training data too well but does not generalize effectively to unseen data. The code also includes visualization of log loss over training epochs, which helps in assessing the model's convergence and learning behavior.

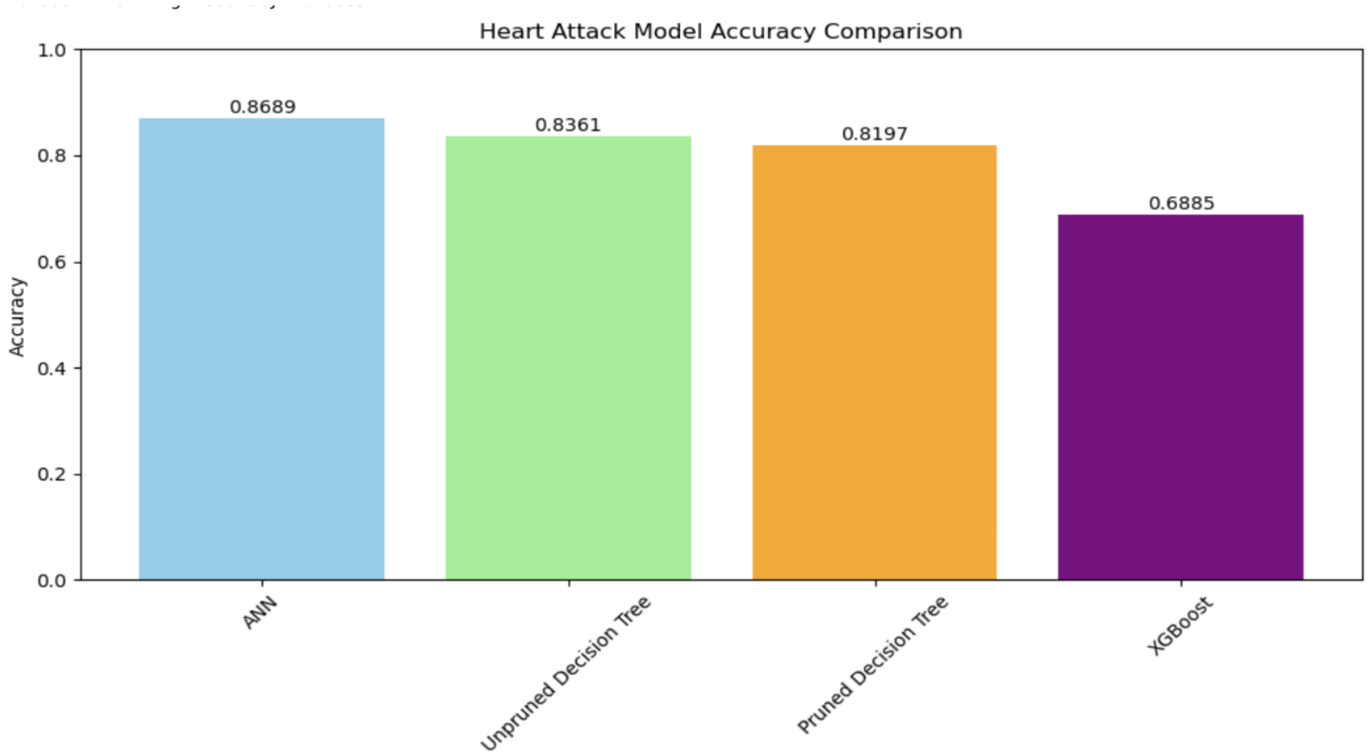


In wine dataset, XGBoost classifier is implemented using the XGBClassifier class from the XGBoost library. Gradient boosting works by sequentially adding models to correct errors made by existing models, optimizing a loss function through gradient descent. The initial model's performance is evaluated using accuracy, precision, recall, and F1 score, revealing a perfect train accuracy (1.0000) but a lower test accuracy (0.6969), indicating potential overfitting. To improve generalization, I implemented hyperparameter tuning using GridSearchCV, which explores different combinations of parameters such as max_depth, learning_rate, n_estimators, and min_child_weight. Despite these efforts, the best model slightly reduces overfitting but still shows a decrease in test accuracy (0.6750) compared to the initial model, suggesting that further tuning or alternative approaches may be necessary.

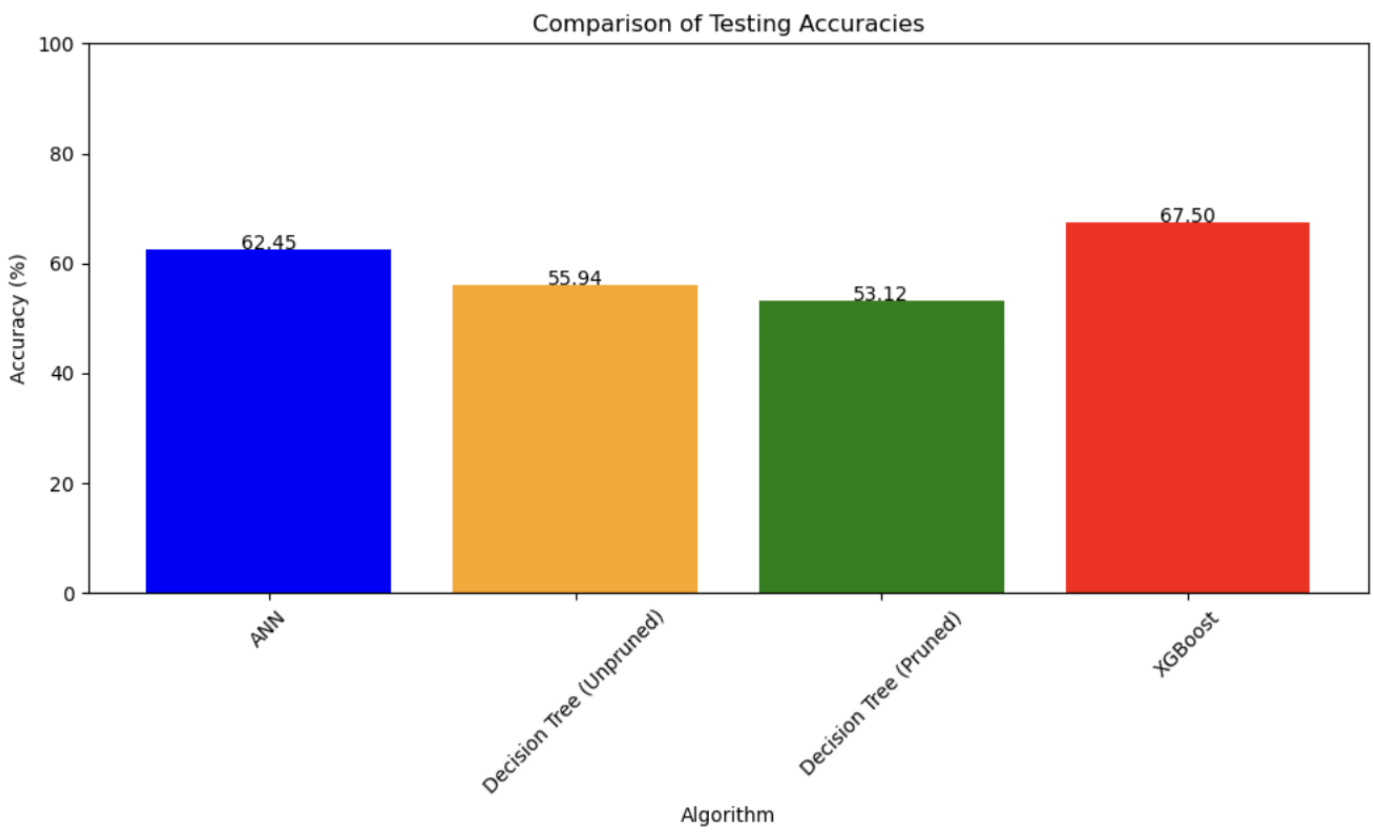


9. Conclusion

For heart dataset , if your primary goal is high accuracy on training data and you have mechanisms to prevent overfitting (like cross-validation), ANN might be preferred. For balanced performance with interpretability, a pruned decision tree could be more suitable. If computational efficiency and handling structured data are key concerns, exploring further tuning for XGBoost might be worthwhile.



For wine quality dataset, the comparison of testing accuracies among different algorithms—Artificial Neural Network (ANN), Decision Tree (Unpruned), Decision Tree (Pruned), and XGBoost—reveals distinct performance characteristics. XGBoost achieves the highest accuracy at 67.50%, followed by ANN at 62.45%. The unpruned decision tree scores 55.94%, while the pruned version achieves 53.12%. XGBoost stands out as the most accurate model, likely due to its ensemble learning approach, which combines multiple weak learners to improve predictive performance. If the primary goal is maximizing accuracy and computational resources are not a constraint, XGBoost is the best choice. For simpler models with faster training times and interpretability, decision trees (especially pruned) could be more appropriate. ANN serves as a middle ground with good accuracy and flexibility but requires more data and tuning.



10. References

- https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset
- https://labelyourdata.com/articles/machine-learning-for-wine-quality-prediction#
- https://archive.ics.uci.edu/
- ChatGPT