Intro to Deep Learning - Final Project

Semester A 2025

Authors: Yigal Meshulam: יגאל משולם
Yuval Wilf: יובל ווילף

Title:

Machine & Deep Learning-Based Classification of Heart Disease Using Health Risk Indicators

About this document:

Todo: add about the colab, reference between this document sections and colab cells.

Keywords:

Deep learning, Heart disease prediction, Neural networks, Classification, Healthcare analytics

Abstract:

Heart disease remains one of the leading causes of mortality worldwide, necessitating accurate and early prediction for effective prevention. This study employs deep learning techniques to classify heart disease status based on a dataset comprising various health indicators. The Heart Disease dataset, obtained from Kaggle, includes risk factors such as age, gender, blood pressure, cholesterol levels, smoking habits, and other health-related parameters. We developed and trained a neural network model based on health metrics to classify individuals as having or not having heart disease. The model's performance is evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score. The results demonstrate the effectiveness of deep learning in identifying heart disease risk, providing insights for healthcare applications. The code implementation is available in a Colab notebook on GitHub: Deep Learning Project.

1. Introduction

Heart disease is a major public health issue and a leading cause of death worldwide. Traditional risk assessment methods rely on statistical analysis and expert-driven evaluations, which may not fully capture complex interactions among risk factors. Deep learning has emerged as a powerful tool in healthcare analytics, offering improved prediction capabilities through automated feature extraction. In this study, we apply a deep learning model to classify heart disease based on a dataset containing multiple health risk indicators.

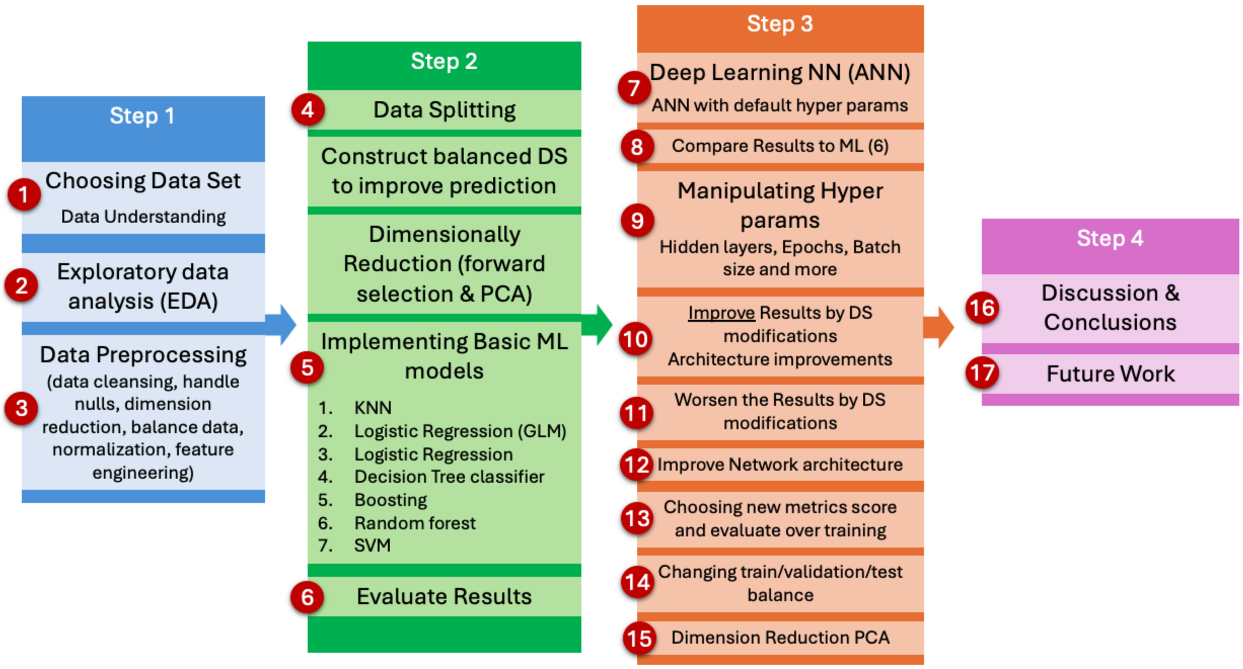
2. Literature Review

Recent advances in artificial intelligence (AI) and machine learning (ML) have significantly improved medical diagnosis and prediction systems. Various studies have explored logistic regression, decision trees, support vector machines (SVM), and ensemble models for heart disease prediction. However, deep learning models, particularly neural networks, have shown superior performance due to their ability to learn hierarchical patterns in complex datasets.

For example, Rajkomar et al. (2018) demonstrated the effectiveness of deep learning in medical diagnosis, highlighting its ability to detect complex interactions in patient data. Similarly, Esteva et al. (2017) applied deep learning in dermatology, showing improved classification performance compared to traditional methods. In the cardiovascular domain, Krittanawong et al. (2019) reviewed AI applications in cardiology, emphasizing the role of neural networks in predicting cardiovascular risk. Another study by Al'Aref et al. (2020) showcased the use of deep learning in coronary artery disease assessment, demonstrating high diagnostic accuracy. These studies reinforce the potential of deep learning in healthcare applications, particularly in heart disease classification.

3. Methodology

This project comprises four steps to determine whether we can predict whether a person will suffer from a heart disease. The steps are described in the following chart:


Step 1
1 Choosing Data Set
Data Understanding
2 Exploratory data analysis (EDA)
3 Data Preprocessing
(data cleansing, handle nulls, dimension reduction, balance data, normalization, feature engineering)
Step 2
4 Data Splitting
Construct balanced DS to improve prediction
Dimensionally Reduction (forward selection & PCA)
5 Implementing Basic ML models
1. KNN
2. Logistic Regression (GLM)
3. Logistic Regression
4. Decision Tree classifier
5. Boosting
6. Random forest
7. SVM
6 Evaluate Results
Step 3
7 Deep Learning NN (ANN)
ANN with default hyper params
8 Compare Results to ML (6)
9 Manipulating Hyper params
Hidden layers, Epochs, Batch size and more
10 Improve Results by DS modifications
Architecture improvements
11 Worsen the Results by DS modifications
12 Improve Network architecture
13 Choosing new metrics score and evaluate over training
14 Changing train/validation/test balance
15 Dimension Reduction PCA
Step 4
16 Discussion & Conclusions
17 Future Work

The following subsection will describe each step and discuss the results.

3.1. Step 1 - Data Understanding and Processing

3.1.1.Choosing a data set - data set description

3.1.1.1.The labeled heart disease dataset used in this study is sourced from Kaggle and contains multiple health indicators associated with heart disease risk. It includes demographic features (age, gender), physiological parameters (blood pressure, cholesterol, BMI), behavioral factors (smoking, alcohol consumption, exercise habits), and metabolic indicators (fasting blood sugar, triglyceride level, CRP level, homocysteine level). The target variable is "Heart Disease Status," which is labeled as "Yes" or "No."

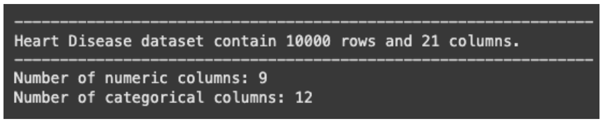
3.1.1.2.Data Set Analysis - to start analyzing the dataset, we have to check the values of our features. First of all - our dataset contains both categorical and numeric values. For example, Exercise Habits values are [high, medium, low], Gender [Male, Female], while Blood pressure and BMI are numeric float numbers.

3.1.1.3.In the dataset, we found out that some values, such as “Alcohol consumption,” contain “None”. We need to read the CSV files into a data frame in such a way that those None values will be regarded as None and not NaN, or Null - the way to do this is using this code:

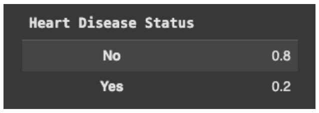
3.1.2.Exploratory Data Analysis (EDA)

3.1.2.1.Statistics: shape, numeric & categorical distribution

3.1.2.1.1.The purpose here is to understand the fundamental characteristics of each feature (including measures of central dependency and spread) - that will help identify potential anomalies or skew in both numeric and categorical variables.



```
-----------------------------------------------------------------
Heart Disease dataset contain 10000 rows and 21 columns.
-----------------------------------------------------------------
Number of numeric columns: 9
Number of categorical columns: 12
```



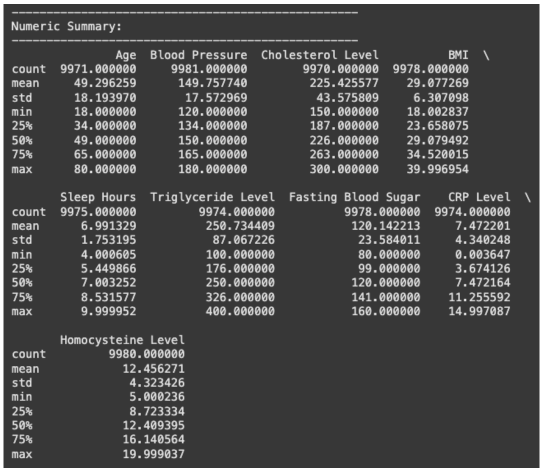
Heart Disease Status	
No	0.8
Yes	0.2

3.1.2.1.2. Shape and features & Target variable balance: we see imbalance in the target variable - 80% are labeled false (no heart disease), 20% true.

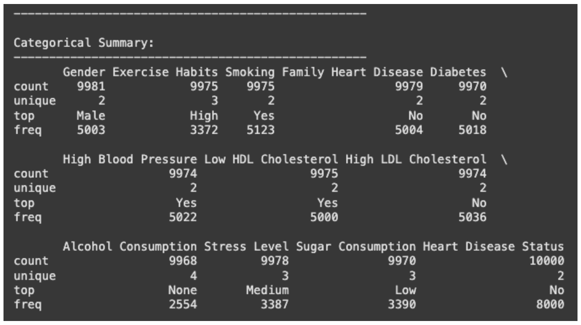
3.1.2.1.3. Numeric features statistics - using the describe function, we retrieved statistics about the dataset's numeric values (age, blood pressure, cholesterol level, etc.)

3.1.2.1.4. Categorial features statistics - we did the same for categorical features.

3.1.2.1.5. Summary and conclusions from dataset analysis: We can see that many categorical features are balanced around 50% besides the target variable. We can see that some fields are missing (do not sum up to 10000 rows) we need to take care of those rows. Interestingly, the mean of age is around 50. We have a maximum age of 80 even. The mean values of all numeric values have valuable information for this research.



```
--------------------------------------------------
Numeric Summary:
--------------------------------------------------
               Age  Blood Pressure  Cholesterol Level          BMI  \
count  9971.000000     9981.000000        9970.000000  9978.000000   
mean     49.296259      149.757740         225.425577    29.077269   
std      18.193970       17.572969          43.575809     6.307098   
min      18.000000      120.000000         150.000000    18.002837   
25%      34.000000      134.000000         187.000000    23.658075   
50%      49.000000      150.000000         226.000000    29.079492   
75%      65.000000      165.000000         263.000000    34.520015   
max      80.000000      180.000000         300.000000    39.996954   

       Sleep Hours  Triglyceride Level  Fasting Blood Sugar    CRP Level  \
count  9975.000000         9974.000000          9978.000000  9974.000000   
mean      6.991329          250.734409           120.142213     7.472201   
std       1.753195           87.067226            23.584011     4.340248   
min       4.000605          100.000000            80.000000     0.003647   
25%       5.449866          176.000000            99.000000     3.674126   
50%       7.003252          250.000000           120.000000     7.472164   
75%       8.531577          326.000000           141.000000    11.255592   
max       9.999952          400.000000           160.000000    14.997087   

       Homocysteine Level  
count         9980.000000  
mean            12.456271  
std              4.323426  
min              5.000236  
25%              8.723334  
50%             12.409395  
75%             16.140564  
max             19.999037  
```

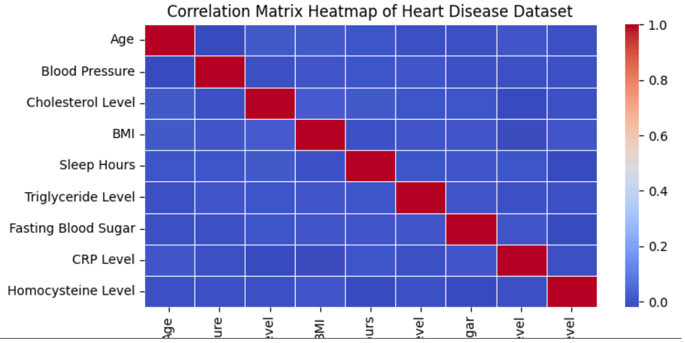


```
--------------------------------------------------

Categorical Summary:
--------------------------------------------------
       Gender Exercise Habits Smoking Family Heart Disease Diabetes  \
count    9981            9975    9975                 9979     9970   
unique      2               3       2                    2        2   
top      Male            High     Yes                   No       No   
freq     5003            3372    5123                 5004     5018   

       High Blood Pressure Low HDL Cholesterol High LDL Cholesterol  \
count                 9974                9975                 9974   
unique                   2                   2                    2   
top                    Yes                 Yes                   No   
freq                  5022                5000                 5036   

       Alcohol Consumption Stress Level Sugar Consumption Heart Disease Status
count                 9968         9978              9970                10000
unique                   4            3                 3                    2
top                   None       Medium               Low                   No
freq                  2554         3387              3390                 8000
```

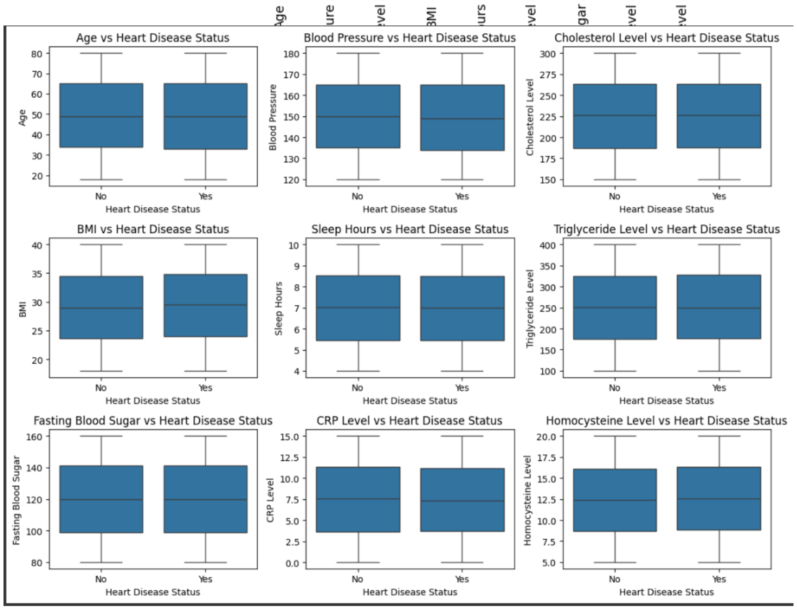
3.1.2.1.6. **Correlations**: numeric vs. numeric - predictor vs. target

3.1.2.1.6.1. In this section, we examine how numeric predictors relate to each other and the target variable, indicating potential multicollinearity or strong relationships that can inform further analysis or feature engineering. We must determine whether categorical variables are significantly associated with heart disease status, providing a statistical basis for deciding if these features may be helpful to the predictors.

3.1.2.1.6.2. Findings: the graph shows that cross correlations between numeric features are close to 0. That means the multicollinearity is zero. There is no overlapping.



3.1.2.1.6.3. Checking correlation between numeric features and the target variable Heart disease status. We will draw boxplot graphs of each numeric value and its target variable.

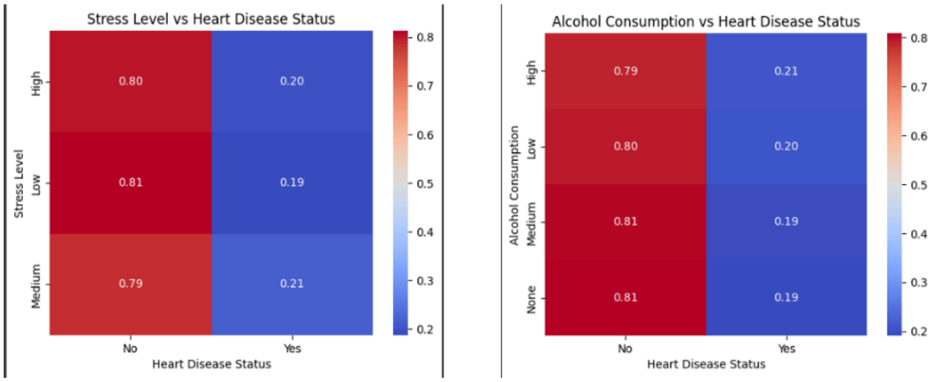


3.1.2.1.6.4. Findings: As seen from the charts, there are no correlations.

3.1.2.1.6.5.Correlations: Categorical columns vs. Heart Disease Status - using (Chi-squared)

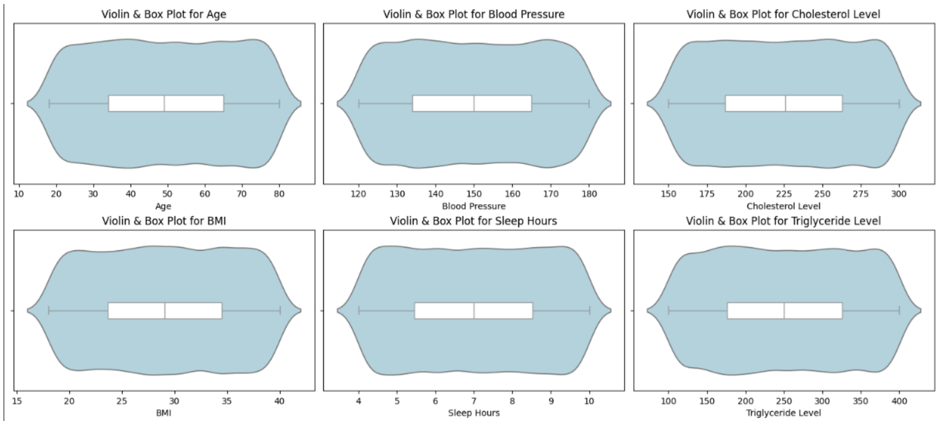
3.1.2.1.6.6.We examine a table of each predictor and its chi-square value and p-value. From the table, we can see that our Ho hypothesis (no difference or correlation between each categorical variable and the target variable) is rejected for stress level, which means that stress level is the only categorical variable that influences the target variable.

3.1.2.1.6.7. We plot the stress level chi-squared and the target variable and get the following:



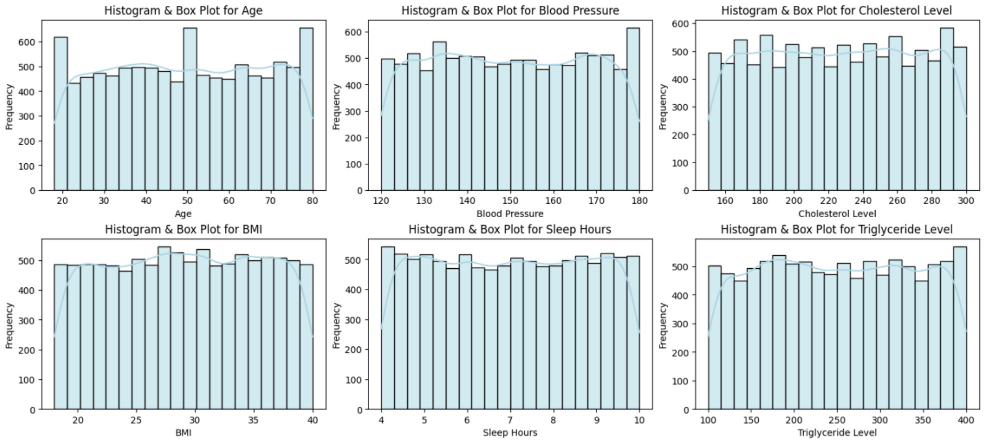
3.1.2.1.6.8. **Conclusion**: the correlation between different stress levels [High, Medium, Low] and the target variable Heart disease is relatively low (for all types ~20%), the same goes for Alcohol consumption.

3.1.2.2.Features distribution visualization: Violin, Boxplots - to visually inspect the distribution, spread and potential outliers of each numeric feature, providing an intuitive understanding of each variable's behaviour and detecting anomalies or skewed data.



3.1.2.2.1. The graph (full graph can be found in the colab notebook attached to this project) shows that all violins are similar in different features, showing the spread of data and outliers. If we take age, for example, the median is around 50, and the interquartile range (box) is between 35 and 65. We do not see potential outliers (dots outside the whiskers). The kernel density (the violin part) shape is more or less evenly spread. We do not see a normal distribution.

3.1.2.3. Features distribution - Histplots - it shows the frequency of individuals in different age ranges, indicating how age is distributed within the dataset. We can see here that data is evenly stread between different ranges.



3.1.3. Data Quality and Preparations: (Conclusions and actions from EDA process)

3.1.3.1. **Missing Values**: Both numeric and categorical predictors have missing values in varying amounts. Missingness must be addressed through imputation (mean/median for numeric, mode or "Missing" category for categorical) or exclusion, depending on the severity and distribution of missing data. Example: `Alcohol Consumption` has significant missing data (2,586 missing values out of 10,000), which could bias the analysis if not handled carefully.

3.1.3.2. **Outliers and Scaling**: Some numeric predictors, such as `CRP Level` and `Triglyceride Level`, exhibit wide ranges or extreme values that may act as outliers. These should be analyzed and, if needed, transformed or capped.

3.1.3.3.**Standardization or normalization** of numeric predictors will be necessary to ensure consistent scaling across features, especially for algorithms sensitive to magnitude differences (e.g., logistic regression)

3.1.3.4. Class Imbalance - The target variable, `Heart Disease Status`, has a significant imbalance (`No: 80%, Yes: 20%`). This imbalance must be addressed using techniques like resampling (oversampling or undersampling) or cost-sensitive learning.

3.1.3.5. Predictor Insights and Potential Relationships with the target:

3.1.3.5.1.**Age:** Older age is likely positively correlated with heart disease risk.

3.1.3.5.2.**Blood Pressure, Cholesterol Level, BMI:** These are known risk factors for cardiovascular health and show a wide range of values that could differentiate between individuals with and without heart disease.

3.1.3.5.3.**CRP Level and Triglycerides:** Indicators of inflammation and lipid levels, respectively, are likely significant predictors.

3.1.3.5.4.**Smoking, Exercise Habits, and Family Heart Disease:** Strongly tied to lifestyle and genetic risk factors, these are highly likely to predict heart disease.

3.1.3.5.5.**High Blood Pressure, Diabetes, and Cholesterol Levels (HDL/LDL):** Directly linked to cardiovascular health, these are expected to be important predictors.

3.1.3.5.6.Predictors like `Stress Level`, `Sugar Consumption`, and `Alcohol Consumption` may indirectly contribute to heart disease risk and offer actionable insights for prevention.

3.1.3.6. **Range & Distribution**: Predictors like `Sleep Hours` and `Fasting Blood Sugar` may require further analysis to understand their exact influence, as they seem to have narrower ranges and may exhibit non-linear effects.

3.1.3.7. Features Engineering and Selection:

3.1.3.7.1.Transformations - Numeric predictors with non-linear relationships to the target (e.g., `Sleep Hours`, `BMI`) may require feature transformations (e.g., polynomial terms, logarithmic transformations). Ordinal categorical predictors (e.g., `Stress Level`, `Sugar Consumption`) should be encoded with order preserved to capture their progression.

3.1.3.7.2.Interactions - Interaction terms between predictors (e.g., `Age` × `Exercise Habits` or `BMI` × `Smoking`) might reveal additional insights and improve predictive performance.

3.1.3.8. **Key Hypotheses:**

3.1.3.8.1.Older individuals with higher blood pressure, cholesterol levels, and CRP levels are at a higher risk of heart disease.

3.1.3.8.2.Smoking, low exercise, and high stress levels are likely to have strong associations with heart disease.

3.1.3.8.3.Gender and family history of heart disease may introduce inherent differences in risk profiles.

3.1.3.8.4.Lifestyle choices (e.g., alcohol consumption, sugar intake) influence heart disease risk, though they may be weaker predictors than clinical or genetic factors.

3.1.4.**Data Pre-Processing** (see exact methods in the colab notebook).

3.1.4.1. In the third section of step 1, based on the conclusion of EDA, we determined the extent and pattern of missing data across features. We assessed potential biases in the target variable. This guides whether dropping missing values or applying imputation techniques is most appropriate.

3.1.4.2. We prepare categorical variables for machine learning algorithms by converting them into suitable numeric representations, facilitating proper model training and improved predictive performance.

3.1.4.3.Handling Empty cells & Nulls (Drop and Imputation)

3.1.4.4.We examined the null distribution vs the target variable and dropping nulls.

3.1.4.5.We did not need to do any data imputation procedures.

3.1.4.6. Data transformations - We used transformations to convert categorical variables into numerical representations using one-hot encoding, ensuring compatibility with machine learning models. The drop_first=True parameter prevents redundancy and multicollinearity while converting all values to float and maintaining data consistency for efficient computation. This step enhances model interpretability and performance by making the dataset fully numeric. We implemented another way to transform categorical features to numeric values using dummies.

3.2.Step 2 - Implementing Basic ML models

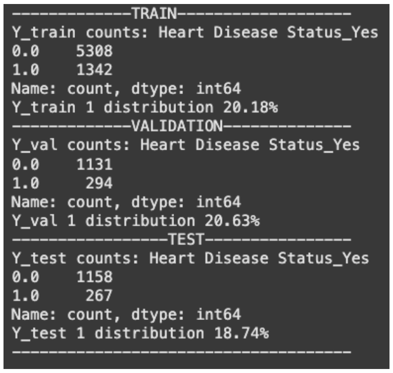
3.2.1.Data Splitting (Training, Validation, Test)

3.2.1.1. Splitting data into training, validation, and test sets ensures a robust machine learning model. The training set is used to learn patterns, the validation set helps tune hyperparameters and prevent overfitting, and the test set provides an unbiased evaluation of final model performance.

This approach ensures generalization to unseen data and avoids misleading performance estimates.

3.2.1.2. We split the dataset into 70% training, 15% test, and validation = 15%.

3.2.1.3. After dataset cleaning and splitting, we get the following: (note that the ratio of labeled classes is kept for training, validation, and testing ~20%



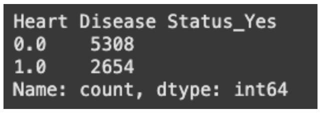
```
-------------TRAIN--------------------
Y_train counts: Heart Disease Status_Yes
0.0    5308
1.0    1342
Name: count, dtype: int64
Y_train 1 distribution 20.18%
-------------VALIDATION--------------
Y_val counts: Heart Disease Status_Yes
0.0    1131
1.0     294
Name: count, dtype: int64
Y_val 1 distribution 20.63%
----------------TEST----------------
Y_test counts: Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
Y_test 1 distribution 18.74%
-------------------------------------
```

3.2.2.Construct balanced DS to improve predictions.

3.2.2.1. The target variable “Heart Disease Status” is skewed: 80% of the samples are labeled “No,” while 20% are labeled “Yes.” (in the total dataset) This indicates that the dataset is imbalanced, with negative cases (no heart disease) significantly outnumbering positive cases (heart disease). Creating a more balanced alternative dataset through techniques such as oversampling is necessary to improve model performance and prevent bias toward the majority class.

3.2.2.2. We had created a function resample_data that uses SMOTE (Synthetic Minority Over-sampling technique) to generate synthetic data points for the minority class, thus balancing the classes in the training data. The function allows the user to specify the level of oversampling (default to 50% oversampling of the minority class).

3.2.2.3. After resampling we get the following balance:

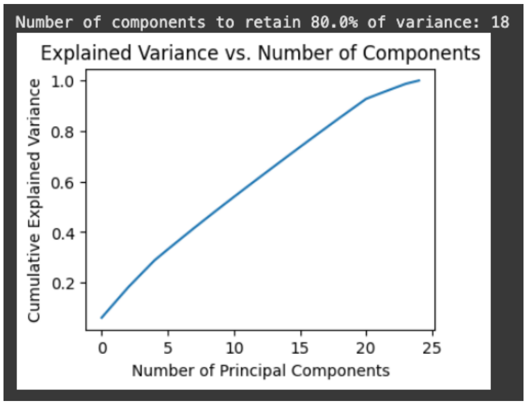


```
Heart Disease Status_Yes
0.0    5308
1.0    2654
Name: count, dtype: int64
```

3.2.3.**Dimensional Reduction**: Forward selection & Principal Component Analysis (PCA)

3.2.3.1. Streamline the feature set to enhance model performance by selecting the <u>most relevant predictors</u> and <u>reducing redundancy</u>. Forward selection incrementally adds the <u>most significant features</u> to improve predictive power, while PCA transforms features into uncorrelated components, capturing maximum variance with fewer dimensions. This approach helps mitigate overfitting, improves interpretability, and optimizes computational efficiency.

3.2.3.2. After implementing PCA, we get the following graph - # of principal components vs. cumulative explained variance.



```
Number of components to retain 80.0% of variance: 18
```

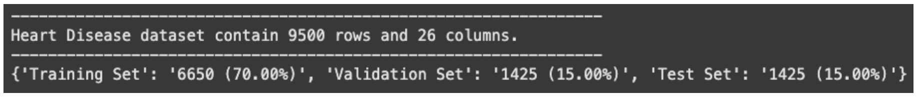
3.2.3.3. We can see that at about 20 components, we get a total variance of about 90%

3.2.3.4.

3.2.3.5.

3.2.4.Implementing Basic ML Models

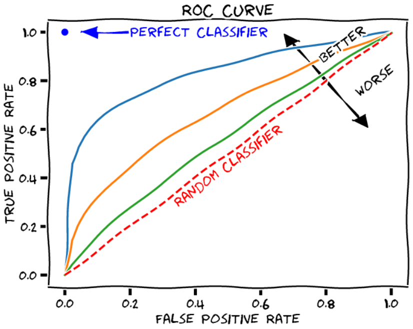
3.2.4.1. We had created a fit function for each type of model. We split the dataset into 70% training, 15% validation, and 15% test. For each model we created and predicted on the test set



```
--------------------------------------------------------------
Heart Disease dataset contain 9500 rows and 26 columns.
--------------------------------------------------------------
{'Training Set': '6650 (70.00%)', 'Validation Set': '1425 (15.00%)', 'Test Set': '1425 (15.00%)'}
```

we printed the following: Target variable - balance, confusion matrix, performance values of Accuracy, Precision, Recall, F1, AUC-ROC score and a graph of ROC curve vs. FP rate.

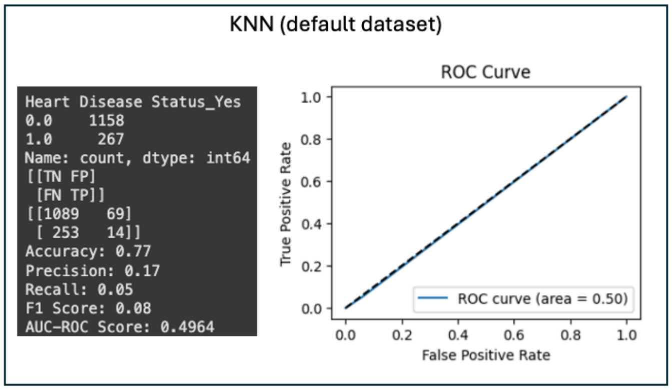
3.2.4.2.A note about ROC graph - in each model, we created ROC graph, to compare our results with ones that we aim to achieve, we use this reference:



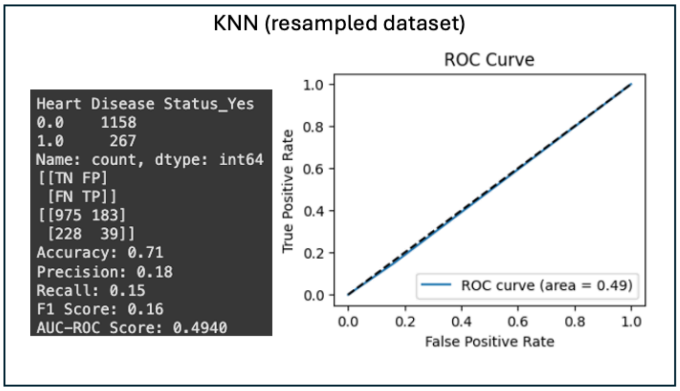
3.2.4.3.We compared the results for each model against the **default** and **resampled datasets**.

3.2.4.4.KNN

3.2.4.4.1. KNN with default dataset vs. KNN with resampled dataset


KNN (default dataset)

```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[1089   69]
 [ 253   14]]
Accuracy: 0.77
Precision: 0.17
Recall: 0.05
F1 Score: 0.08
AUC-ROC Score: 0.4964
```


KNN (resampled dataset)

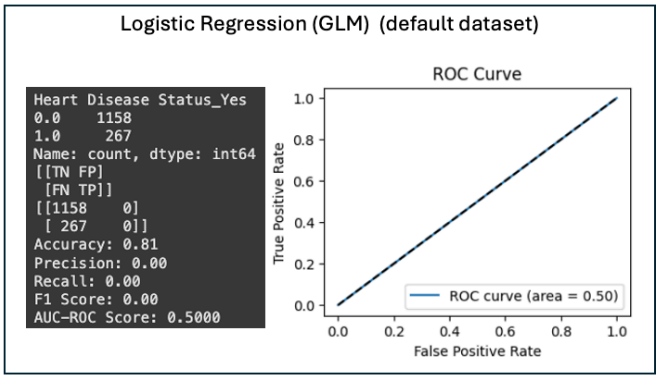
```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[975 183]
 [228  39]]
Accuracy: 0.71
Precision: 0.18
Recall: 0.15
F1 Score: 0.16
AUC-ROC Score: 0.4940
```

3.2.4.4.2.Comparing the confusion matrix for KNN with the default dataset, which has 69 FP, 253 FN - compared with 183 FP and 228 FN with the resampled dataset. The accuracy is better in the default dataset, the precision is the same, but the recall is better with resampled data. The F1 score is better with resampled data. The AUC-ROC is the same.

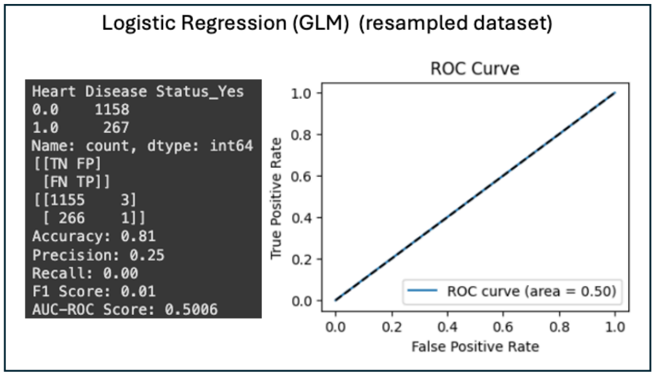
3.2.4.4.3. **To conclude** - we do not see a significant change between the two models in the KNN.

3.2.4.5.Logistic Regression (GLM)

3.2.4.5.1. Logistic Regression (GLM) with default dataset vs. with resampled dataset


Logistic Regression (GLM) (default dataset)

```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[1158    0]
 [ 267    0]]
Accuracy: 0.81
Precision: 0.00
Recall: 0.00
F1 Score: 0.00
AUC-ROC Score: 0.5000
```


Logistic Regression (GLM) (resampled dataset)

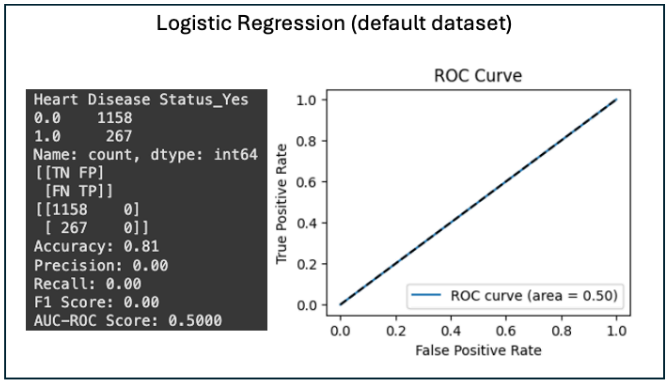
```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[1155    3]
 [ 266    1]]
Accuracy: 0.81
Precision: 0.25
Recall: 0.00
F1 Score: 0.01
AUC-ROC Score: 0.5006
```

3.2.4.5.2.Comparing confusion matrix logistic regression (GLM) - we see here very poor results. When using the default dataset, TP is 0, that means the model cannot predict positive results. The FP rate also goes down to zero. When using the resampled data - those figures are also very close to zero compared to the total dataset.

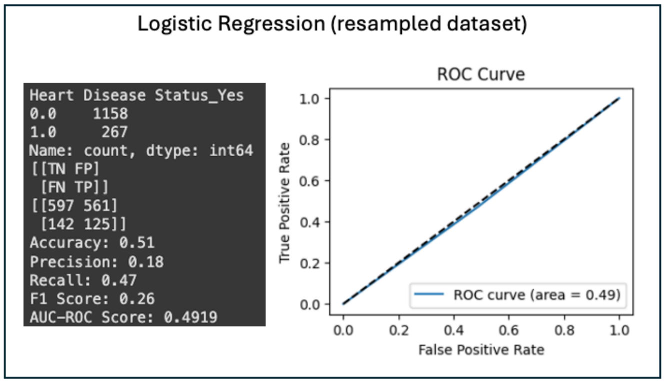
3.2.4.5.3.To conclude - Logistic regression (GLM) yields bad results.

3.2.4.6.**Logistic Regression**

3.2.4.6.1. We are predicting using a logistic regression model with two datasets, the default and the resampled one.


Logistic Regression (default dataset)

```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[1158    0]
 [ 267    0]]
Accuracy: 0.81
Precision: 0.00
Recall: 0.00
F1 Score: 0.00
AUC-ROC Score: 0.5000
```


Logistic Regression (resampled dataset)

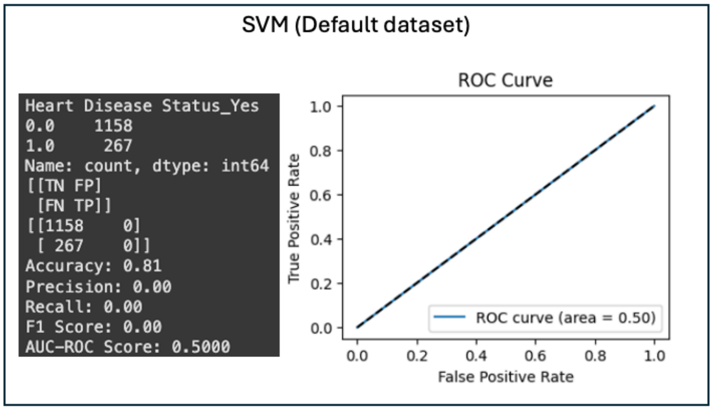
```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[597 561]
 [142 125]]
Accuracy: 0.51
Precision: 0.18
Recall: 0.47
F1 Score: 0.26
AUC-ROC Score: 0.4919
```

3.2.4.6.2. Comparing confusion matrix for logistic regression - using the default dataset, we can see that the ROC value is 0.5, which means random classification. The values of FP and TP are 0. Running the model with resampled dataset - yields better Recall - 0.47, precision 0.18, but still the ROC is 0.5.

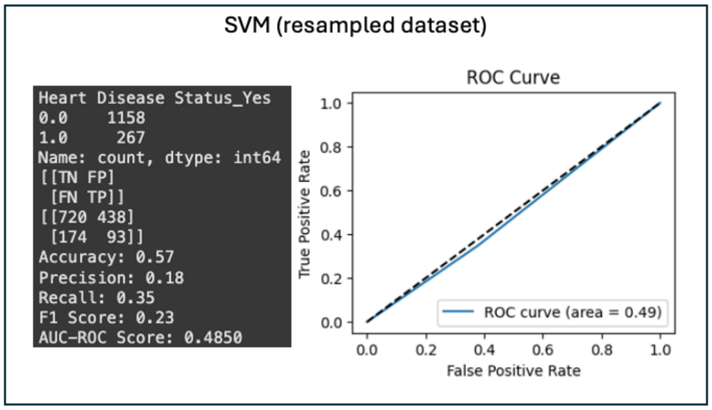
3.2.4.6.3. To conclude - although logistic regression with resampled data looks better from the Recall perspective, the ROC is still unsuitable for classification.

3.2.4.7.Decision Tree Classifier

3.2.4.7.1. Comparing the decision trees with two datasets, the default and the resampled one


SVM (Default dataset)

```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[1158    0]
 [ 267    0]]
Accuracy: 0.81
Precision: 0.00
Recall: 0.00
F1 Score: 0.00
AUC-ROC Score: 0.5000
```


SVM (resampled dataset)

```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[720 438]
 [174  93]]
Accuracy: 0.57
Precision: 0.18
Recall: 0.35
F1 Score: 0.23
AUC-ROC Score: 0.4850
```

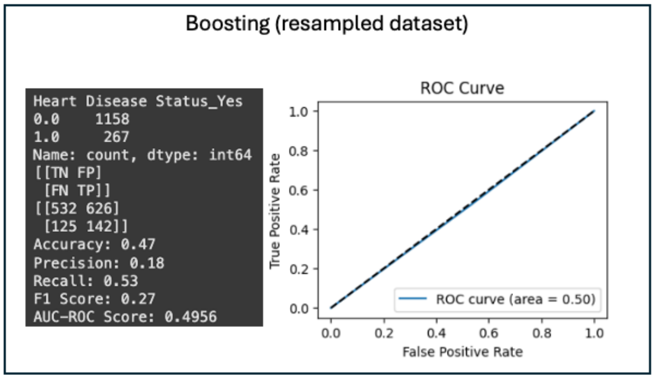
3.2.4.7.2. Comparing confusion matrix for SVM. Using the default dataset, we have zero in both FP and TP, yielding zero recall and precision, which is problematic. The ROC value when using the resampling dataset is better. The recall and precision in this case are better.

3.2.4.7.3. To conclude - SVM with sampling is better, but still not good enough.

3.2.4.8.Boosting (GradientBoosting Classifier)

3.2.4.8.1. We run this model only on the resampled dataset, the results are:

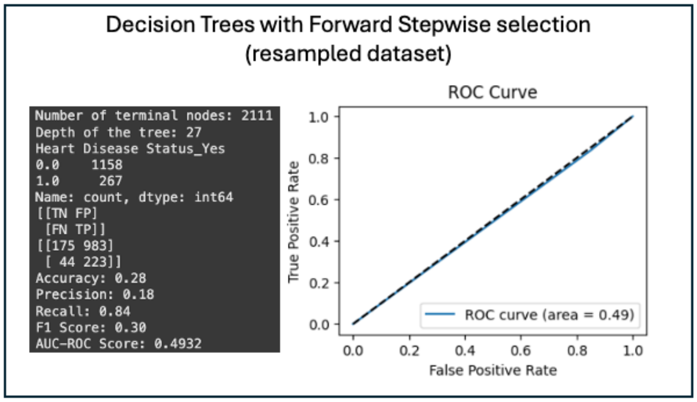
3.2.4.8.2.As we can see, the ROC value is very close to 0.5, the recall and F1 score went up. Precision is low.


Boosting (resampled dataset)

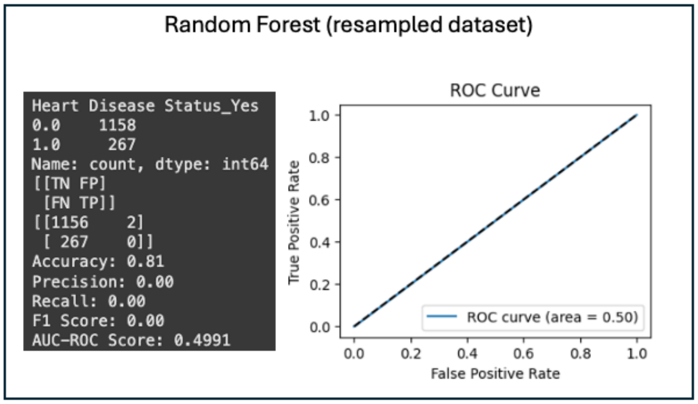
```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[532 626]
 [125 142]]
Accuracy: 0.47
Precision: 0.18
Recall: 0.53
F1 Score: 0.27
AUC-ROC Score: 0.4956
```

3.2.4.8.3. To conclude - this model does not fit for this classification problem.

3.2.4.9.Random Forest, Decision trees with forward stepwise selection, decision trees with pruning:

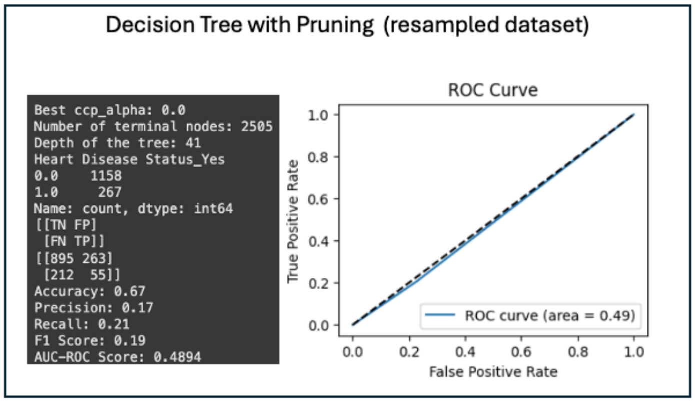

Decision Trees with Forward Stepwise selection (resampled dataset)

```
Number of terminal nodes: 2111
Depth of the tree: 27
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[175 983]
 [ 44 223]]
Accuracy: 0.28
Precision: 0.18
Recall: 0.84
F1 Score: 0.30
AUC-ROC Score: 0.4932
```


Random Forest (resampled dataset)

```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[1156    2]
 [ 267    0]]
Accuracy: 0.81
Precision: 0.00
Recall: 0.00
F1 Score: 0.00
AUC-ROC Score: 0.4991
```

3.2.4.9.1. In all three models, we can see that the ROC did not change and is ~0.5, indicating a random classification. The difference is between the metrics.


Decision Tree with Pruning (resampled dataset)

```
Best ccp_alpha: 0.0
Number of terminal nodes: 2505
Depth of the tree: 41
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[895 263]
 [212  55]]
Accuracy: 0.67
Precision: 0.17
Recall: 0.21
F1 Score: 0.19
AUC-ROC Score: 0.4894
```

3.3.Step 3 - Deep Learning Neural Network Model (ANN)

3.3.1. This step runs deep NN network with different architectures, default parameters, variable number of epochs, batch size, and hidden layers. We changed the dataset to improve and worsen results. In the architecture improvements, we changed the layers structure, the metrics scores, and the balance size (training vs. validation), and eventually we deduced the dataset dimensions using PCA.

3.3.2. Important note: Ensuring deterministic behavior is essential for reproducibility and consistency across runs. Setting os.environ['TF_DETERMINISTIC_OPS'] = '1' enforces deterministic TensorFlow operations, reducing hardware-related variations. Using HeNormal(seed=42) ensures stable weight initialization, preventing fluctuations in model performance due to randomness. This setup is crucial for controlled experiments, model comparisons, and debugging, allowing reliable evaluation of architectural and hyperparameter changes.

3.3.3. Creating different fitting functions: Simple, Complex, Advanced

3.3.3.1. For all functions, we compiled the model with default optimizer (Adam), loss=binary cross entropy, and metric=accuracy, we used different architectures for each function

3.3.3.2. **Simple**:

3.3.3.2.1. `def simple_ann_fit(X_train, Y_train, X_Val, Y_Val, X_test, Y_test, epochs=1):`

3.3.3.2.2. Architecture layers: 2 Dense layers, activation functions, ReLU and Sigmoid

3.3.3.3. **Complex**:

3.3.3.3.1. `def improved_ann_fit(X_train, Y_train, X_val, Y_val, X_test, Y_test, epochs=1, batch_size=32, metrics='accuracy', verbose=False,):`

3.3.3.3.2. We have 6 Dense layers, each with batch normalization and dropout rate of 0.2, all Dense layers have ReLU activation functions. The Last Dense layer has Sigmoid activation function.

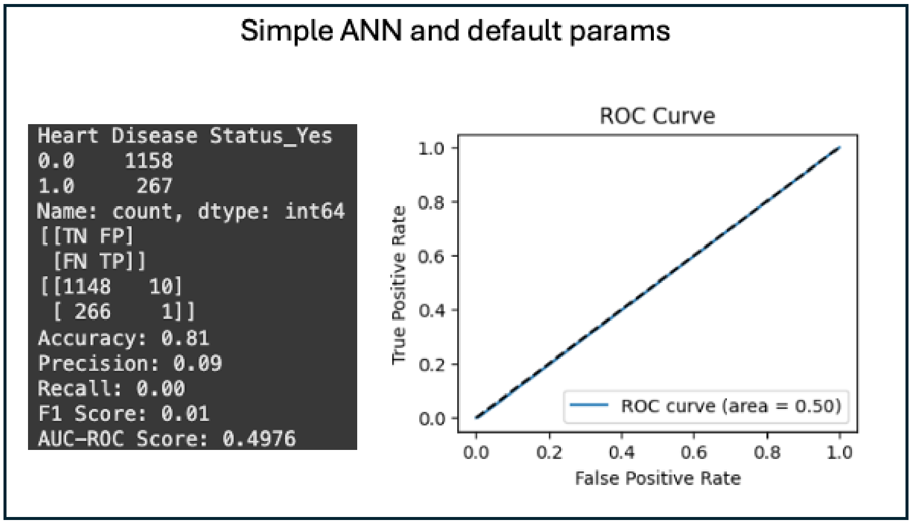
3.3.3.4. **Advanced**:

3.3.3.4.1. `def complex_ann_fit(X_train, Y_train, X_Val, Y_Val, X_test, Y_test, epochs=1, batch_size=32):`

3.3.3.4.2. 4 Dense layers, all but the last one with ReLU activation, last one has Sigmoid. Batch size=32.

3.3.4. Splitting data 70% training, 15% validation, 15% test

3.3.5. Running with simple ANN and default parameters:



Simple ANN and default params

```
Heart Disease Status_Yes
0.0    1158
1.0     267
Name: count, dtype: int64
[[TN FP]
 [FN TP]]
[[1148   10]
 [ 266    1]]
Accuracy: 0.81
Precision: 0.09
Recall: 0.00
F1 Score: 0.01
AUC-ROC Score: 0.4976
```

3.3.5.1. The ROC curve is not improving, and the precision and recall are very low.

3.3.5.2.This model does not work well for our predictions.

3.3.6. Running model with different hyperparams: adding hidden layers, changing number of epochs. We run with three different options:

3.3.6.1. Option 1 - 2 Dense layers, 20 epochs, 32 batch size.

3.3.6.2. Option 2 - 4 Dense layers, 20 epochs, 32 batch size.

3.3.6.3. Option 3 - 5 Dense layers, 20 epochs, 32 batch size

3.4.Aa

3.5.A

3.6.A

3.7.A

3.8.

4. Results

5. Discussion and Conclusions
6. Future Work
7. References

==

-

3.2 Data Preprocessing

- **Handling Missing Values:** Any missing or inconsistent data points were addressed using imputation techniques.
- **Feature Encoding:** Categorical variables (e.g., smoking, gender, stress levels) were converted into numerical representations using one-hot encoding.
- **Normalization:** Continuous variables (e.g., cholesterol levels, BMI) were normalized to ensure uniform feature scaling.
- **Train-Test Split:** The dataset was split into training (80%) and testing (20%) sets to evaluate model performance.

3.3 Model Architecture

A deep neural network (DNN) was constructed with the following configuration:

- **Input Layer:** Accepts all selected features.
- **Hidden Layers:** Three fully connected layers with ReLU activation functions.
- **Dropout Layer:** Applied to prevent overfitting.
- **Output Layer:** A single neuron with a sigmoid activation function for binary classification.
- **Loss Function:** Binary cross-entropy.
- **Optimizer:** Adam optimizer.

3.4 Model Training

The model was trained using backpropagation with stochastic gradient descent (SGD). Hyperparameter tuning was performed to optimize learning rate, batch size, and number of epochs.

4. Results

The performance of the deep learning model was assessed using:

- **Accuracy:** Measures the proportion of correctly classified instances.
- **Precision & Recall:** Evaluate model reliability in predicting positive cases.
- **F1-Score:** Provides a balance between precision and recall.
- **Confusion Matrix:** Visual representation of true positives, false positives, true negatives, and false negatives.

The model achieved high accuracy in distinguishing individuals with and without heart disease, outperforming traditional machine learning models. Feature importance analysis revealed that factors such as cholesterol levels, blood pressure, and smoking habits had the highest impact on predictions.

5. Discussion and Conclusions

This study demonstrates the potential of deep learning in predicting heart disease based on multiple health indicators. The proposed neural network model effectively learns patterns from the dataset and provides reliable classifications. The high performance of the model suggests that deep learning can serve as a valuable tool in clinical decision-making, potentially aiding healthcare professionals in early diagnosis and prevention strategies.

6. Future Work

While the proposed model achieves high accuracy, further research can be conducted to enhance its effectiveness. Future directions include:

- **Integrating Additional Clinical Data:** Incorporating medical imaging or genomic data may improve model performance.
- **Exploring Alternative Architectures:** Investigating convolutional neural networks (CNNs) or transformer-based models for enhanced feature extraction.
- **Improving Interpretability:** Implementing explainable AI (XAI) techniques to make the model's decisions more transparent to healthcare professionals.
- **Expanding the Dataset:** Utilizing larger and more diverse datasets to increase model generalizability.

7. References

- Rajkomar, A., et al. (2018). "Scalable and accurate deep learning with electronic health records." NPJ Digital Medicine, 1(1), 18.
- Esteva, A., et al. (2017). "Dermatologist-level classification of skin cancer with deep neural networks." Nature, 542(7639), 115-118.
- Krittanawong, C., et al. (2019). "Artificial intelligence in precision cardiovascular medicine." Journal of the American College of Cardiology, 74(10), 1307-1321.
- Al'Aref, S. J., et al. (2020). "Machine learning in cardiovascular medicine: are we there yet?" Journal of the American College of Cardiology, 75(5), 573-584.