Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

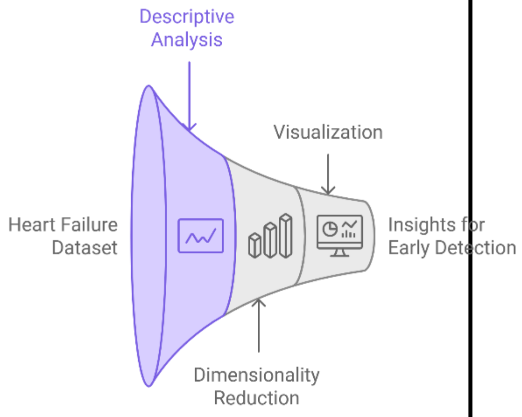
Introduction to Machine Learning

Project- Fall 2024

	Name	ID
1	Alaa yasser fathy bekhit	21P0408
2	Mohamed ahmed abdelraouf	2001038
3	Ahmed mohamed abdelmonem	2000052

Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

1.Data Exploration and visualization

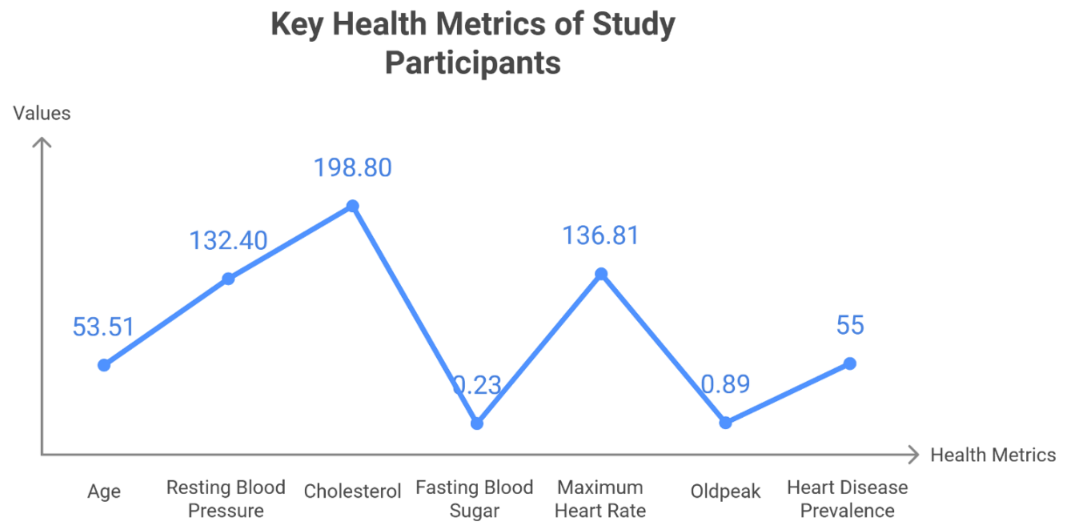


Descriptive Analysis: The summary statistics of the dataset's numerical features are displayed for your review.

Statistic	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
Count	918	918	918	918	918	918	918
Mean	53.51	132.40	198.80	0.23	136.81	0.89	0.55
Std Dev	9.43	18.51	109.38	0.42	25.46	1.07	0.50
Min	28	0	0	0	60.00	-2.60	0
25th %	47	120	173.25	0	120	0	0
50th %	54	130	223	0	138	0.60	1
75th %	60	140	267	0	156	1.50	1
Max	77	200	603	1	202	6.20	1

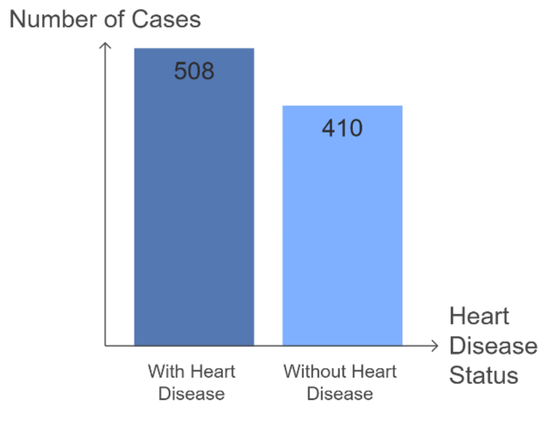
- **Age**: The average age of participants is approximately 53.51 years, with a minimum age of 28 and a maximum of 77 years.

- **Resting Blood Pressure (RestingBP)**: The mean resting blood pressure is 132.40 mmHg, indicating a generally healthy range, but with some individuals reaching up to 200 mmHg.
- **Cholesterol**: The average cholesterol level is 198.80 mg/dL, with a wide range from 0 to 603 mg/dL, suggesting significant variability in cholesterol levels among participants.
- **Fasting Blood Sugar (FastingBS)**: The fasting blood sugar levels show a binary distribution, with a mean of 0.23, indicating that most participants have a fasting blood sugar level below 120 mg/dL.
- **Maximum Heart Rate (MaxHR)**: The average maximum heart rate recorded is 136.81 bpm, with a maximum of 202 bpm, which is within a normal range for healthy individuals.
- **Oldpeak**: The average oldpeak value is 0.89, with a minimum of -2.60, indicating variability in exercise-induced ST depression.
- **Heart Disease**: The prevalence of heart disease in the dataset is approximately 55%, as indicated by the mean value of the HeartDisease variable.



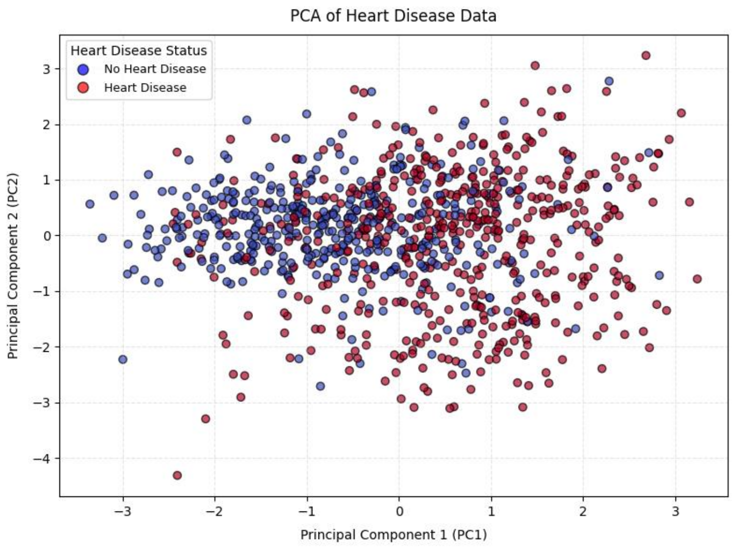
Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

1. Target Variable Distribution: The dataset contains 508 cases with heart disease and 410 without it.

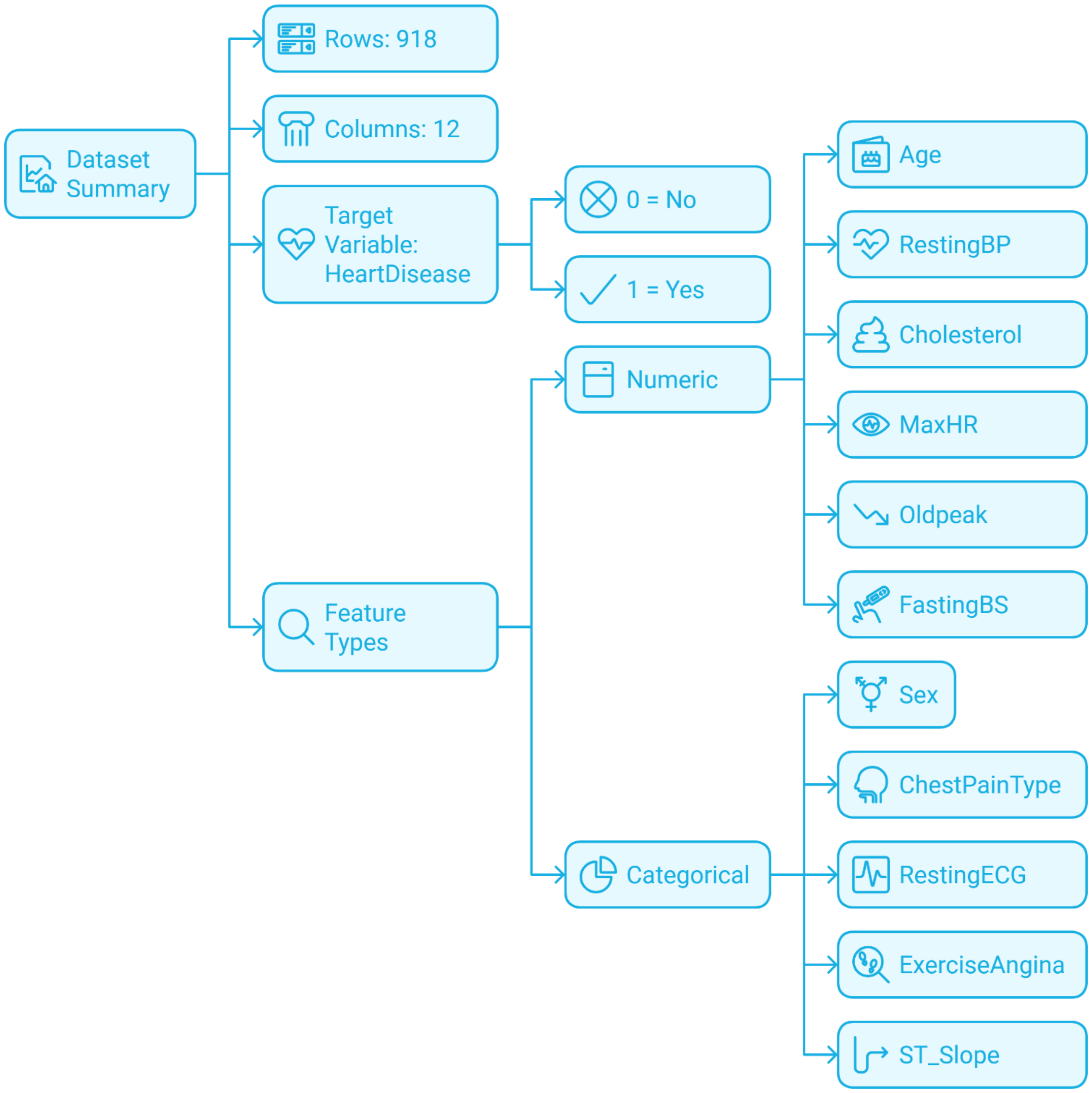


Distribution of Heart Disease Cases

2. Dimensionality Reduction via PCA: The dataset's numerical features were reduced to two dimensions using PCA. A scatter plot shows the data distribution with respect to the presence of heart disease.



Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

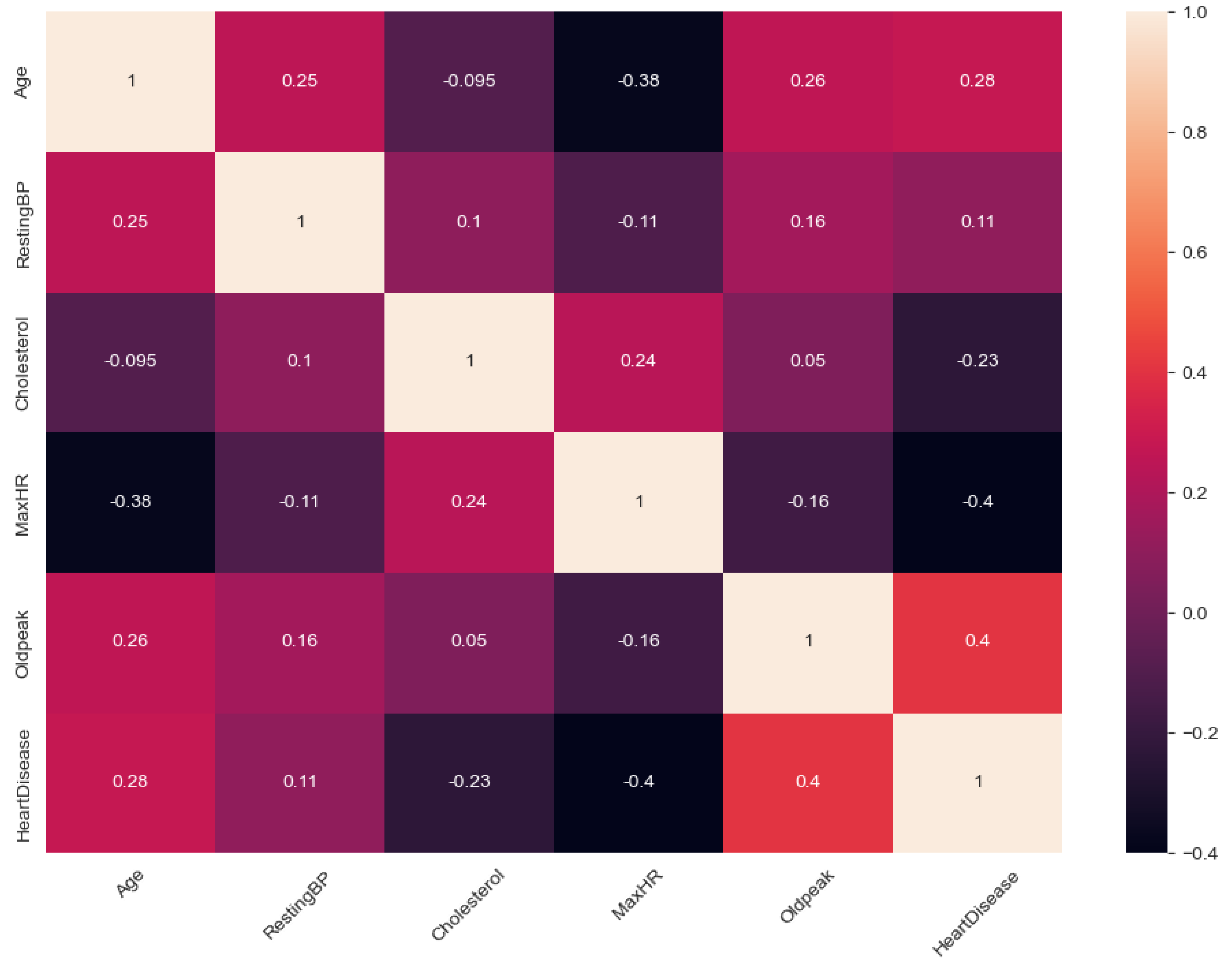


Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

Correlation Matrix

Correlation Values:

- The values range from **-1** (perfect negative correlation) to **1** (perfect positive correlation).
- A correlation close to 0 indicates little or no linear relationship between features.



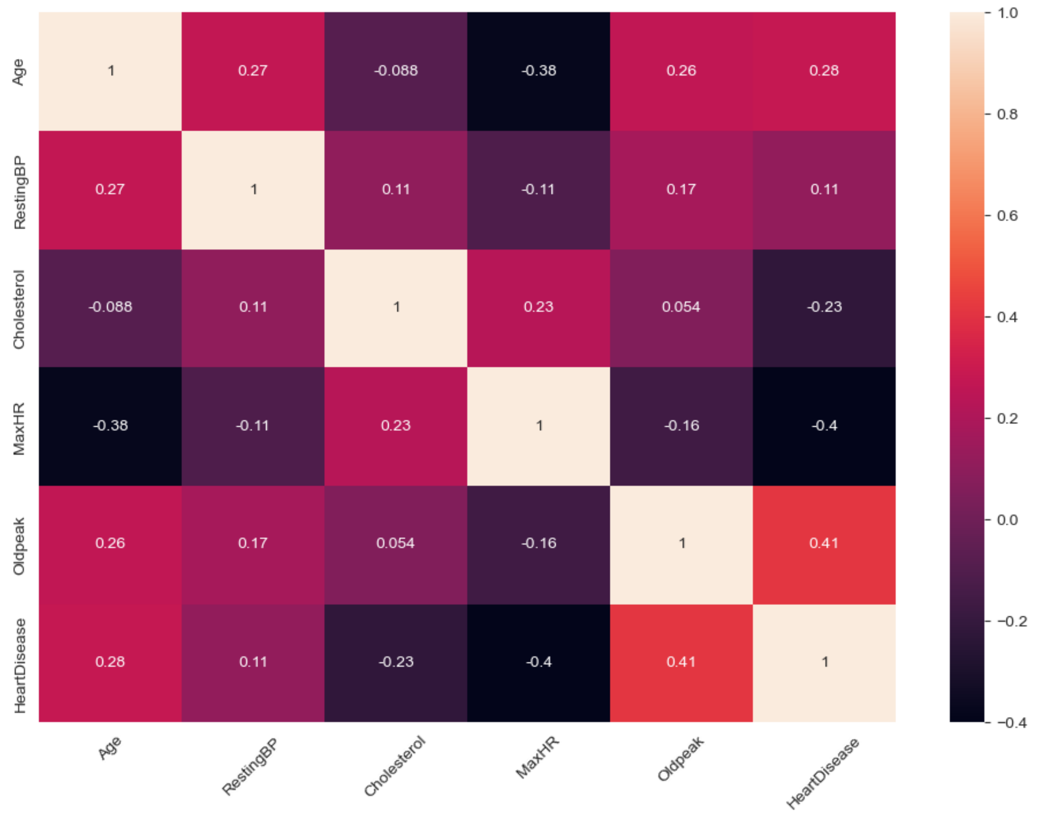
HeartDisease Correlations:

- **Oldpeak (0.4)**: The strongest positive correlation with HeartDisease.
- **MaxHR (-0.4)**: The strongest negative correlation with HeartDisease.
- **Age (0.28)**: A moderate positive correlation with HeartDisease.
- **RestingBP (0.11)** and **Cholesterol (-0.23)**: Weak correlations with HeartDisease.

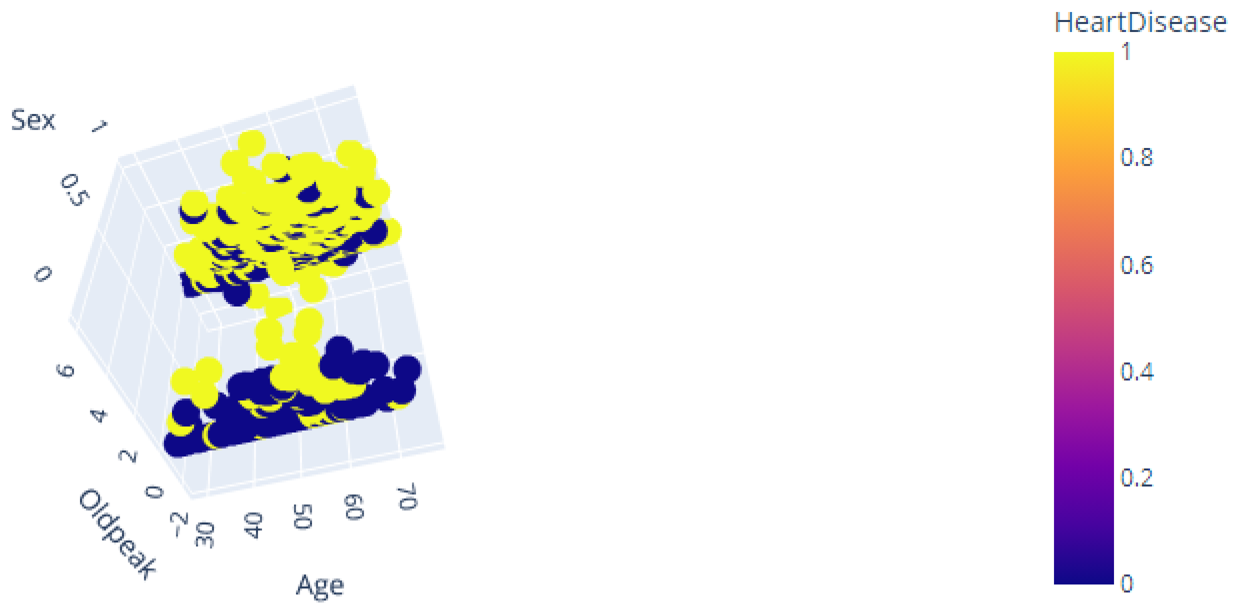
Feature Interdependencies:

- **Age and MaxHR (-0.38)**: Strong negative correlation
- **Oldpeak and Age (0.26**): Positive correlation

After preprocessing



3D scatter plot visualizes the relationship between oldpeak, Age, and Sex



Key Observations

1. Axes:
   - X-Axis (Oldpeak): Represents the ST depression
   - Y-Axis (Age) : form 30 to higher than 70.
   - Z-Axis (Sex): Encoded as 0 (female) and 1 (male).
2. Color Representation:
   - Yellow points represent individuals with heart disease (HeartDisease = 1).
   - Purple points represent individuals without heart disease (HeartDisease = 0)

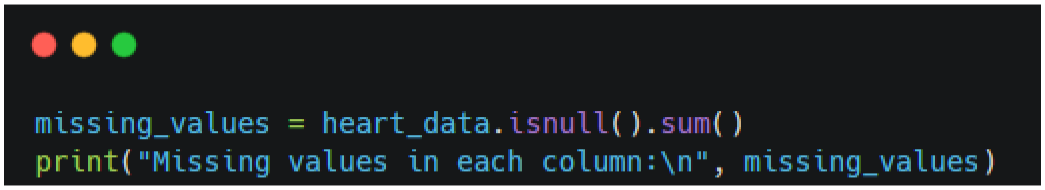
- Data points are distributed showing how these three features relate to the target variable (HeartDisease).
- Higher Oldpeak values are strongly associated with heart disease, consistent with its high correlation (0.4) from the heatmap analysis.
- Older individuals appear more likely to have heart disease, as seen by the clustering of yellow points at higher Age values.

2. Data cleaning and processing:

Step 1: Handling Missing Values

No missing values were detected in the dataset, so no imputation was necessary.

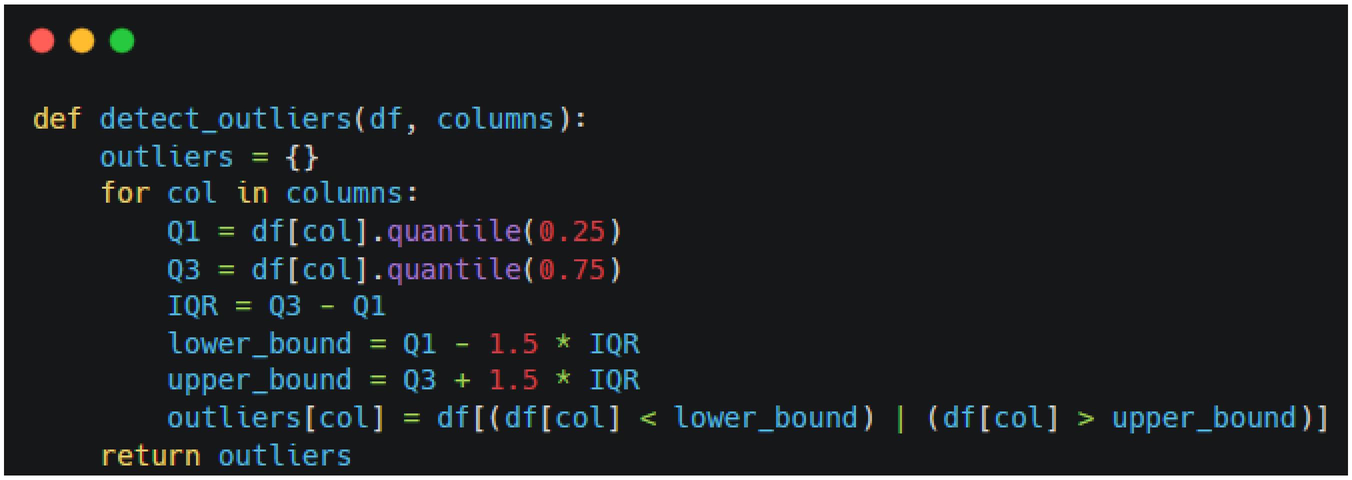
Feature	**Missing Values**
Age	0
Sex	0
ChestPainType	0
RestingBP	0
Cholesterol	0
FastingBS	0
RestingECG	0
MaxHR	0
ExerciseAngina	0
Oldpeak	0
ST_Slope	0
HeartDisease	0



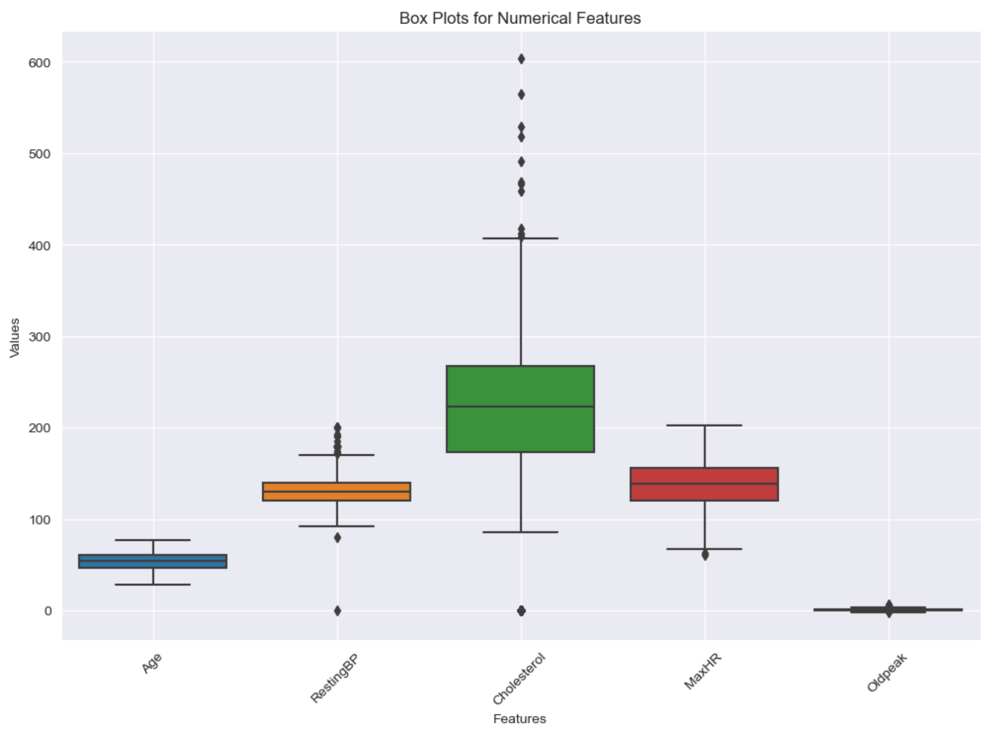
```
missing_values = heart_data.isnull().sum()
print("Missing values in each column:\n", missing_values)
```

Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

Step 2: Dealing with Outliers



```python
def detect_outliers(df, columns):
    outliers = {}
    for col in columns:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        outliers[col] = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
    return outliers
```



1. **Age (0 outliers):**

The absence of outliers indicates that the age distribution is well-contained within the IQR bounds.

2. **RestingBP (28 outliers):**

A significant number of outliers were detected in **RestingBP** (Resting Blood Pressure).

3. **Cholesterol (183 outliers):**

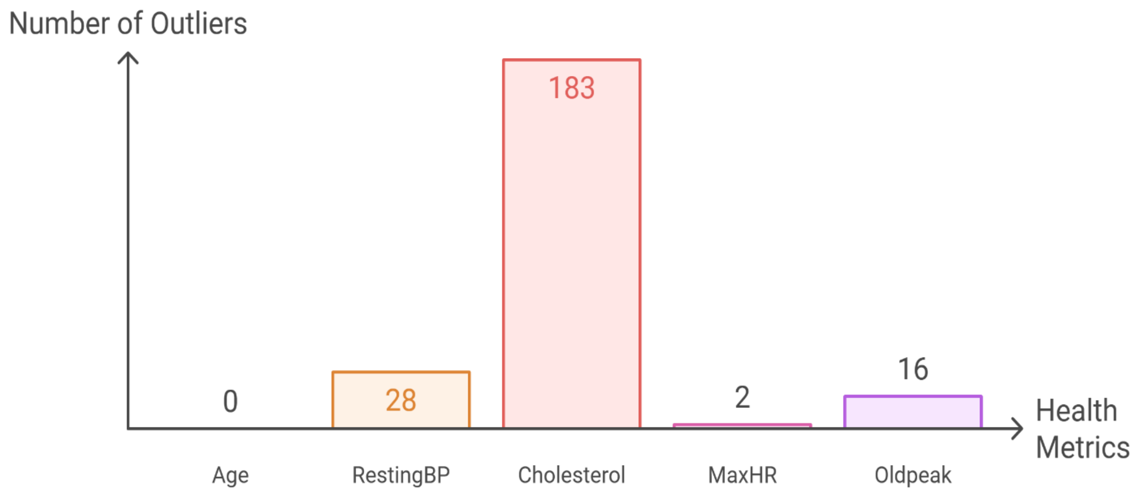
A very high number of outliers indicates that cholesterol levels in the dataset have a wide range, with many values falling outside the IQR.

4. **MaxHR (2 outliers):**

Few outliers indicate that most maximum heart rate values fall within the expected range.

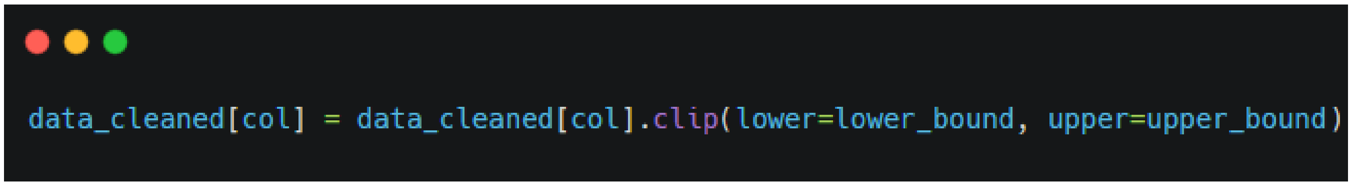
5. **Oldpeak (16 outliers):**

The presence of 16 outliers in **Oldpeak** suggests unusual values in ST depression during exercise.



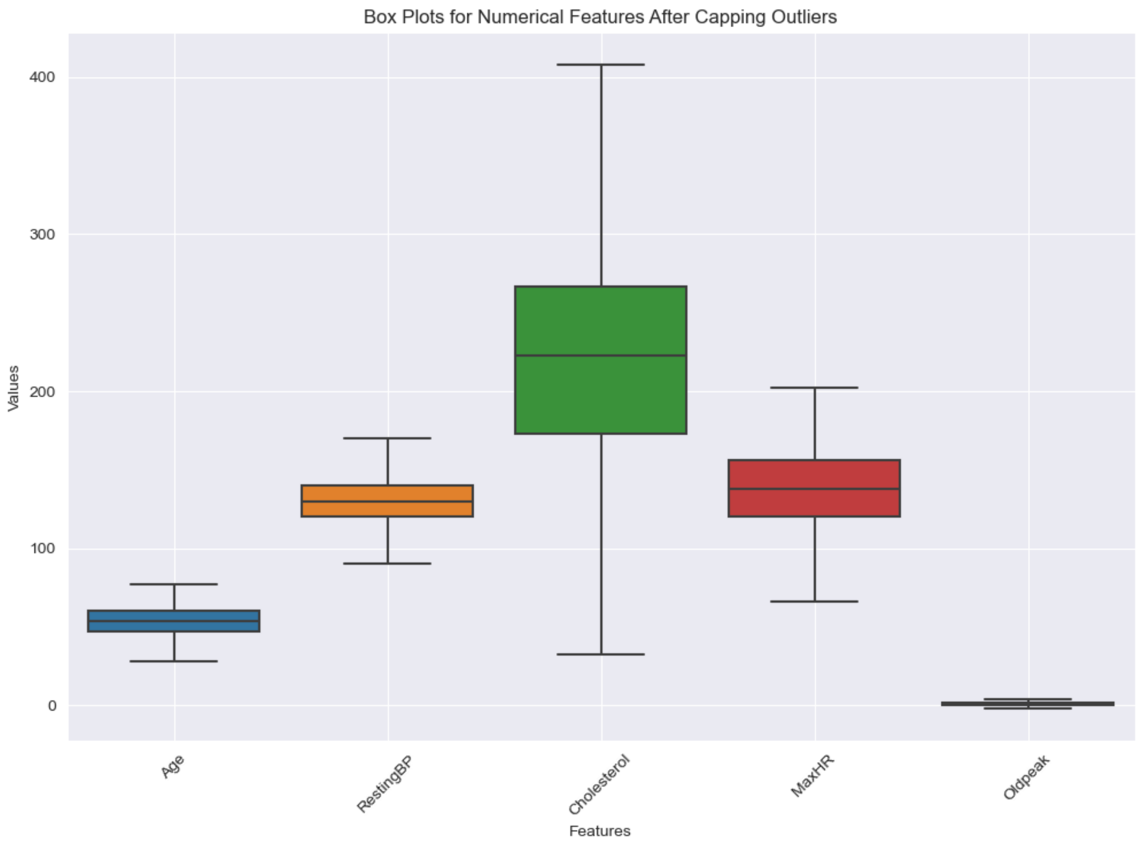
Outlier Counts in Health Metrics

We aims to handle outliers in the numerical features of a dataset by capping them at the upper and lower bounds based on the Interquartile Range (IQR). This helps reduce the influence of extreme values without completely removing data points.



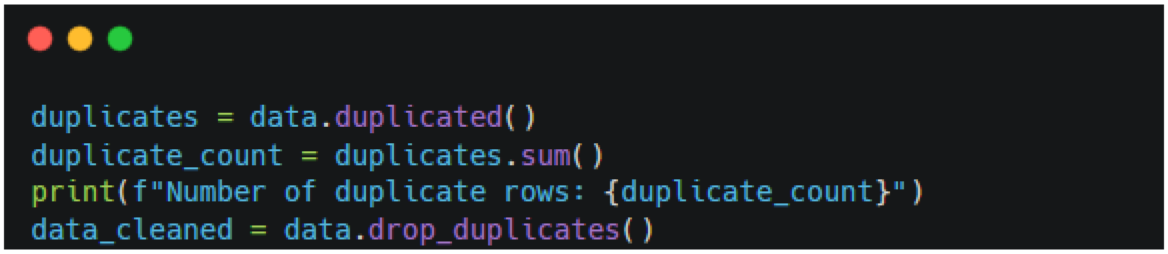
```
data_cleaned[col] = data_cleaned[col].clip(lower=lower_bound, upper=upper_bound)
```

The `clip` function is applied to limit values below the lower bound or above the upper bound. Any value outside these bounds is replaced by the corresponding boundary value.



Step 3:

Dealing with Duplicates

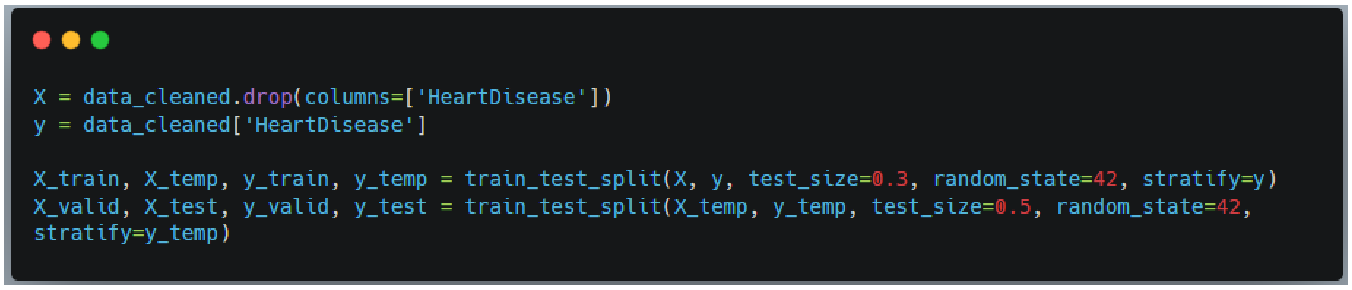


```
duplicates = data.duplicated()
duplicate_count = duplicates.sum()
print(f"Number of duplicate rows: {duplicate_count}")
data_cleaned = data.drop_duplicates()
```

Number of duplicate rows: 0, No duplicate rows were found.

Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

Step 4: Splitting Data into Training, Validation, and Testing Sets

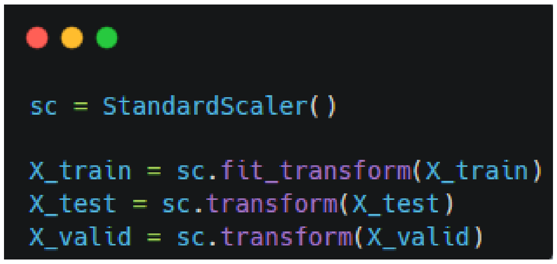


```python
X = data_cleaned.drop(columns=['HeartDisease'])
y = data_cleaned['HeartDisease']

X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)
X_valid, X_test, y_valid, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42,
stratify=y_temp)
```

Dataset	Proportion of Original Data	Purpose
Training	70%	Model training
Validation	15%	Hyperparameter tuning
Testing	15%	Final evaluation of the model

Scaling the Dataset

standardizes the feature values in the dataset using StandardScaler. Scaling is crucial for ensuring that features are treated equally and not biased due to differences in scale or units.



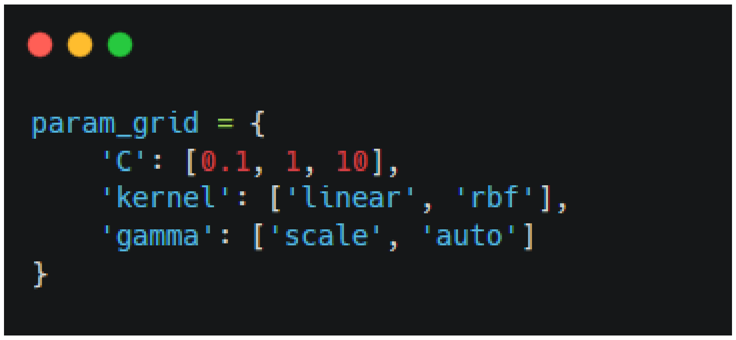
```python
sc = StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
X_valid = sc.transform(X_valid)
```

Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

1st model: GridSearchCV Hyperparameter Tuning with SVM

performs hyperparameter tuning for a Support Vector Machine (SVM) model using GridSearchCV. It optimizes the model's hyperparameters to maximize classification accuracy and evaluates performance on both the training and validation sets.

Define the Hyperparameter Grid



```
param_grid = {
    'C': [0.1, 1, 10],
    'kernel': ['linear', 'rbf'],
    'gamma': ['scale', 'auto']
}
```

C (Regularization Parameter):

- Controls the trade-off between achieving a low error on the training data and minimizing model complexity.
- Smaller values enforce stronger regularization, while larger values reduce regularization.

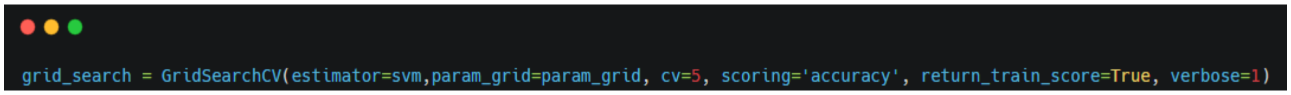
`kernel`:

- Specifies the type of kernel function to transform data into higher dimensions.
- **`Linear`** : Linear kernel for linear decision boundaries.
- **`Rbf`** : Radial Basis Function kernel for non-linear decision boundaries.

`gamma` (Kernel Coefficient):

- Determines the influence of individual data points.
- `scale`: Uses *1 /number of feature* as default.
- `auto`: Uses *1/ number of samples* as default.

Configure GridSearchCV



```
grid_search = GridSearchCV(estimator=svm,param_grid=param_grid, cv=5, scoring='accuracy', return_train_score=True, verbose=1)
```

`estimator= svm` : The SVM model to be optimized.

`param_grid` : Specifies the hyperparameter grid to search.

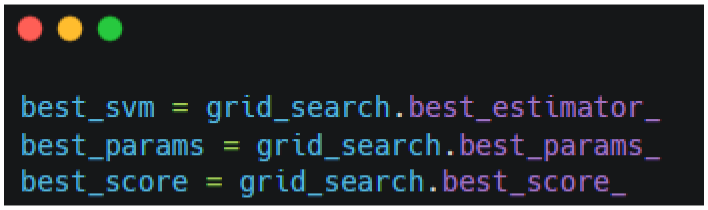
`cv=5`: Performs 5-fold cross-validation to evaluate each hyperparameter configuration.

`scoring='accuracy'`: Optimizes the model based on classification accuracy.

`return_train_score = True`: Stores the training scores for each configuration.

`verbose=1`: Displays progress messages during execution.

Extract Best Model and Hyperparameters



```
best_svm = grid_search.best_estimator_
best_params = grid_search.best_params_
best_score = grid_search.best_score_
```

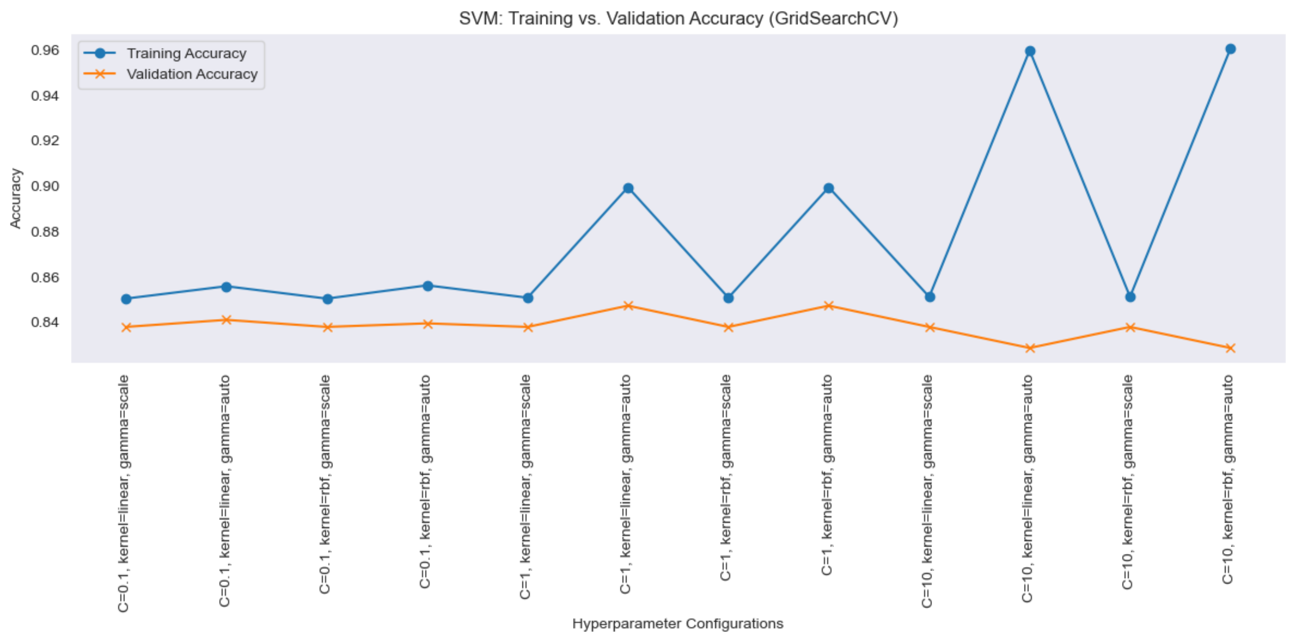
- `best_estimator_`: The SVM model with the best hyperparameters.
- `best_params_`: The hyperparameter combination that achieved the highest cross-validation accuracy.
- `best_score_`: The highest cross-validation accuracy achieved.

Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

Observations for Support Vector Machine(SVM:

1. Best hyperpararmeters of SVM:
   Best Hyperparameters: {'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}

2. **Grid search Training and Validation Accuracy of SVM:**
   Best Cross-Validation Score: 0.85



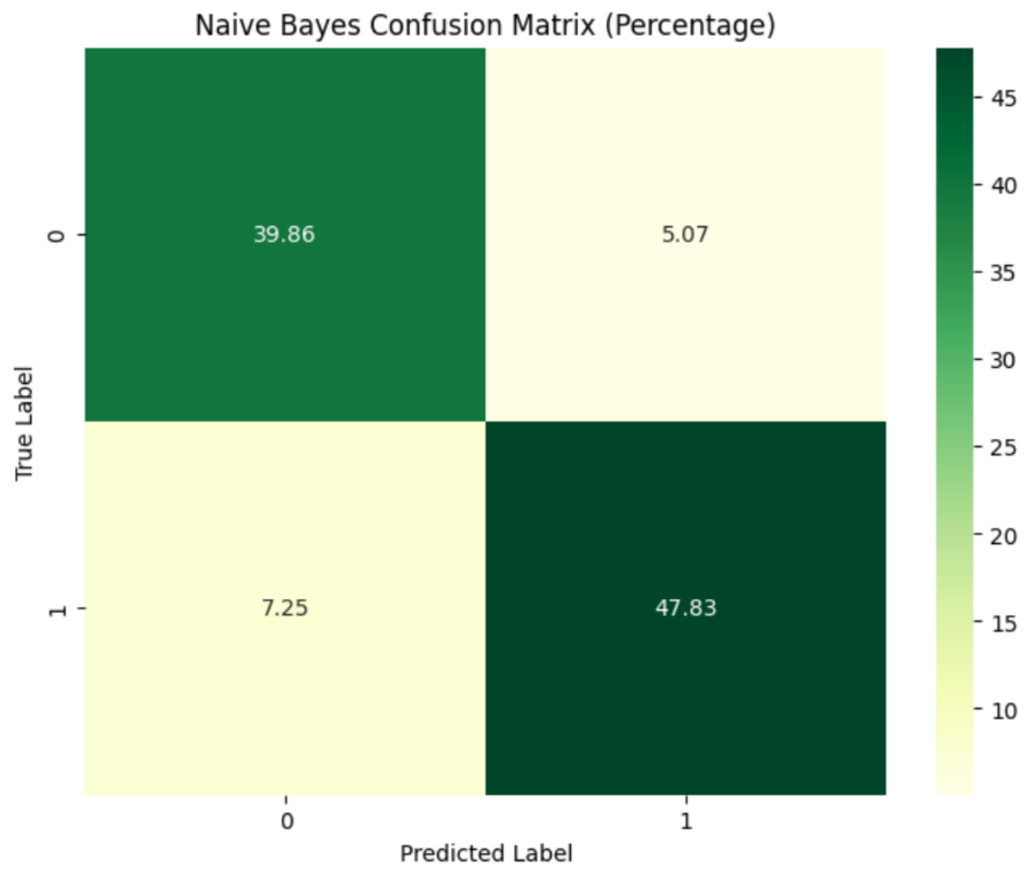
3. **Validation & Testing Performance of SVM:**

```
Validation Accuracy (SVM): 0.91
Test Accuracy: 0.88
Precision: 0.89
Recall: 0.88
F1-Score: 0.88
```

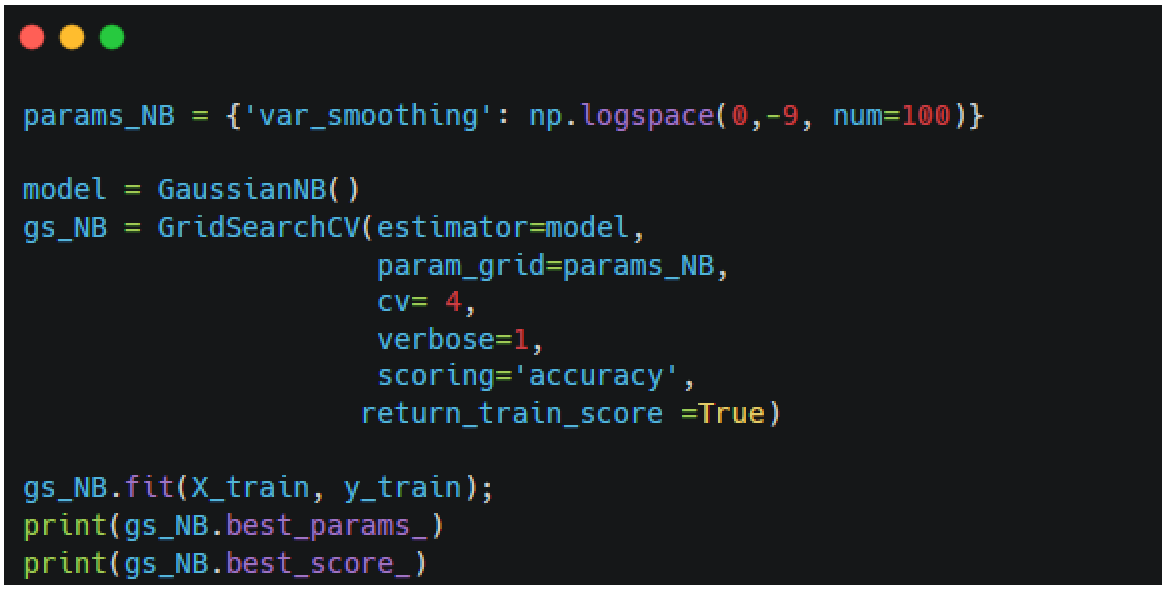
4. **Classification report of SVM:**

```
    precision    recall  f1-score   support

           0       0.86      0.87      0.86        62
           1       0.89      0.88      0.89        76

    accuracy                           0.88       138
   macro avg       0.88      0.88      0.88       138
weighted avg       0.88      0.88      0.88       138
```

5. **Confusion matrix of SVM:**



2nd model: GridSearchCV with Gaussian Naive Bayes (NB)



```
params_NB = {'var_smoothing': np.logspace(0,-9, num=100)}

model = GaussianNB()
gs_NB = GridSearchCV(estimator=model,
                     param_grid=params_NB,
                     cv= 4,
                     verbose=1,
                     scoring='accuracy',
                    return_train_score =True)

gs_NB.fit(X_train, y_train);
print(gs_NB.best_params_)
print(gs_NB.best_score_)
```

Define the Hyperparameter Grid

`var_smoothing`:

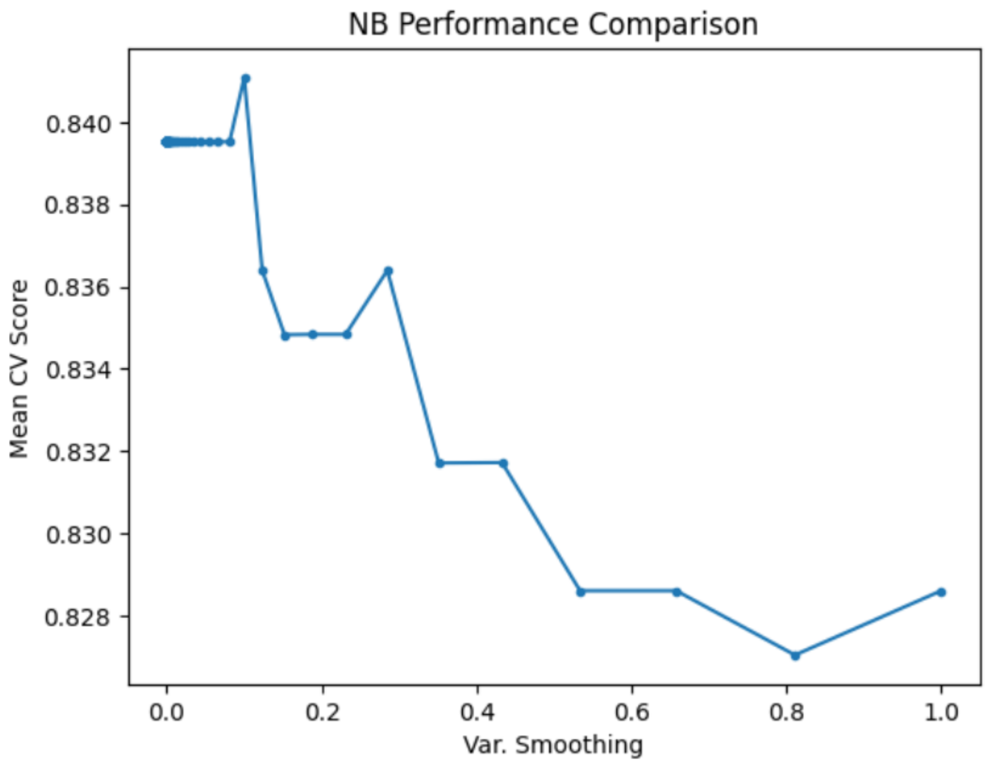
- Adds a small constant to the variance of each feature to avoid numerical instabilities during computation.
- Affects how sensitive the model is to small fluctuations in the data.
- The range is specified as logarithmic, from 10^0 to 10^{-9} ,split into 100 values using `np.logspace`

Observations for Naive Bayes(NB):

6. Best Hyperpararmeters:

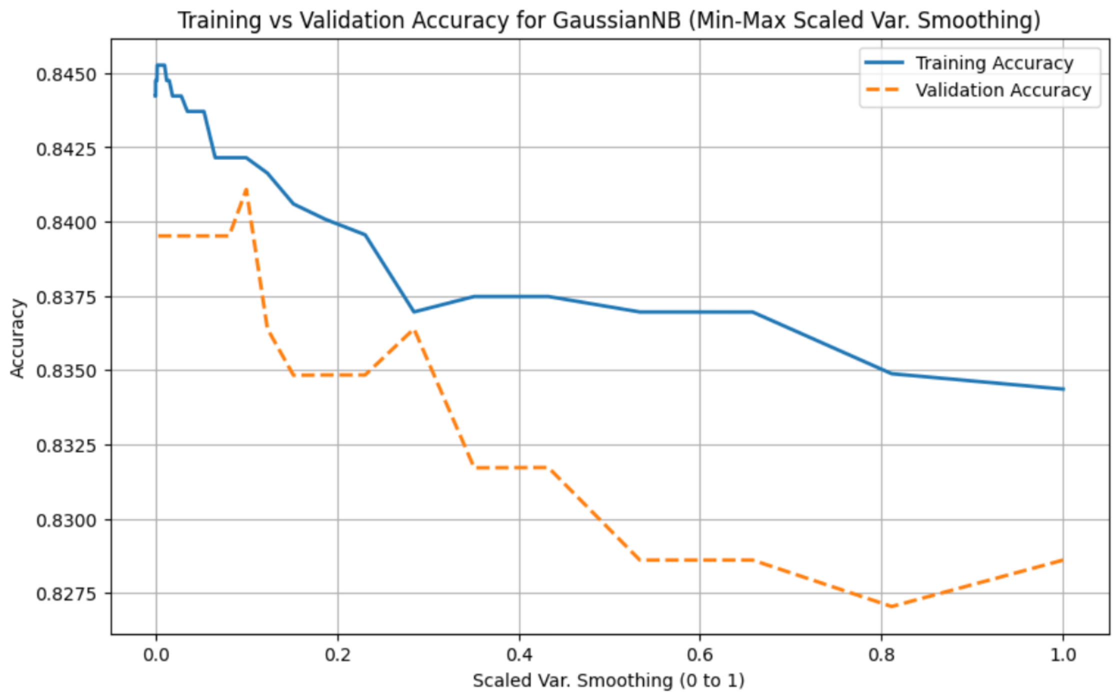
```
Fitting 4 folds for each of 100 candidates, totalling 400 fits
{'var_smoothing': 0.1}
best score: 0.84
```

7. **Naive Bayes CV performance comparison with var_smoothing:**
   - Plot visualizes the relationship between the `var_smoothing` parameter and the **mean cross-validation (CV) score**. It helps identify the value of `var_smoothing` that results in the highest accuracy during cross-validation.



8. **Grid search Training and Validation Accuracy of NB:**

   - compares the **training accuracy** and **validation accuracy** of the Gaussian Naive Bayes (GaussianNB) model as a function of the **min-max scaled `var_smoothing` parameter**.



- Training Accuracy :
- The training accuracy starts high (around 85%) even at small scaled `var_smoothing` values.
- Validation Accuracy :
- The validation accuracy starts lower than the training accuracy (around 84%) for small `var_smoothing` values.

- **Optimal Region**:
- Both training and validation accuracy reach their peaks in the higher range of scaled `var_smoothing` (around 0.0 to 2.0).
- This indicates that a larger `var_smoothing` helps stabilize the model's variance and improves performance across training and validation sets.

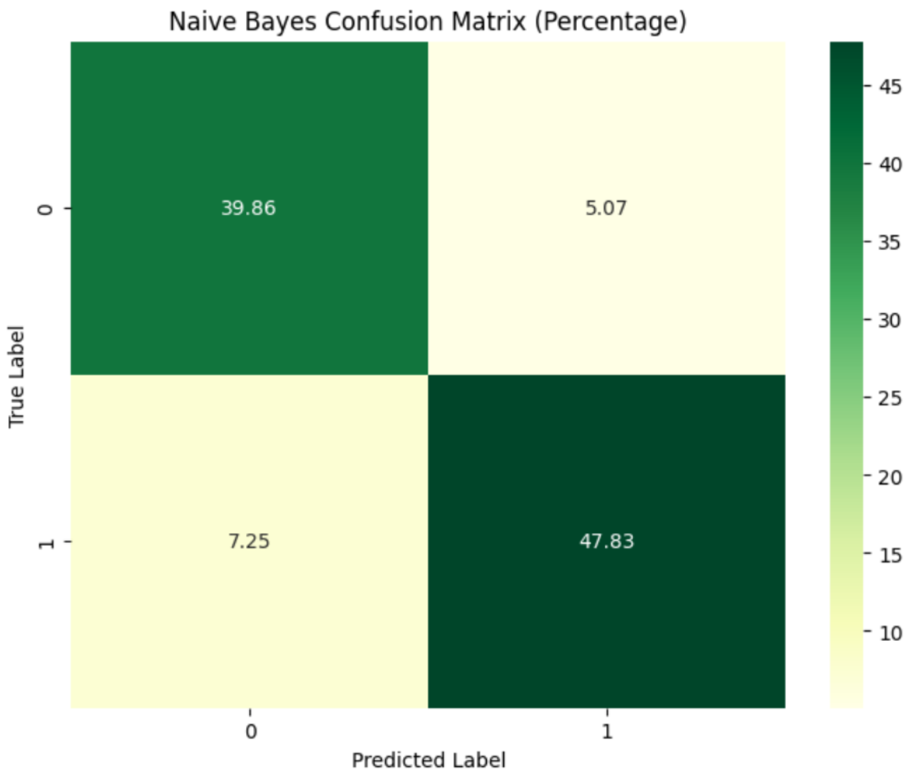
9. **Validation & Testing Performance of NB**:

```
Validation Accuracy (Naive Bayes): 0.8696
Naive Bayes Recall: 0.8684
Naive Bayes Precision: 0.9041
Naive Bayes F1 Score: 0.8770
Naive Bayes Accuracy: 0.8768
```

10. **Classification Report of NB:**

```
              precision    recall  f1-score   support

           0       0.85      0.89      0.87        62
           1       0.90      0.87      0.89        76

    accuracy                           0.88       138
   macro avg       0.88      0.88      0.88       138
weighted avg       0.88      0.88      0.88       138
```

11. Confusion Matrix of NB:



- In-detail Comparison & Results Explanation:

Points Of Comparison	**Naïve Bayes Classifier**	**SVM classifier**
1.Assumptions:	**1.Feature independence:** The features of the data are conditionally independent of each other, given the class label. **2.Continuous features are normally distributed:** If a feature is continuous, then it is assumed to be normally distributed within each class.	**1. Linear Separability:** There exists a hyperplane that can distinctly separate the classes. **2. Margin Maximization**: SVM assumes that a larger margin contributes to better generalization and improved performance. **3.Noisy Data Handling:** SVM are sensitive to noisy data and outliers,
2.Strengths:	**1. Handles continuous and discrete data,** and it is not sensitive to irrelevant features. **2. Very simple**, fast, and easy to implement. **3. Highly scalable** as it scales linearly with the number of predictor features and data points.	**1.Powerful in High-Dimensional Spaces**: Effective when the number of features is much larger than the number of data points. **2.Flexible**: Can handle both linear and non-linear decision boundaries (through kernel functions).

	4. When the Naive Bayes conditional independence assumption holds true, it will converge quicker than discriminative models.	**3.Robust to Overfitting:** SVMs are less prone to overfitting, due to the use of a margin that penalizes data points inside the margin. **4.Kernel Trick for Non-Linear Data:** The kernel trick allows SVMs to handle non-linear decision boundaries by implicitly mapping data into higher-dimensional spaces.
3.Weaknesses:	1. **The assumption of independent predictors/features:** which is almost impossible to find in real-world data. **2.Feature Equality:** it Simplifies calculations, Ignores potential feature importance differences.	1. **Sensitivity to Noise and Outliers:** SVMs can be sensitive to noise and outliers, as they may influence the position and orientation of the decision boundary. **2. Kernel and Parameters:** tuning of hyperparameters can be challenging, and the performance may be sensitive to these choices. **3.Not Suitable for Imbalanced Datasets:** SVMs may not perform

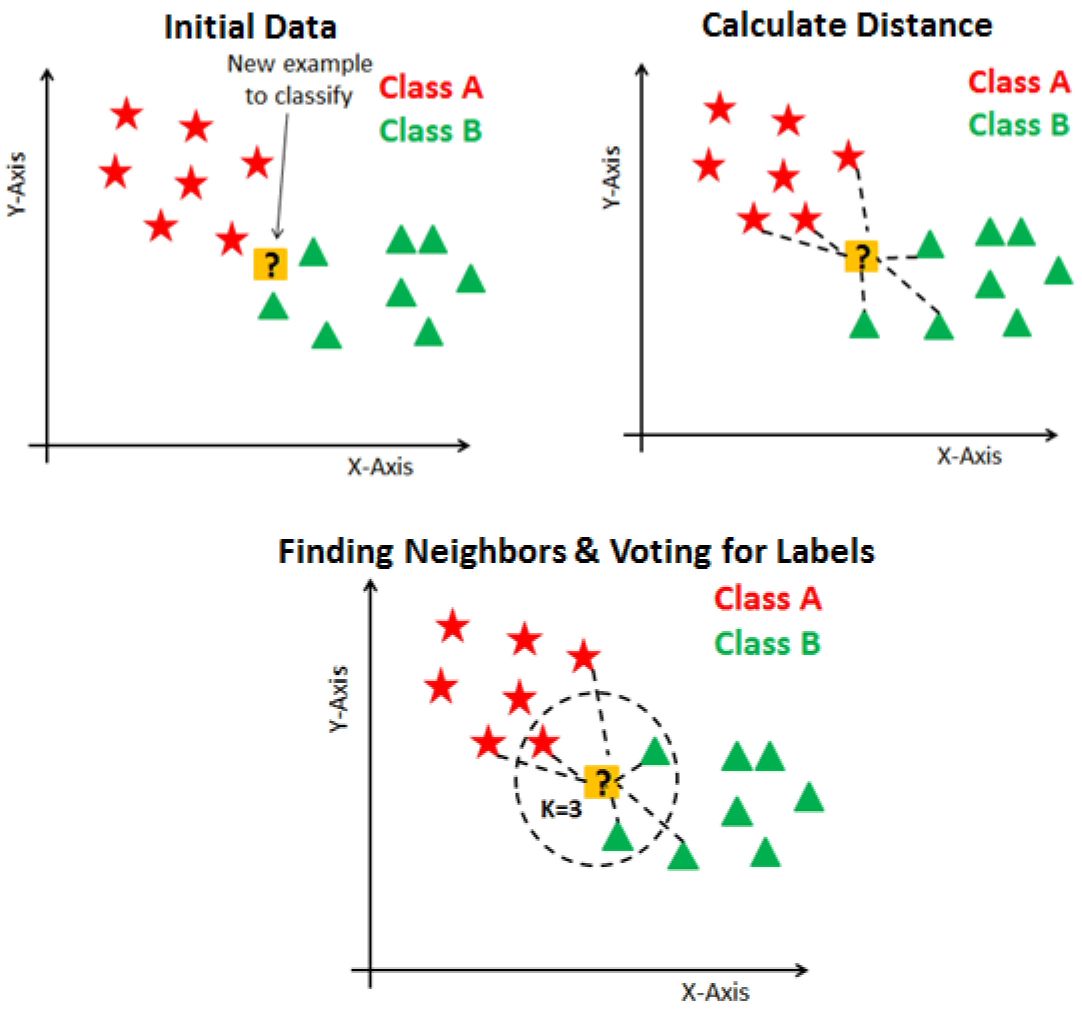
		well on highly imbalanced datasets where one class significantly outnumbers the other.
4.Recall_Precision:	**Area Under the Curve(AUC):** Naive Bayes can sometimes exhibit a larger area under the curve (AUC) because of its probabilistic nature, which allows it to assign probabilities to each class and thus handle imbalanced datasets relatively well. **Decay: However, Naive Bayes tends to "decay faster" in its performance as the decision boundaries can be overly simplistic.** **The decay refers to a faster decrease in precision and recall as the data becomes more challenging (i.e., features are correlated, the dataset becomes noisy, or the classes are not as well separated).**	**Area Under the Curve(AUC):** SVM, particularly with a non-linear kernel, tends to perform better on complex datasets. **Decay**: **Slower Decay than Naïve Bayes** as It handles complex feature interactions better due to the kernel trick, and its ability to create a non-linear decision boundary often allows it to maintain higher precision and recall even as the dataset becomes more difficult.

Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

3rd model: GridSearchCV with KNN

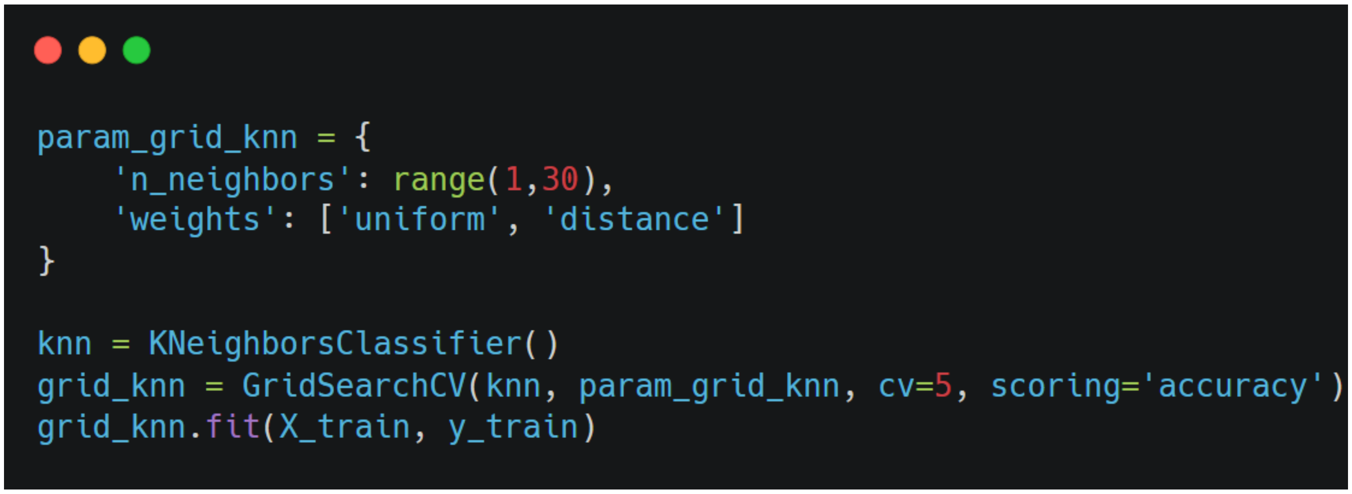
The optimal K value usually found is the square root of N, where N is the total number of samples

K-nearest neighbors (k-NN) is a pattern recognition algorithm that uses training datasets to find the k closest relatives in future examples.



Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

Code:



```python
param_grid_knn = {
    'n_neighbors': range(1,30),
    'weights': ['uniform', 'distance']
}

knn = KNeighborsClassifier()
grid_knn = GridSearchCV(knn, param_grid_knn, cv=5, scoring='accuracy')
grid_knn.fit(X_train, y_train)
```

- n_neighbors: Number of neighbors to use ranging from 1 to 29.
- weights: Two options for assigning weights to neighbors:
  - 'uniform':All points in each neighborhood are weighted equally
  - 'distance': weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- `KNeighborsClassifier():` creates an instance of the KNN model without any predefined parameters.

Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

Observations for K-Nearest Neighbors (KNN):

12. Best Hyperparameters

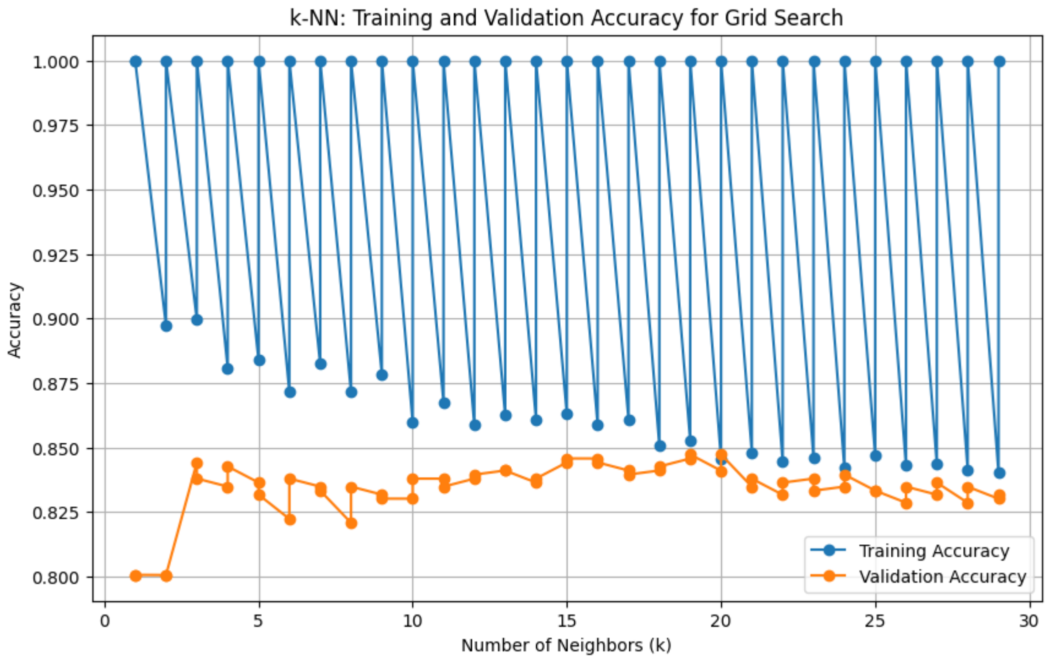
- n_neighbors: 19
- Weights: distance

13. Grid search Training and Validation Accuracy:

Training Accuracy: As k increases, training accuracy decreases.

Validation Accuracy: stable range around k=10 to 20, k>20 shows no improvement.

Optimal Range for k : Range from k = 10 to 20 where validation accuracy stabilizes.



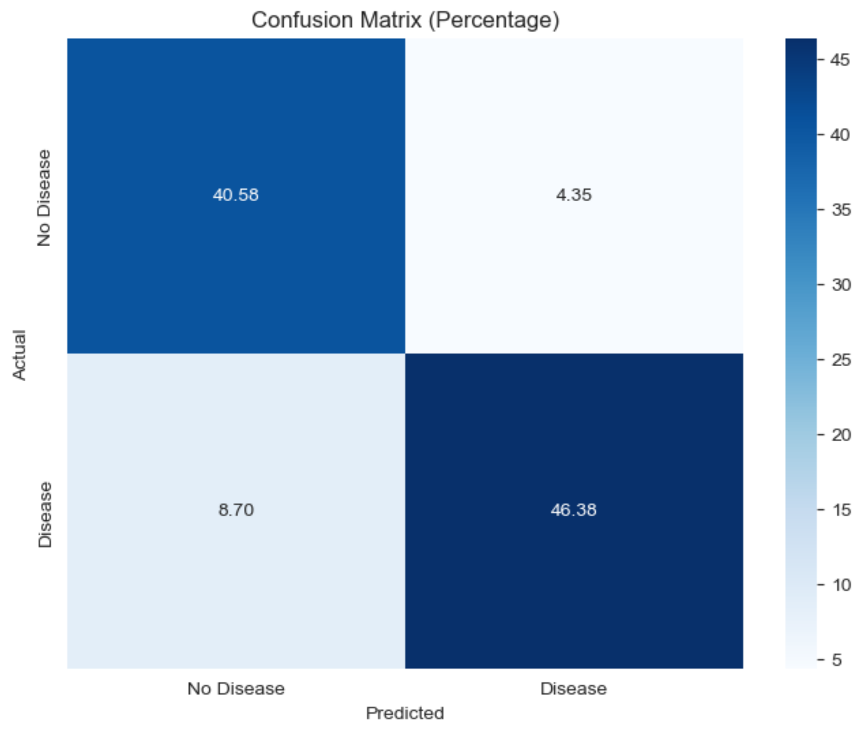
14. **Validation & Testing Performance of KNN**:

- **Test Accuracy:** 0.87
- **Precision:** 0.87
- **Recall:** 0.87
- **F1-Score:** 0.87

15. **Classification Report of KNN:**

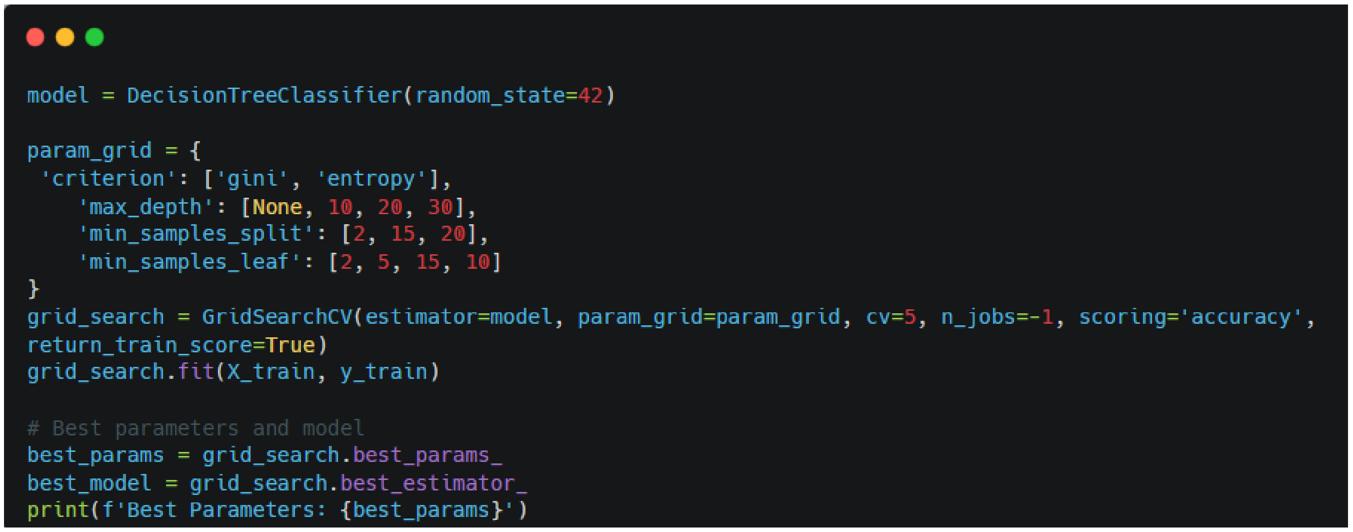
```
              precision    recall  f1-score   support

           0       0.82      0.90      0.86        62
           1       0.91      0.84      0.88        76

    accuracy                           0.87       138
   macro avg       0.87      0.87      0.87       138
weighted avg       0.87      0.87      0.87       138
```

16. Confusion matrix of KNN:



4th model: GridSearchCV with Decision Trees:

code



```
model = DecisionTreeClassifier(random_state=42)

param_grid = {
 'criterion': ['gini', 'entropy'],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 15, 20],
    'min_samples_leaf': [2, 5, 15, 10]
}
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5, n_jobs=-1, scoring='accuracy',
return_train_score=True)
grid_search.fit(X_train, y_train)

# Best parameters and model
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_
print(f'Best Parameters: {best_params}')
```

The Hyperparameter grid:

- **Criteria:** The quality of the split in the decision tree is measured by the function called criteria.
    - The **Gini index** measures the impurity of a dataset, with lower values indicating a purer (more homogeneous) node.
    - **Information gain**: It is an impurity measure that uses the entropy measure to split a node in a way that it yields the most amount of information gain.
- **max_depth**: max_depth hyperparameter controls the maximum depth to which the decision tree is allowed to grow.

  - The grid search is allowed to use None (No limit on the depth), which means the tree will grow until all leaves are pure or until they contain fewer than `min_samples_split` samples.
  - Or [10, 20, 30] specific limits on the depth of the tree as shallow trees (smaller values) prevent overfitting.

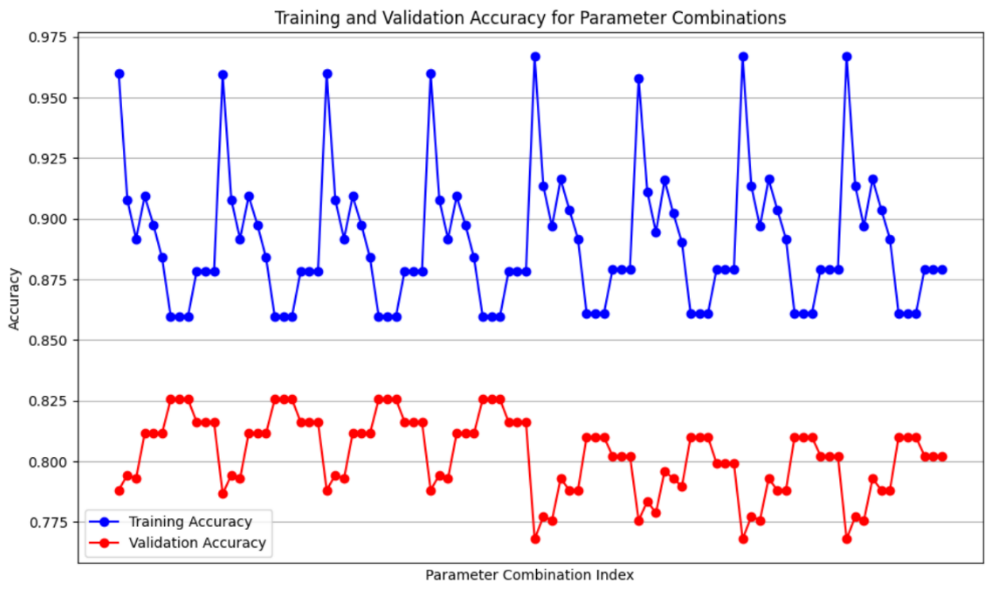
- **min_samples_split:** controls the minimum number of samples required to split an internal node.
- **min_samples_leaf**: controls the minimum number of samples required to be at a leaf node.
  - **`[5, 10, 15]`**: Increasing this number forces each leaf to have more samples, which can help the model generalize better but may reduce its ability to fit the training data closely.

Observations for Decision Tree (DT):

1. **The Best Hyperparameters of DT:**

{'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 15, 'min_samples_split': 2}

2. Grid Search Training and validation Accuracy of DT:



- The fluctuations in the accuracy graph you're seeing are a direct result of how the hyperparameters interact with the data and model complexity.

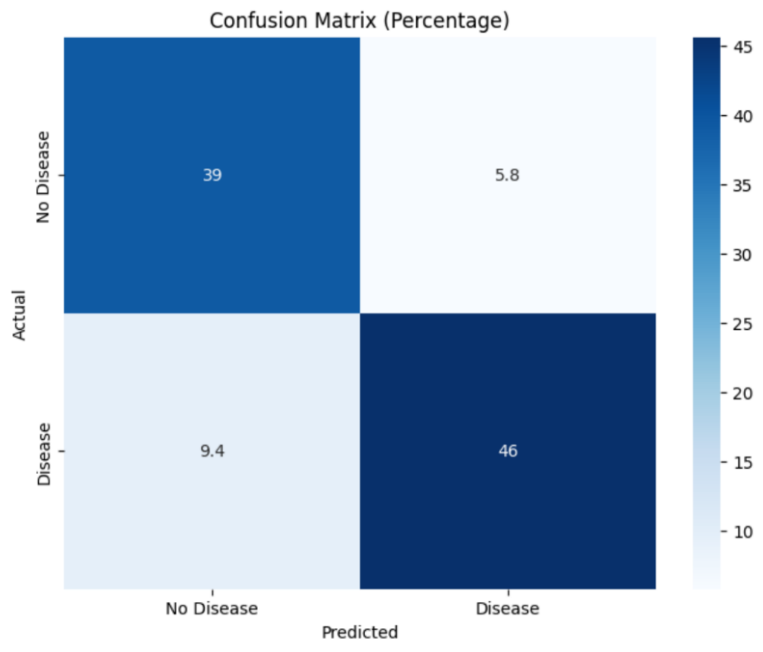
3. Validation & Testing Performance of DT:

```
Validation Accuracy: 0.87
Test Accuracy: 0.85
Precision: 0.85
Recall: 0.85
F1-Score: 0.85
```

4. **Classification Report of DT:**

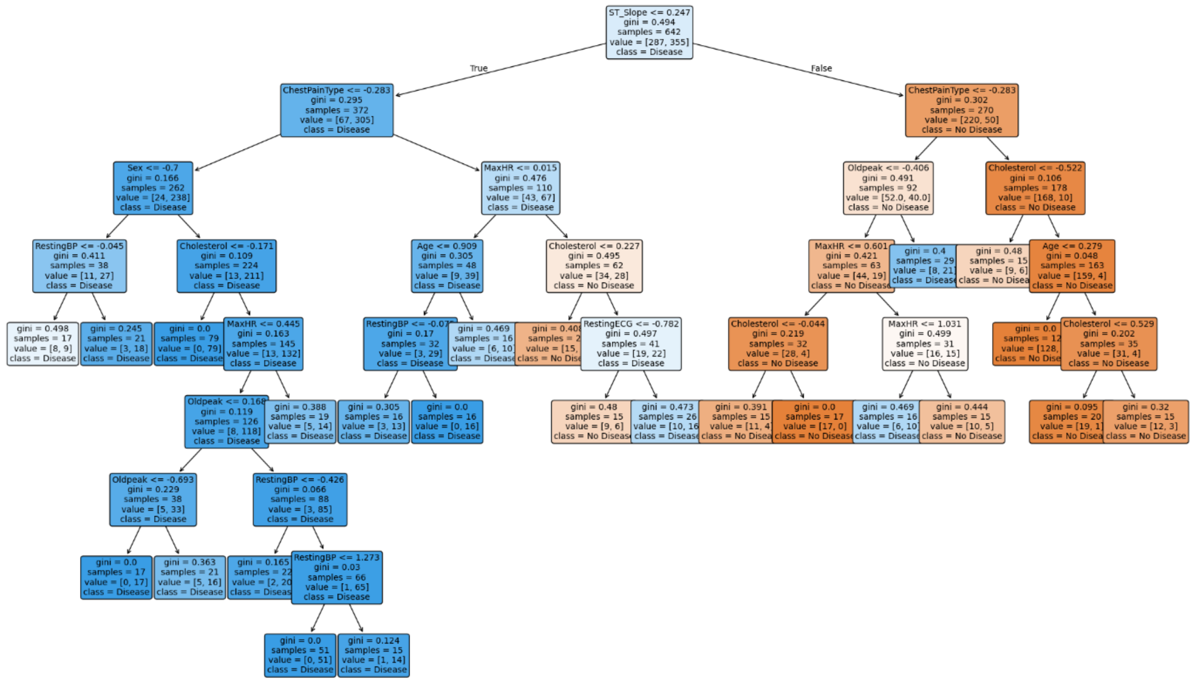
```
              precision    recall  f1-score   support

           0       0.81      0.87      0.84        62
           1       0.89      0.83      0.86        76

    accuracy                           0.85       138
   macro avg       0.85      0.85      0.85       138
weighted avg       0.85      0.85      0.85       138
```

5. **Confusion matrix of DT:**



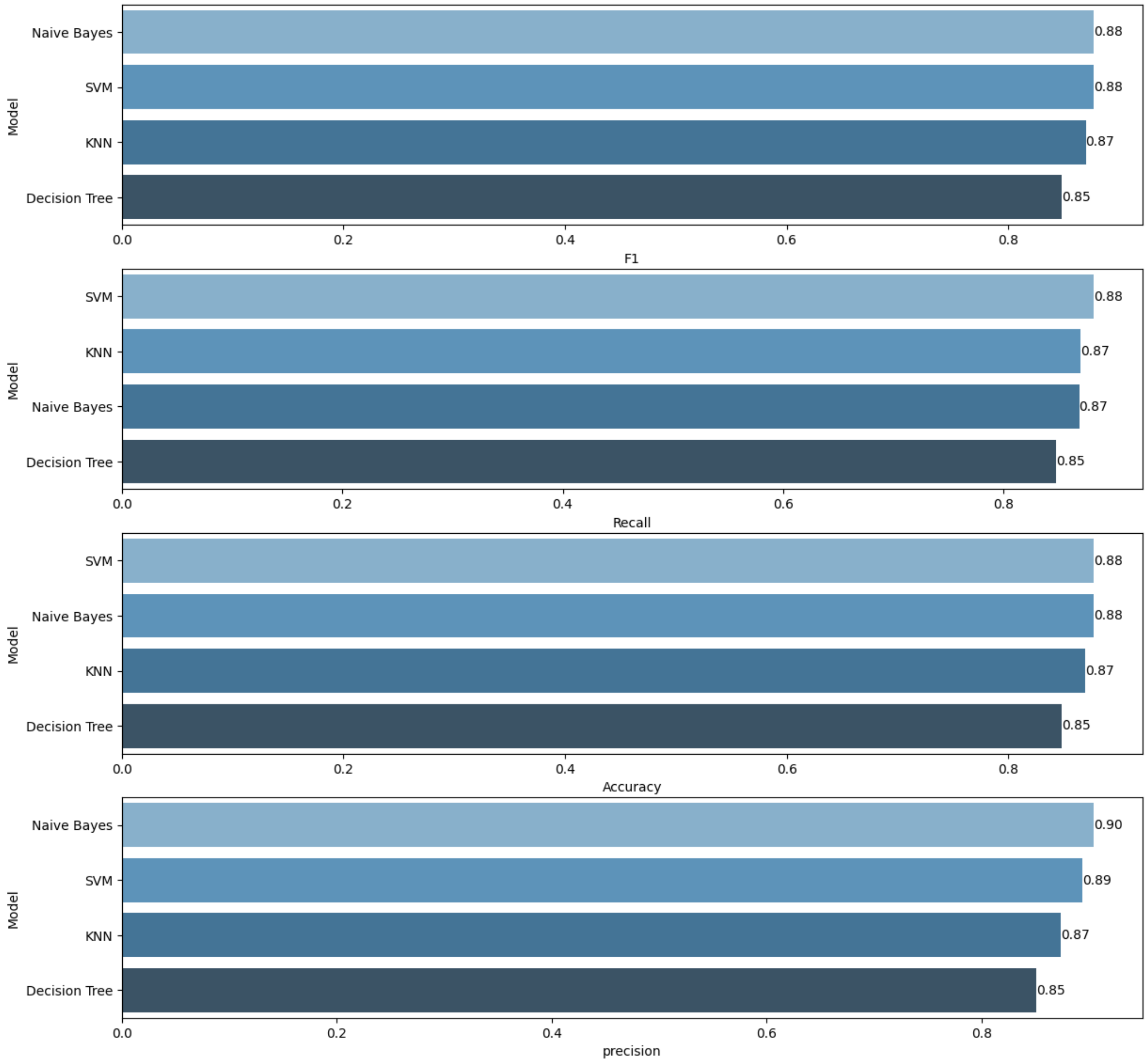
Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

6. Decision Tree of best hyperparameters:


ST_Slope <= 0.247 gini = 0.494 samples = 642 value = [287, 355] class = Disease
True
False
ChestPainType <= -0.283 gini = 0.295 samples = 372 value = [67, 305] class = Disease
ChestPainType <= -0.283 gini = 0.302 samples = 270 value = [220, 50] class = No Disease
Sex <= -0.7 gini = 0.166 samples = 262 value = [24, 238] class = Disease
MaxHR <= 0.015 gini = 0.476 samples = 110 value = [43, 67] class = Disease
Oldpeak <= -0.406 gini = 0.491 samples = 92 value = [52.0, 40.0] class = No Disease
Cholesterol <= -0.522 gini = 0.106 samples = 178 value = [168, 10] class = No Disease
RestingBP <= -0.045 gini = 0.411 samples = 38 value = [11, 27] class = Disease
Cholesterol <= -0.171 gini = 0.109 samples = 224 value = [13, 211] class = Disease
Age <= 0.909 gini = 0.305 samples = 48 value = [9, 39] class = Disease
Cholesterol <= 0.227 gini = 0.495 samples = 62 value = [34, 28] class = No Disease
MaxHR <= 0.601 gini = 0.421 samples = 63 value = [44, 19] class = No Disease
gini = 0.4 samples = 29 value = [8, 21] class = Disease
gini = 0.48 samples = 15 value = [9, 6] class = No Disea
Age <= 0.279 gini = 0.048 samples = 163 value = [159, 4] class = No Disease
gini = 0.498 samples = 17 value = [8, 9] class = Disease
gini = 0.245 samples = 21 value = [3, 18] class = Disease
gini = 0.0 samples = 79 value = [0, 79] class = Disease
MaxHR <= 0.445 gini = 0.163 samples = 145 value = [13, 132] class = Disease
RestingBP <= -0.07 gini = 0.17 samples = 32 value = [3, 29] class = Disease
gini = 0.469 samples = 16 value = [6, 10] class = Disease
gini = 0.408 samples = 2 value = [15, class = No Dis
RestingECG <= -0.782 gini = 0.497 samples = 41 value = [19, 22] class = Disease
Cholesterol <= -0.044 gini = 0.219 samples = 32 value = [28, 4] class = No Disease
MaxHR <= 1.031 gini = 0.499 samples = 31 value = [16, 15] class = No Disease
gini = 0.0 samples = 12 value = [128, class = No Dise
Cholesterol <= 0.529 gini = 0.202 samples = 35 value = [31, 4] class = No Disease
Oldpeak <= 0.168 gini = 0.119 samples = 126 value = [8, 118] class = Disease
gini = 0.388 samples = 19 value = [5, 14] class = Disease
gini = 0.305 samples = 16 value = [3, 13] class = Disease
gini = 0.0 samples = 16 value = [0, 16] class = Disease
gini = 0.48 samples = 15 value = [9, 6] class = No Disease
gini = 0.473 samples = 26 value = [10, 16 class = Disease
gini = 0.391 samples = 15 value = [11, 4 class = No Disea
gini = 0.0 samples = 17 value = [17, 0] class = No Disease
gini = 0.469 samples = 16 value = [6, 10 class = Disease
gini = 0.444 samples = 15 value = [10, 5] class = No Disease
gini = 0.095 samples = 20 value = [19, 1 class = No Disea
gini = 0.32 samples = 15 value = [12, 3] class = No Disease
Oldpeak <= -0.693 gini = 0.229 samples = 38 value = [5, 33] class = Disease
RestingBP <= -0.426 gini = 0.066 samples = 88 value = [3, 85] class = Disease
gini = 0.0 samples = 17 value = [0, 17] class = Disease
gini = 0.363 samples = 21 value = [5, 16] class = Disease
gini = 0.165 samples = 22 value = [2, 20 class = Diseas
RestingBP <= 1.273 gini = 0.03 samples = 66 value = [1, 65] class = Disease
gini = 0.0 samples = 51 value = [0, 51] class = Disease
gini = 0.124 samples = 15 value = [1, 14] class = Disease

Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

Comparison of the performance of different classifiers:



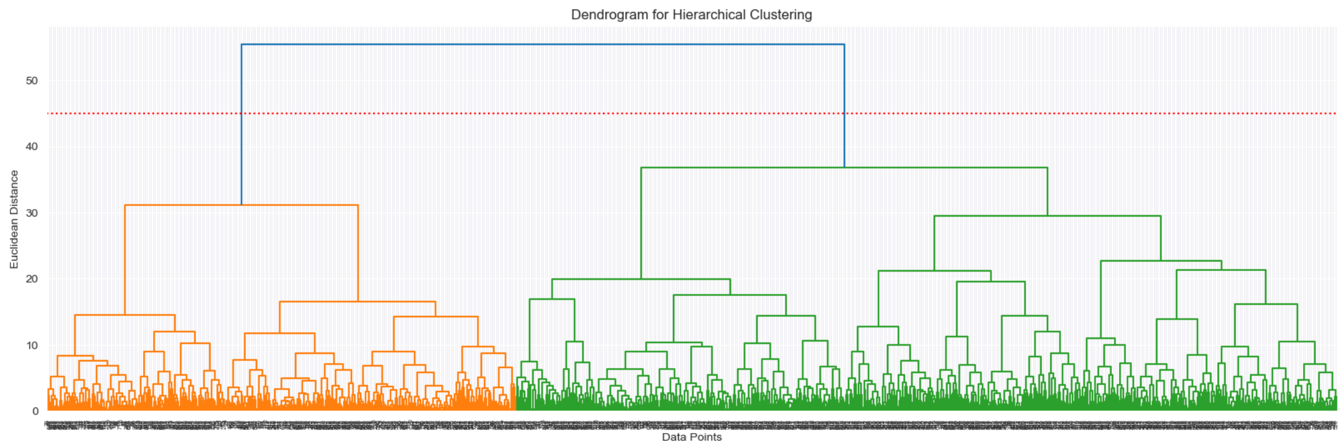
1. **F1-Score**:
    a. **Naive Bayes** and **SVM** have the highest F1-scores 0.88
    b. **KNN** is slightly lower at 0.87
    c. **Decision Tree** performs the lowest with 0.85
2. **Recall**:
    a. **SVM** have the highest recall 0.88
    b. **KNN** and **Naive Bayes** close with 0.87
    c. **Decision Tree** scores the lowest again at 0.85
3. **Accuracy**:
    a. **Naive Bayes** leads in accuracy with 0.90
    b. closely followed by **SVM** with 0.89
    c. **KNN** has an accuracy of 0.87
    d. **Decision Tree** lags at 0.85
4. **Precision**:
    a. **Naive Bayes** achieves the highest precision 0.90
    b. **SVM** follows with 0.89
    c. while **KNN** has a precision of 0.87
    d. **Decision Tree** continues to underperform with 0.85

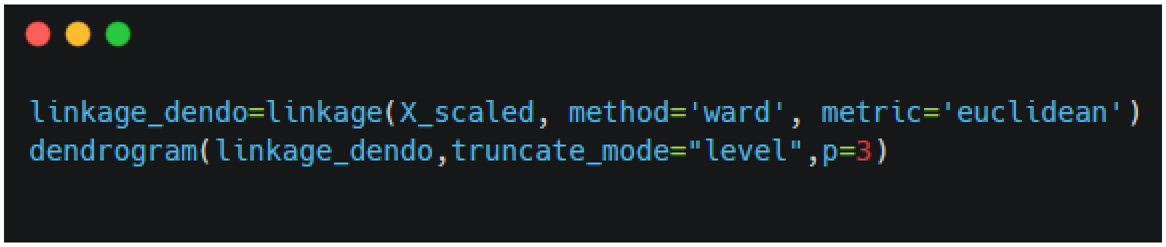
Conclusion

Naive Bayes is the best-performing model overall, SVM & KNN competitive models, **Decision Tree** is Poor Performance.

Course code: CSE381
Course Name: Introduction to Machine Learning
Instructor: Dr. Mahmoud Khalil
TA: Eng. Engy Ahmed Hassan

Dendrogram





```
linkage_dendo=linkage(X_scaled, method='ward', metric='euclidean')
dendrogram(linkage_dendo,truncate_mode="level",p=3)
```

method='ward': Uses Ward's minimum variance criterion to minimize the variance within each cluster.

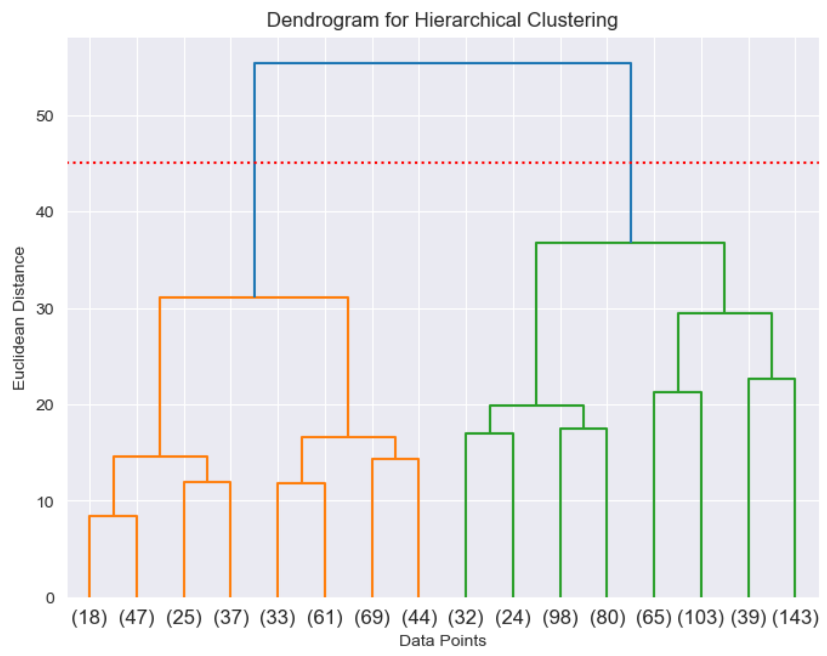
metric='euclidean': Uses Euclidean distance to measure the dissimilarity between data points.

P : set number of levels show in the dendrogram, Removing P causes the dendrogram to default, his results in a more detailed.

Dotted Line represents a threshold to determine the number of clusters.

The y-axis represents the distance (dissimilarity) between clusters when they are merged.

the graph contain all 918 data points

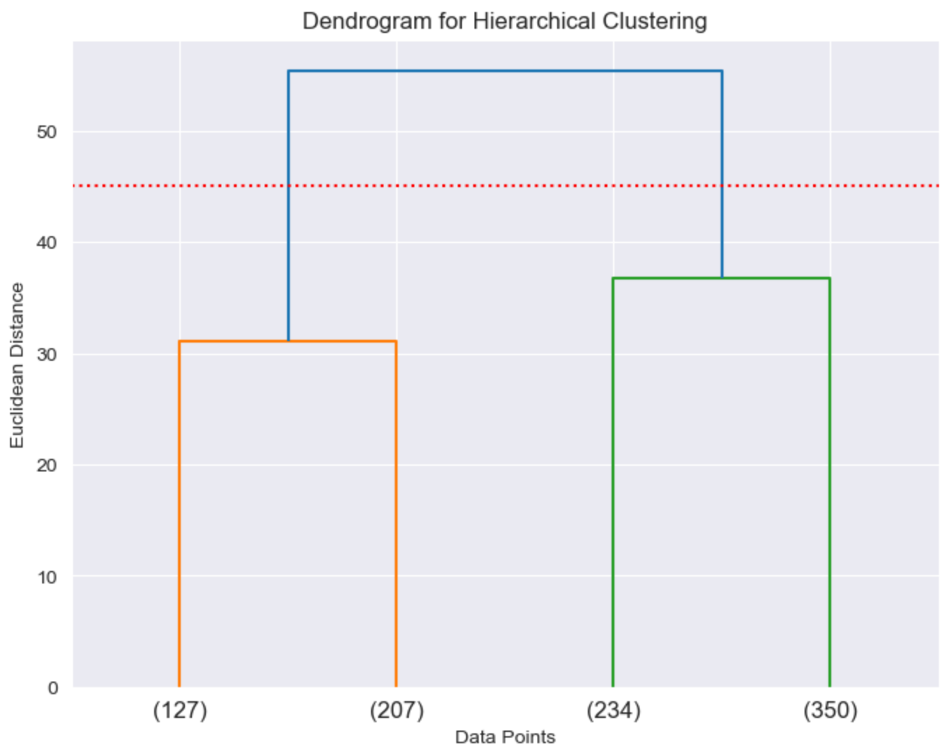


In level 3 clustering

The dendrogram shows the merging of data points (indicated by their indices) and clusters at various levels based on Euclidean distance.

Smaller merges at the bottom indicate highly similar data points, while larger merges near the top indicate less similarity.

- The **orange cluster** on the left.
- The **green cluster** on the right.
- The blue line at the top represents the largest distance between the two main clusters



2 clusters :

- **Orange cluster**: Contains data points 127 and 207. Refer to no Disease
- **Green cluster**: Contains data points 234 and 350. Refer to Disease

Distance Between Clusters:

- The **orange** and **green** clusters are merged at a **high Euclidean distance (~50)**, indicating that the two clusters are relatively far apart in the feature space.
- **Intra-Cluster Proximity**:
- Data points **127 and 207** in the orange cluster are merged at a smaller Euclidean distance (~30)
- Similarly, data points **234 and 350** in the green cluster are also merged at a small distance (~30)

how the hierarchical clustering insights align with the PCA and classifier results.

Dendrogram: Data points are grouped into two main clusters with significant separation, as indicated by the large Euclidean distance at the final merge.

Principal Component Analysis Scatter Plot: The PCA plot shows data projected into the first two principal components, There is some overlap between the two classes (Heart Disease and No Heart Disease)

Alignment:

- Hierarchical clustering consistently identifies two main clusters, which aligns with the binary target variable (Heart Disease: Yes/No).
- PCA shows partial separation in the first two components, which corresponds to the structure observed in the dendrograms. This suggests that the clusters identified in hierarchical clustering are meaningful and reflect real patterns in the data.

Differences:

- PCA does not perfectly separate the two classes, indicating that more principal components or non-linear relationships may contribute to the clustering.
- Classifier performance can provide additional insight into how well the clusters align with the target variable.