In this project, we delve into a dataset encapsulating various health metrics from heart patients, including age, blood pressure, heart rate, and more. Our goal is to develop a predictive model capable of accurately identifying individuals with heart disease. Given the grave implications of missing a positive diagnosis, our primary emphasis is on ensuring that the model identifies all potential patients, making recall for the positive class a crucial metric.

Objectives:

- **Explore the Dataset**: Uncover patterns, distributions, and relationships within the data.
- **Conduct Extensive Exploratory Data Analysis (EDA)**: Dive deep into bivariate relationships against the target.
- **Preprocessing Steps**:
  - Remove irrelevant features.
  - Address missing values.
  - Treat outliers
  - Encode categorical variables.
  - Transform skewed features to achieve normal-like distributions.
- **Model Building**:
  - Establish pipelines for models that require scaling.
  - Implement and tune classification models including KNN, SVM, Decision Trees, and Random Forest
  - Emphasize achieving high recall for class 1, ensuring comprehensive identification of heart patients.
- **Evaluate and Compare Model Performance**: Utilize precision, recall, and F1-score to gauge models' effectiveness.

Dataset Description:

Variable	**Description**
age	Age of the patient in years
sex	Gender of the patient (0 = male, 1 = female)
cp	Chest pain type:

Variable	Description
	0: Typical angina 1: Atypical angina 2: Non-anginal pain 3: Asymptomatic
trestbps	Resting blood pressure in mm Hg
chol	Serum cholesterol in mg/dl
fbs	Fasting blood sugar level, categorized as above 120 mg/dl (1 = true, 0 = false)
restecg	Resting electrocardiographic results: 0: Normal 1: Having ST-T wave abnormality 2: Showing probable or definite left ventricular hypertrophy
thalach	Maximum heart rate achieved during a stress test
exang	Exercise-induced angina (1 = yes, 0 = no)
oldpeak	ST depression induced by exercise relative to rest
slope	Slope of the peak exercise ST segment: 0: Upsloping 1: Flat 2: Downsloping
ca	Number of major vessels (0-4) colored by fluoroscopy
thal	Thalium stress test result: 0: Normal 1: Fixed defect 2: Reversible defect 3: Not described
target	Heart disease status (0 = no disease, 1 = presence of disease)

[ln 3]

Inferences:

- **Number of Entries**: The dataset consists of **303 entries**, ranging from index 0 to 302.
- **Columns**: There are **14 columns** in the dataset corresponding to various attributes of the patients and results of tests.

- **Data Types**:
    - Most of the columns (13 out of 14) are of the **int64** data type.
    - Only the oldpeak column is of the float64 data type.
- **Missing Values**: There don't appear to be any missing values in the dataset as each column has 303 non-null entries.

Note: Based on the data types and the feature explanations we had earlier, we can see those 9 columns (**sex, cp, fbs, restecg, exang, slope, ca, thal, and target**) are indeed **numerical** in terms of data type, but **categorical** in terms of their semantics. These features should be converted to string (**object**) data type for proper analysis and interpretation:

[ln 4 to ln 5]

Numerical Features:

- **age**: The average age of the patients is approximately 54.4 years, with the youngest being 29 and the oldest 77 years.
- **trestbps**: The average resting blood pressure is about 131.62 mm Hg, ranging from 94 to 200 mm Hg.
- **chol**: The average cholesterol level is approximately 246.26 mg/dl, with a minimum of 126 and a maximum of 564 mg/dl.
- **thalach**: The average maximum heart rate achieved is around 149.65, with a range from 71 to 202.
- **oldpeak**: The average ST depression induced by exercise relative to rest is about 1.04, with values ranging from 0 to 6.2.

[ln 6]

Categorical Features (object data type):

- **sex**: There are two unique values, with males (denoted as 0) being the most frequent category, occurring 207 times out of 303 entries.
- **cp**: Four unique types of chest pain are present. The most common type is "**0**", occurring 143 times.
- **fbs**: There are two categories, and the most frequent one is "**0**" (indicating fasting blood sugar less than 120 mg/dl), which appears 258 times.
- **restecg**: Three unique results are present. The most common result is "**1**", appearing 152 times.

- **exang**: There are two unique values. The most frequent value is "**0**" (indicating no exercise-induced angina), which is observed 204 times.
- **slope**: Three unique slopes are present. The most frequent slope type is "**2**", which occurs 142 times.
- **ca**: There are five unique values for the number of major vessels colored by fluoroscopy, with "**0**" being the most frequent, occurring 175 times.
- **thal**: Four unique results are available. The most common type is "**2**" (indicating a reversible defect), observed 166 times.
- **target**: Two unique values indicate the presence or absence of heart disease. The value "**1**" (indicating the presence of heart disease) is the most frequent, observed in 165 entries.

[ln 7]

For our **Exploratory Data Analysis (EDA)**, we'll take it in two main steps:

1. Univariate Analysis: Here, we'll focus on one feature at a time to understand its distribution and range.

2. Bivariate Analysis: In this step, we'll explore the relationship between each feature and the target variable. This helps us figure out the importance and influence of each feature on the target outcome.

With these two steps, we aim to gain insights into the individual characteristics of the data and how each feature relates to our main goal: **predicting the target variable**.

We undertake univariate analysis on the dataset's features, based on their datatype:

- For **continuous data**: We employ histograms to gain insight into the distribution of each feature. This allows us to understand the central tendency, spread, and shape of the dataset's distribution.
- For **categorical data**: Bar plots are utilized to visualize the frequency of each category. This provides a clear representation of the prominence of each category within the respective feature.

By employing these visualization techniques, we're better positioned to understand the individual characteristics of each feature in the dataset.

Inferences:

- **Age (age)**: The distribution is somewhat uniform, but there's a peak around the late 50s. The mean age is approximately 54.37 years with a standard deviation of 9.08 years.

- **Resting Blood Pressure (trestbps)**: The resting blood pressure for most individuals is concentrated around 120-140 mm Hg, with a mean of approximately 131.62 mm Hg and a standard deviation of 17.54 mm Hg.
- **Serum Cholesterol (chol)**: Most individuals have cholesterol levels between 200 and 300 mg/dl. The mean cholesterol level is around 246.26 mg/dl with a standard deviation of 51.83 mg/dl.
- **Maximum Heart Rate Achieved (thalach)**: The majority of the individuals achieve a heart rate between 140 and 170 bpm during a stress test. The mean heart rate achieved is approximately 149.65 bpm with a standard deviation of 22.91 bpm.
- **ST Depression Induced by Exercise (oldpeak)**: Most of the values are concentrated towards 0, indicating that many individuals did not experience significant ST depression during exercise. The mean ST depression value is 1.04 with a standard deviation of 1.16.

Upon reviewing the histograms of the continuous features and cross-referencing them with the provided feature descriptions, everything appears consistent and within expected ranges. **There doesn't seem to be any noticeable noise or implausible values among the continuous variables.**

[ln 8]

Inferences:

- **Gender (sex)**: The dataset is predominantly female, constituting a significant majority.
- **Type of Chest Pain (cp)**: The dataset shows varied chest pain types among patients. Type 0 (Typical angina) seems to be the most prevalent, but an exact distribution among the types can be inferred from the bar plots.
- **Fasting Blood Sugar (fbs)**: A significant majority of the patients have their fasting blood sugar level below 120 mg/dl, indicating that high blood sugar is not a common condition in this dataset.
- **Resting Electrocardiographic Results (restecg)**: The results show varied resting electrocardiographic outcomes, with certain types being more common than others. The exact distribution can be gauged from the plots.
- **Exercise-Induced Angina (exang)**: A majority of the patients do not experience exercise-induced angina, suggesting that it might not be a common symptom among the patients in this dataset.
- **Slope of the Peak Exercise ST Segment (slope)**: The dataset shows different slopes of the peak exercise ST segment. A specific type might be more common, and its distribution can be inferred from the bar plots.

- **Number of Major Vessels Colored by Fluoroscopy (ca)**: Most patients have fewer major vessels colored by fluoroscopy, with '0' being the most frequent.
- **Thalium Stress Test Result (thal)**: The dataset displays a variety of thalium stress test results. One particular type seems to be more prevalent, but the exact distribution can be seen in the plots.
- **Presence of Heart Disease (target)**: The **dataset is nearly balanced** in terms of heart disease presence, with about 54.5% having it and 45.5% not having it.

[ln 9 to ln 10]

For our **bivariate analysis** on the dataset's features with respect to the target variable:

- For **continuous data**: I am going to use **bar plots** to showcase the average value of each feature for the different target classes, and **KDE plots** to understand the distribution of each feature across the target classes. This aids in discerning how each feature varies between the two target outcomes.
- For **categorical data**: I am going to employ **100% stacked bar plots** to depict the proportion of each category across the target classes. This offers a comprehensive view of how different categories within a feature relate to the target.

Through these visualization techniques, we are going to gain a deeper understanding of the relationship between individual features and the target, revealing potential predictors for heart disease.

We are going to visualize each continuous feature against the target using two types of charts:

- **Bar plots** - showing the mean values.
- **KDE plots** - displaying the distribution for each target category.

Inferences:

- **Age (age)**: The distributions show a slight shift with patients having heart disease being a bit younger on average than those without. The mean age for patients without heart disease is higher.
- **Resting Blood Pressure (trestbps)**: Both categories display overlapping distributions in the KDE plot, with nearly identical mean values, indicating limited differentiating power for this feature.
- **Serum Cholesterol (chol)**: The distributions of cholesterol levels for both categories are quite close, but the mean cholesterol level for patients with heart disease is slightly lower.
- **Maximum Heart Rate Achieved (thalach)**: There's a noticeable difference in distributions. Patients with heart disease tend to achieve a higher maximum heart rate during stress tests compared to those without.

- **ST Depression (oldpeak)**: The ST depression induced by exercise relative to rest is notably lower for patients with heart disease. Their distribution peaks near zero, whereas the non-disease category has a wider spread.

Based on the visual difference in distributions and mean values, **Maximum Heart Rate (thalach)** seems to have the most impact on the heart disease status, followed by **ST Depression (oldpeak)** and **Age (age)**.

[ln 11]

We are going to display **100% stacked bar plots** for each categorical feature illustrating the proportion of each category across the two target classes, complemented by the exact counts and percentages on the bars.

Inferences:

- **Number of Major Vessels (ca)**: The majority of patients with heart disease have fewer major vessels colored by fluoroscopy. As the number of colored vessels increases, the proportion of patients with heart disease tends to decrease. Especially, patients with 0 vessels colored have a higher proportion of heart disease presence.
- **Chest Pain Type (cp)**: Different types of chest pain present varied proportions of heart disease. Notably, types 1, 2, and 3 have a higher proportion of heart disease presence compared to type 0. This suggests the type of chest pain can be influential in predicting the disease.
- **Exercise Induced Angina (exang)**: Patients who did not experience exercise-induced angina (0) show a higher proportion of heart disease presence compared to those who did (1). This feature seems to have a significant impact on the target.
- **Fasting Blood Sugar (fbs)**: The distribution between those with fasting blood sugar > 120 mg/dl (1) and those without (0) is relatively similar, suggesting fbs might have limited impact on heart disease prediction.
- **Resting Electrocardiographic Results (restecg)**: Type 1 displays a higher proportion of heart disease presence, indicating that this feature might have some influence on the outcome.
- **Sex (sex)**: Females (1) exhibit a lower proportion of heart disease presence compared to males (0). This indicates gender as an influential factor in predicting heart disease.
- **Slope of the Peak Exercise ST Segment (slope)**: The slope type 2 has a notably higher proportion of heart disease presence, indicating its potential as a significant predictor.

- **Thalium Stress Test Result (thal)**: The reversible defect category (2) has a higher proportion of heart disease presence compared to the other categories, emphasizing its importance in prediction.

In summary, based on the visual representation:

- **Higher Impact on Target: ca, cp, exang, sex, slope, and thal**
- **Moderate Impact on Target: restecg**
- **Lower Impact on Target: fbs**

[ln 12 to ln 13]

All features in the dataset appear to be relevant based on our **EDA**. No columns seem redundant or irrelevant. Thus, we'll retain all features, ensuring no valuable information is lost, especially given the dataset's small size.

Upon our above inspection, it is obvious that there are no missing values in our dataset. This is ideal as it means we don't have to make decisions about imputation or removal, which can introduce bias or reduce our already limited dataset size.

[ln 14]

We are going to check for outliers using the IQR method for the continuous features:

Upon identifying outliers for the specified continuous features, we found the following:

- **trestbps**: 9 outliers
- **chol**: 5 outliers
- **thalach**: 1 outlier
- **oldpeak**: 5 outliers
- **age**: No outliers

Sensitivity to Outliers:

- **SVM (Support Vector Machine)**: SVMs can be sensitive to outliers. While the decision boundary is determined primarily by the support vectors, outliers can influence which data points are chosen as support vectors, potentially leading to suboptimal classification.
- **Decision Trees (DT) and Random Forests (RF)**: These tree-based algorithms are generally robust to outliers. They make splits based on feature values, and outliers

often end up in leaf nodes, having minimal impact on the overall decision-making process.

- **K-Nearest Neighbors (KNN)**: KNN is sensitive to outliers because it relies on distances between data points to make predictions. Outliers can distort these distances.
- **AdaBoost:** This ensemble method, which often uses decision trees as weak learners, is generally robust to outliers. However, the iterative nature of AdaBoost can sometimes lead to overemphasis on outliers, making the final model more sensitive to them.

Approaches for Outlier Treatment:

- **Removal of Outliers**: Directly discard data points that fall outside of a defined range, typically based on a method like the Interquartile Range (IQR).
- **Capping Outliers**: Instead of removing, we can limit outliers to a certain threshold, such as the 1st or 99th percentile.
- **Transformations**: Applying transformations like log or Box-Cox can reduce the impact of outliers and make the data more Gaussian-like.
- **Robust Scaling**: Techniques like the RobustScaler in Scikit-learn can be used, which scales features using statistics that are robust to outliers.

Conclusion:

Given **the nature of the algorithms (especially SVM and KNN)** and **the small size of our dataset**, direct removal of outliers might not be the best approach. Instead, **we'll focus on applying transformations like Box-Cox in the subsequent steps** to reduce the impact of outliers and make the data more suitable for modeling.

[ln 15 to ln 16]

One-hot Encoding Decision:

Based on the feature descriptions, let's decide on one-hot encoding:

1. **Nominal Variables**: These are variables with no inherent order. They should be one-hot encoded because using them as numbers might introduce an unintended order to the model.
2. **Ordinal Variables**: These variables have an inherent order. They don't necessarily need to be one-hot encoded since their order can provide meaningful information to the model.

Given the above explanation:

- **sex**: This is a binary variable with two categories (male and female), so it doesn't need one-hot encoding.

- **cp**: Chest pain type can be considered as nominal because there's no clear ordinal relationship among the different types of chest pain (like Typical angina, Atypical angina, etc.). It should be one-hot encoded.
- **fbs**: This is a binary variable (true or false), so it doesn't need one-hot encoding.
- **restecg**: This variable represents the resting electrocardiographic results. The results, such as "Normal", "Having ST-T wave abnormality", and "Showing probable or definite left ventricular hypertrophy", don't seem to have an ordinal relationship. Therefore, it should be one-hot encoded.
- **exang**: This is a binary variable (yes or no), so it doesn't need one-hot encoding.
- **slope**: This represents the slope of the peak exercise ST segment. Given the descriptions (Upsloping, Flat, Downsloping), it seems to have an ordinal nature, suggesting a particular order. Therefore, it doesn't need to be one-hot encoded.
- **ca**: This represents the number of major vessels colored by fluoroscopy. As it indicates a count, it has an inherent ordinal relationship. Therefore, it doesn't need to be one-hot encoded.
- **thal**: This variable represents the result of a thalium stress test. The different states, like "Normal", "Fixed defect", and "Reversible defect", suggest a nominal nature. Thus, it should be one-hot encoded.

Summary:

- **Need One-Hot Encoding**: **cp**, **restecg**, **thal**
- **Don't Need One-Hot Encoding**: **sex**, **fbs**, **exang**, **slope**, **ca**

[ln 17 to ln 18]

Feature Scaling is a crucial preprocessing step **for algorithms that are sensitive to the magnitude or scale of features**. Models like **SVM**, **KNN**, and many linear models rely on distances or gradients, making them susceptible to variations in feature scales. **Scaling ensures that all features contribute equally to the model's decision rather than being dominated by features with larger magnitudes.**

Why We Skip It Now:

While feature scaling is vital for some models, not all algorithms require scaled data. For instance, **Decision Tree-based models** are scale-invariant. Given our intent to use a mix of models (some requiring scaling, others not), **we've chosen to handle scaling later using pipelines**. This approach lets us apply scaling specifically for models that benefit from it, ensuring flexibility and efficiency in our modeling process.

Box-Cox transformation is a powerful method to stabilize variance and make the data more normal-distribution-like. It's particularly useful when you're unsure about the exact nature of the distribution you're dealing with, as it can adapt itself to the best power transformation. However, the Box-Cox transformation only works for positive data, so one must be cautious when applying it to features that contain zeros or negative values.

linkcode

Transforming Skewed Features & Data Leakage Concerns:

When preprocessing data, especially applying transformations like the Box-Cox, it's essential to be wary of **data leakage**. **Data leakage** refers to a mistake in the preprocessing of data in which information from outside the training dataset is used to transform or train the model. This can lead to overly optimistic performance metrics.

To avoid data leakage and ensure our model generalizes well to unseen data:

1- Data Splitting: We'll first split our dataset into a training set and a test set. This ensures that we have a separate set of data to evaluate our model's performance, untouched during the training and preprocessing phases.

2- Box-Cox Transformation: We'll examine the distribution of the continuous features in the training set. If they appear skewed, we'll apply the Box-Cox transformation to stabilize variance and make the data more normal-distribution-like. Importantly, we'll determine the Box-Cox transformation parameters solely based on the training data.

3- Applying Transformations to Test Data: Once our transformation parameters are determined from the training set, we'll use these exact parameters to transform our validation/test set. This approach ensures that no information from the validation/test set leaks into our training process.

4. Hyperparameter Tuning & Cross-Validation: Given our dataset's size, to make the most of the available data during the model training phase, we'll employ **cross-validation on the training set for hyperparameter tuning**. This allows us to get a better sense of how our model might perform on unseen data, without actually using the test set. The test set remains untouched during this phase and is only used to evaluate the final model's performance.

By following this structured approach, we ensure a rigorous training process, minimize the risk of data leakage, and set ourselves up to get a realistic measure of our model's performance on unseen data.

[ln 19 to ln 20]

The Box-Cox transformation requires all data to be strictly positive. To transform the oldpeak feature using Box-Cox, we can add a small constant (e.g., 0.001) to ensure all values are positive:

[ln 21]

Inference:

1- age: The transformation has made the age distribution more symmetric, bringing it closer to a normal distribution.

2- Trestbps: The distribution of trestbps post-transformation appears to be more normal-like, with reduced skewness.

3- Chol: After applying the Box-Cox transformation, chol exhibits a shape that's more aligned with a normal distribution.

4- Thalach: The thalach feature was already fairly symmetric before the transformation, and post-transformation, it continues to show a similar shape, indicating its original distribution was close to normal.

5- Oldpeak: The transformation improved the oldpeak distribution, but it still doesn't perfectly resemble a normal distribution. This could be due to the inherent nature of the data or the presence of outliers. To enhance its normality, we could consider utilizing advanced transformations such as the Yeo-Johnson transformation, which can handle zero and negative values directly.

Conclusion:

Transforming features to be more normal-like primarily helps in mitigating the impact of outliers, which is particularly beneficial for distance-based algorithms like SVM and KNN. By reducing the influence of outliers, we ensure that these algorithms can compute distances more effectively and produce more reliable results.

[ln 22 to ln 23]

First, let's define the base DT model:

[ln 25]

Note: In medical scenarios, especially in the context of diagnosing illnesses, it's often more important **to have a high recall (sensitivity) for the positive class (patients with the condition)**. A high recall ensures that most of the actual positive cases are correctly identified, even if it means some false positives (cases where healthy individuals are misclassified as having the condition). The rationale is that it's generally better to have a few false alarms than to miss out on diagnosing a patient with a potential illness.

I am establishing a function to determine the optimal set of hyperparameters that yield the highest **recall** for the model. This approach ensures a reusable framework for hyperparameter tuning of subsequent models:

[ln 26]

We’ll set up the hyperparameters grid and utilize the **tune_clf_hyperparameters** function to pinpoint the optimal hyperparameters for our DT model:

[ln 27 to ln 29]

Now let's evaluate our DT model performance on both the training and test datasets:

[ln 30 to ln 31]

Given that the metric values for both the training and test datasets are closely aligned and not significantly different, the model doesn't appear to be overfitting.

Let's create a function that consolidates each model's metrics into a dataframe, facilitating an end-to-end comparison of all models later:

[ln 32 to ln 33]

First, let's define the base RF model:

[ln 34]

Afterward, We are setting up the hyperparameters grid and utilize the tune_clf_hyperparameters function to pinpoint the optimal hyperparameters for our RF model:

[ln 35 to ln 36]

Finally, we are evaluating the model's performance on both the training and test datasets:

[ln 37 to ln 38]

The RF model's similar performance on both training and test data suggests it isn't overfitting.

[ln 39]

First of all, let's define the base KNN model and set up the pipeline with scaling:

[ln 40]

Let's evaluate the model's performance on both the training and test datasets:

[ln 43 to ln 44]

The KNN model's consistent scores across training and test sets indicate no overfitting.

[ln 45]

First, let's define the base SVM model and set up the pipeline with scaling:

[ln 46]

Let's configure the hyperparameters grid and employ the tune_clf_hyperparameters function to determine the best hyperparameters for our SVM pipeline:

[ln 47 to ln 48]

Let's evaluate our SVM model's performance on both the training and test datasets:

[ln 49 to ln 50]

Inference:

The recall of 0.97 for class 1 indicates that almost all the true positive cases (i.e., patients with heart disease) are correctly identified. This high recall is of utmost importance in a medical context, where missing a patient with potential heart disease could have dire consequences.

However, it's also worth noting the balanced performance of the model. With an F1-score of 0.83 for class 1, it's evident that the model doesn't merely focus on maximizing recall at the expense of precision. This means the reduction in False Negatives hasn't significantly increased the False Positives, ensuring that the cost and effort of examining healthy individuals are not unnecessarily high.

Overall, the model's performance is promising for medical diagnostics, especially when prioritizing the accurate identification of patients with heart disease without overburdening the system with false alarms.

[ln 51]

In the critical context of diagnosing heart disease, our primary objective is to ensure a high recall for the positive class. It's imperative to accurately identify every potential heart disease case, as even one missed diagnosis could have dire implications. However, while striving for this high recall, it's essential to maintain a balanced performance to avoid unnecessary medical interventions for healthy individuals. We'll now evaluate our models against these crucial medical benchmarks.

[ln 52 to ln 53]

The SVM model demonstrates a commendable capability in recognizing potential heart patients. With a recall of 0.97 for class 1, it's evident that almost all patients with heart disease are correctly identified. This is of paramount importance in a medical setting. However, the model's balanced performance ensures that while aiming for high recall, it doesn't compromise on precision, thereby not overburdening the system with unnecessary alerts.