Heart disease Prediction Model

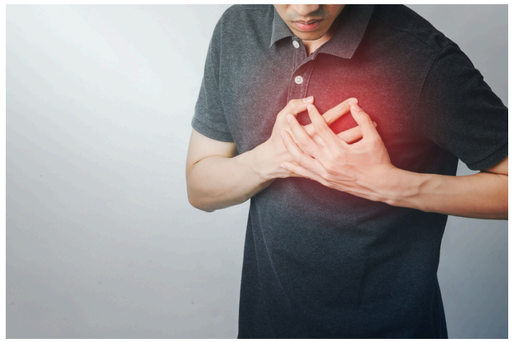
by Harish Muhammad

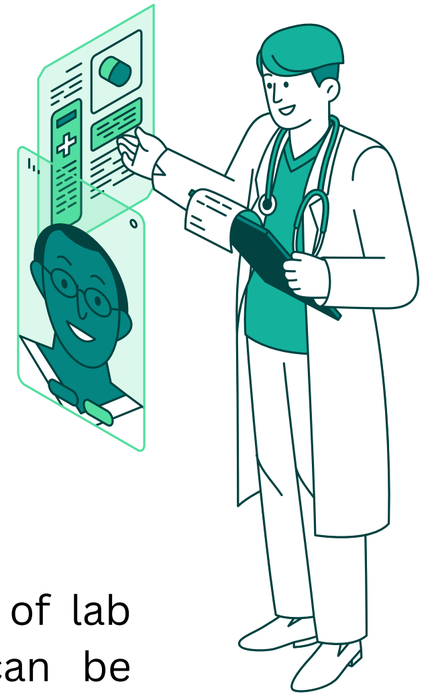
Outlines

- Business problem and background
- Business objective (Menentukan objektif bisnis)
- Technical data science objective (Menentukan tujuan teknis data science)
- Examining data (Menelaah data)
- Validate data (Memvalidasi data)
- Determine object data (Menentukan objek data)
- Data construction (Mengkonstruksi data)
- Building a model scenario (Membangun skenario model)
- Model building (Membangun model)
- Model evaluation (Mengevaluasi hasil pemodelan)
- Reviewing the modeling process (Melakukan review pemodelan)

BUSINESS PROBLEM UNDERSTANDING

Background





- Heart disease is the leading mortality
- According to the WHO, this disease causes up to 17.9 M global mortality on an annual basis.
- **Delayed** or **missed diagnosis** can lead to critical health outcomes and high medical costs.

- Hospitals and clinics often rely on a mix of lab results and doctor's judgment, which can be inconsistent and time-constrained.

Business Problem:

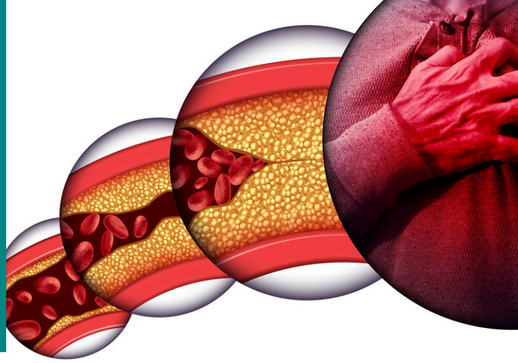
- How can we reduce missed diagnoses of patients who are at risk of heart disease?

BUSINESS OBJECTIVE

🎯 Objective:

- Build a supporting diagnostic tool that can assist medical staff (cardiologists) by flagging high-risk patients.
- This tool aims to:
  - Improve early detection rates.
  - Reduce false negative diagnoses.
  - Prioritize patients for further testing or immediate care.

Business objective



💼 Stakeholders:

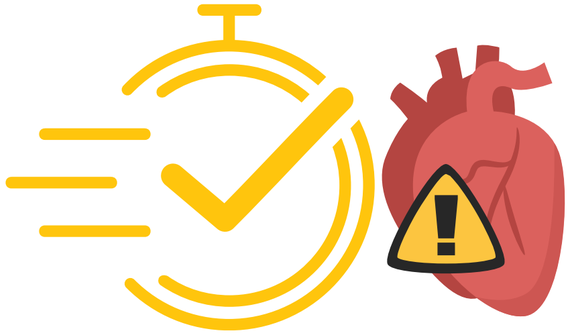
- Hospital managers (efficiency & cost).
- Cardiologists (accuracy in diagnosis).
- Patients (life-saving detection).

📊 Success Indicators:

- High recall for detecting actual heart disease cases.
- Practical interpretability to support doctors, not confuse them.
- Potential for deployment into hospital decision support systems.

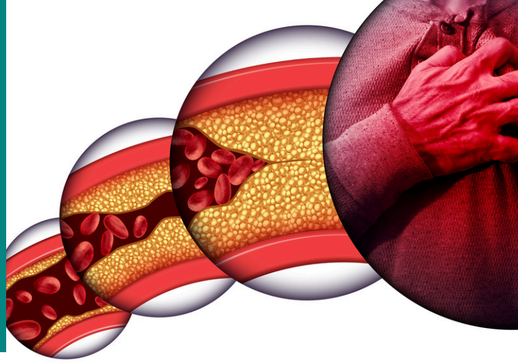
Easy & practical

Cost effective



Fast

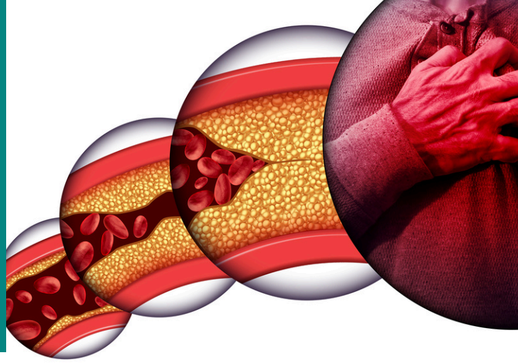
Technical Data Science objective: Predicting Heart Disease Risk



🔢 Technical objective:

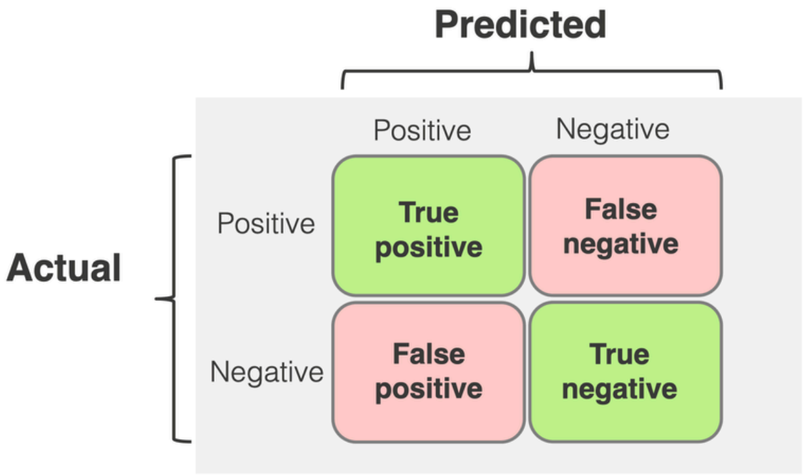
- Build a binary classification model to predict whether a patient has heart disease (num: 0 or 1) using 13 clinical features.
- Focus on maximizing **recall** to catch as many true positive heart disease cases as possible.
- Highlighting important features that contribute to the model.

Technical Data Science objective: Predicting Heart Disease Risk



🎯 Target:

- 0 : Patients who dont have heart diseases
- 1 : Patients who have heart diseases



Based on the model, we will have two kinds of errors

Type 1 error: False Positive

- Patients are predicted to have heart disease. However, in actual conditions, they do not have heart diseases
- Consequences: The patients need to conduct further medical examination, but only for assessment or verification to confirm. Due to the mistake, the image of the hospital & app developer became less reliable to the patients and the public.

T**ype 2 error: False Negative**

- Patients are predicted as not having heart disease. However, in actual conditions, they do have heart diseases.
- Consequences: Patient conditions may become worse. If they do manage to get diagnosed, their heart disease treatments are more likely to be more difficult and far more expensive. Even, they may have a probability of dying before being treated.

EXAMINING THE DATA (MENELAAN DATA)

Data understanding

Dataset Source

UC Irvine Machine Learning Repository

Heart Disease
Donated on 6/30/1988

4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Health and Medicine	Classification
Feature Type	**# Instances**	**# Features**
Categorical, Integer, Real	303	13

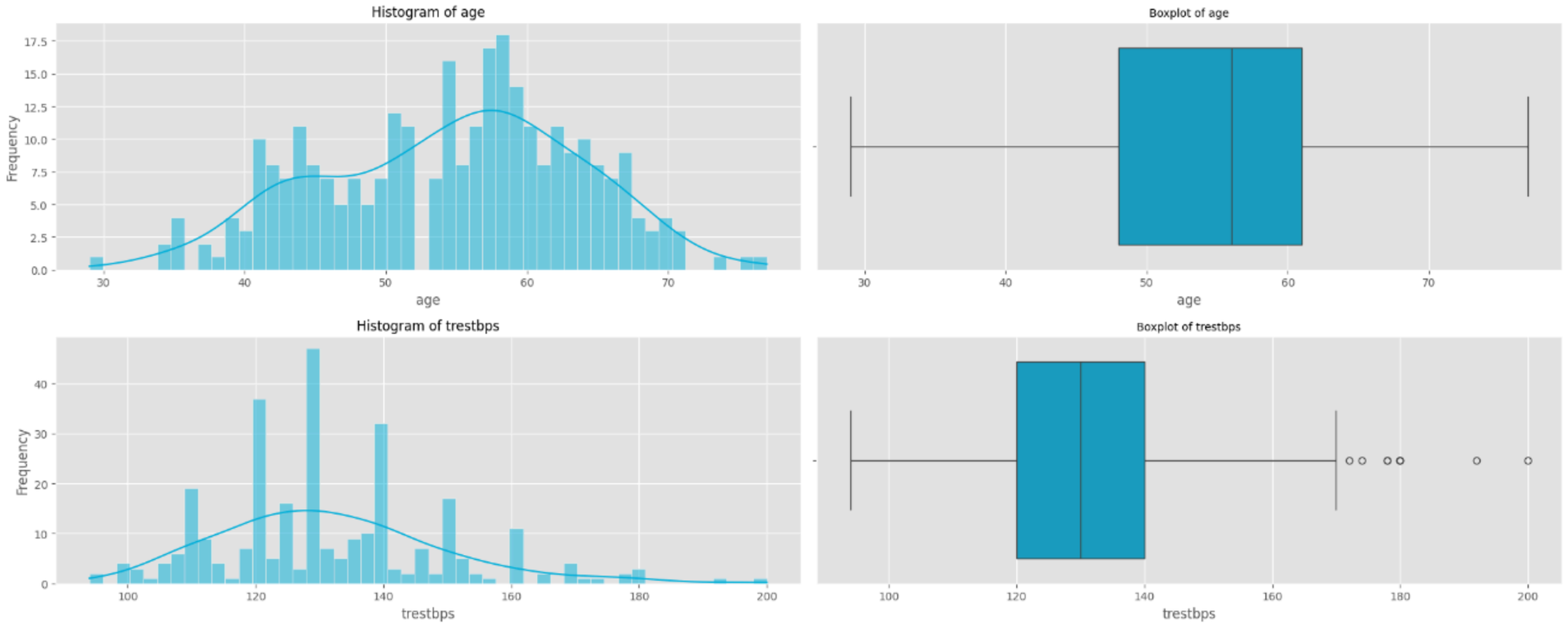
- Source: UCI Machine Learning Repository – Heart Disease Dataset (Cleveland subset).
- Size: 303 records, 14 selected attributes.
- Mix of numeric and categorical variables.

13 features

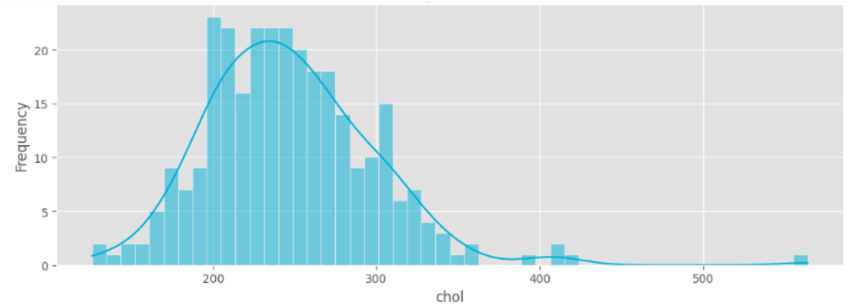
Target

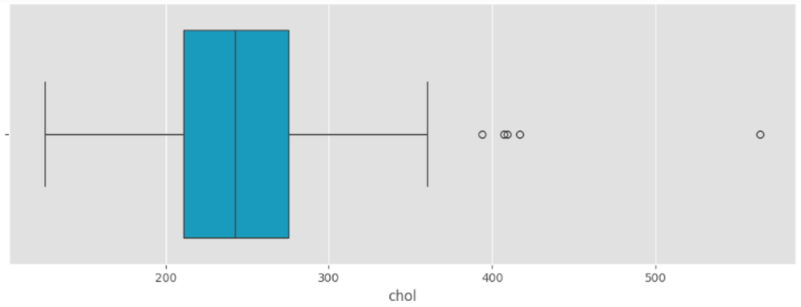
Feature	Description
age	The age of the patient measured in years.
sex	The gender of the patient with a value of 1 for male and 0 for female.
cp	The type of chest pain perceived by the patient with 4 possible category values:
	(1) indicates typical angina chest pain
	(2) indicates atypical angina chest pain
	(3) indicates non-anginal pain
	(4) indicates asymptomatic chest pain.
trestbps	Resting blood pressure or The patient's blood pressure at rest, measured in mmHg (millimeters of mercury).
chol	The serum cholesterol level in the patient's blood, measured in mg/dl (milligrams per deciliter).
fbs	The patient's fasting blood sugar level with a value of 1 if blood sugar level > 120 mg/dl and a value of 0 otherwise.
restecg	The resting electrocardiographic results of patients with 3 possible category values:
	(0) indicates normal results
	(1) indicates ST-T wave abnormality
	(2) indicates left ventricular hypertrophy.
thalach	The maximum heart rate achieved by the patient during exercise testing, measured in bpm (beats per minute).
exang	Exercise induced angina, This variable represents whether the patient experienced exercise-induced angina (chest pain)
	(0) Patients did not experiencing exercise-induced angina
	(1) Patients experienced exercise-induced angina
oldpeak	The amount of ST segment depression during physical activity compared to rest.
slope	The slope of the ST segment on the electrocardiogram (EKG) during maximal exercise with 3 category values.
ca	The number of major blood vessels (0-3) visible on fluoroscopy examination.
thal	The result of the thallium scan test with 3 possible category values:
	(1) indicates normal condition.
	(2) indicates fixed defect in thalassemia.
	(3) indicates reversible defect in thalassemia.
Target	The target result of the prediction test
	(0) Predicted with No heart disease
	(1) Predicted with Heart disease

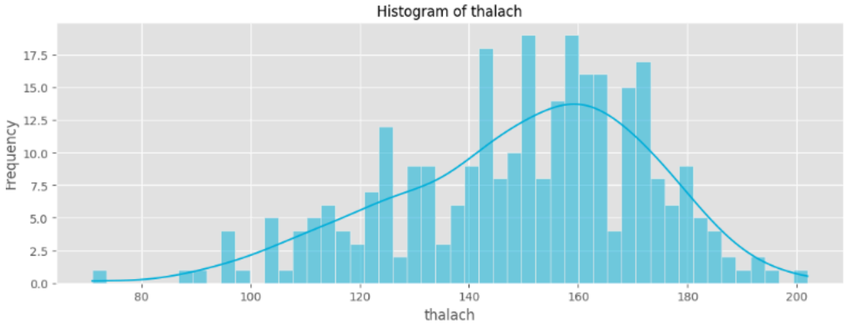
NUMERICAL - FEATURES

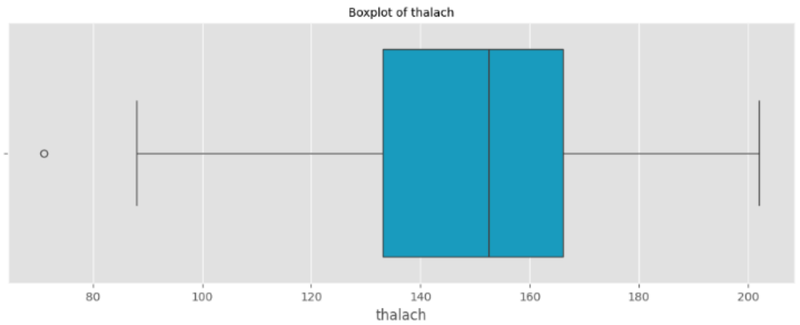


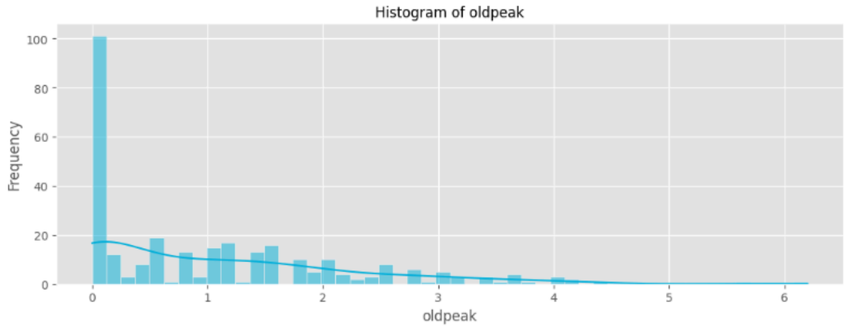
NUMERICAL - FEATURES

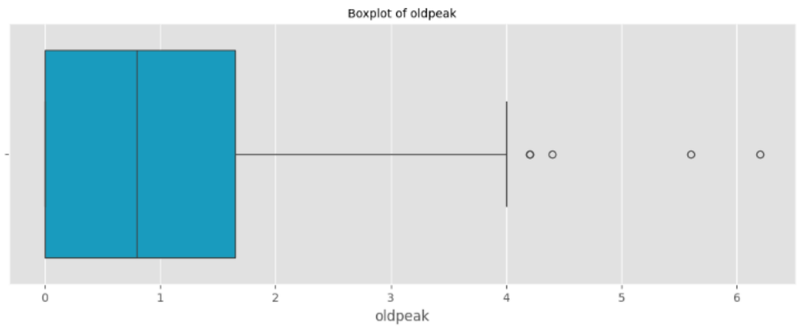




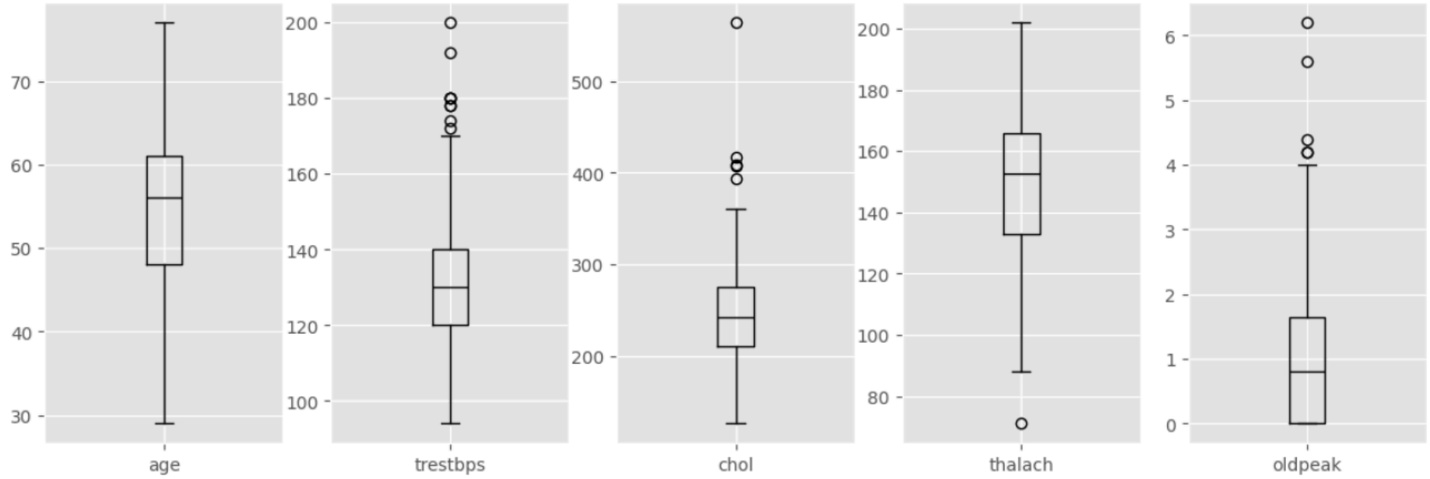








OUTLIERS



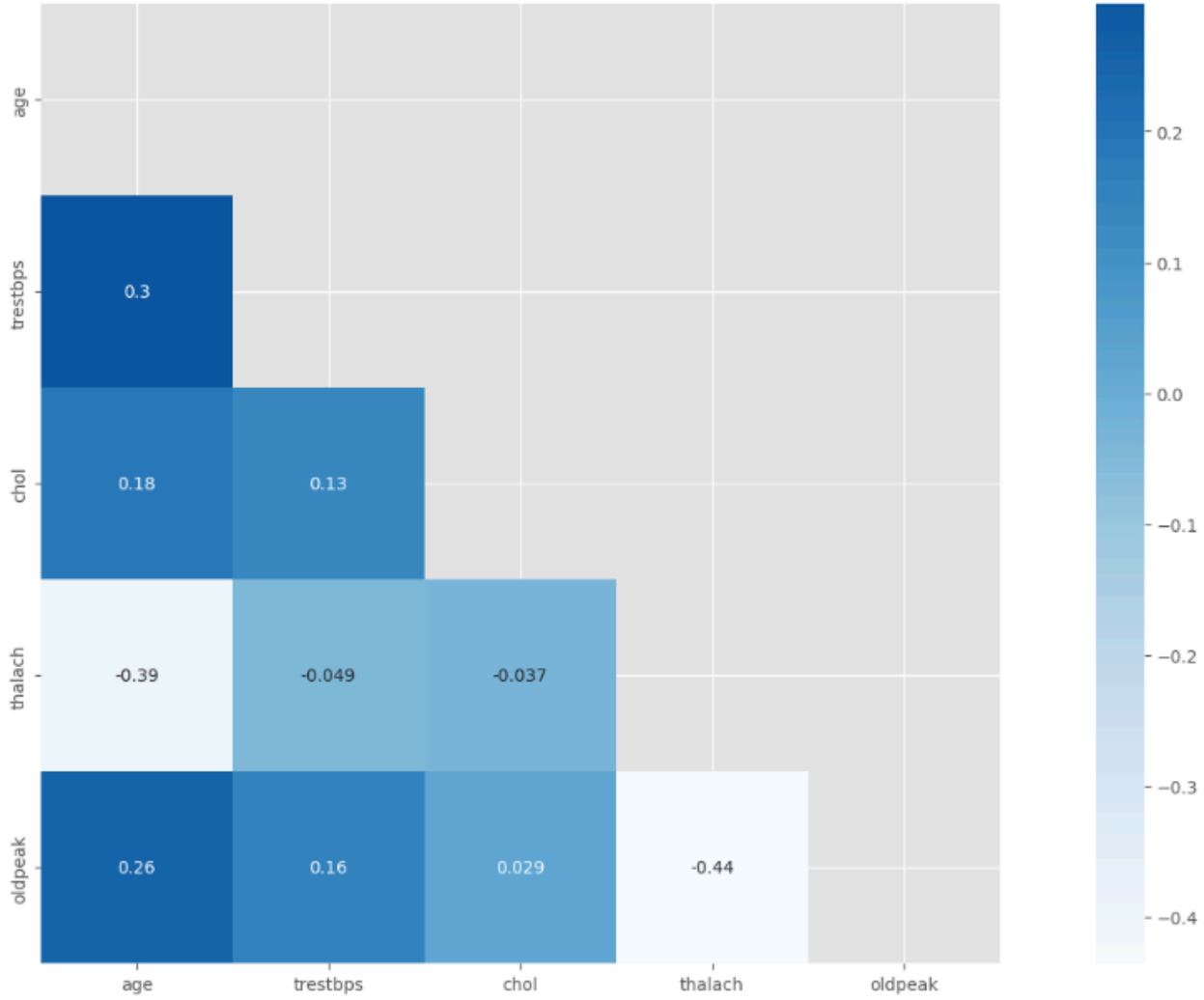
	Columns	Number of Outliers	(%) of Outliers	Lower bound	Upper bound
0	age	0	0.00	28.500	80.500
1	trestbps	9	3.04	90.000	170.000
2	chol	5	1.69	114.625	371.625
3	thalach	1	0.34	83.500	215.500
4	oldpeak	5	1.69	-2.475	4.125

DATA DISTRIBUTION

	Feature	D'Agostino-Pearson Statistic	P-value	Distribution	Skewness	Skewness Type
0	age	32.332197	9.531311e-08	Not Normally Distributed	-0.248866	Left Skew
1	trestbps	97.463126	6.857268e-22	Not Normally Distributed	0.739768	Right Skew
2	chol	236.904517	3.604439e-52	Not Normally Distributed	1.074073	Right Skew
3	thalach	41.002935	1.248319e-09	Not Normally Distributed	-0.513777	Left Skew
4	oldpeak	193.533646	9.434150e-43	Not Normally Distributed	1.210899	Right Skew

NUMERICAL - EDA

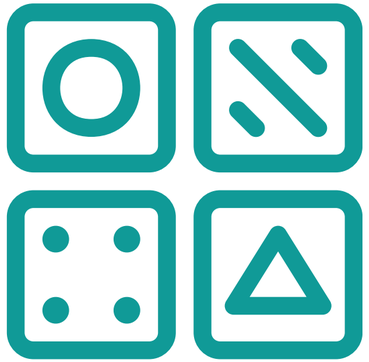
Spearman - correlation heatmap



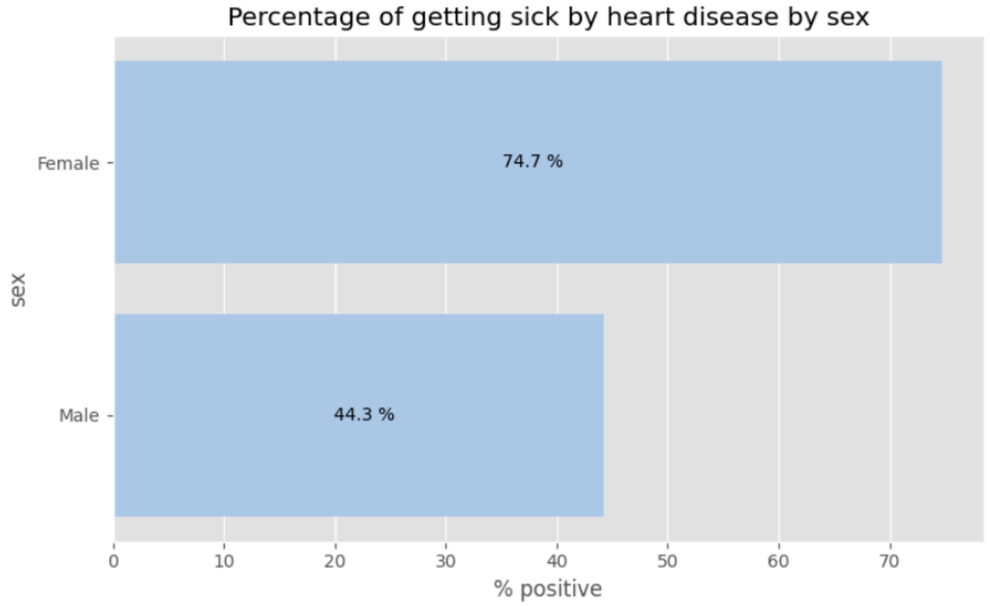
Exploratory Data Analysis (EDA)

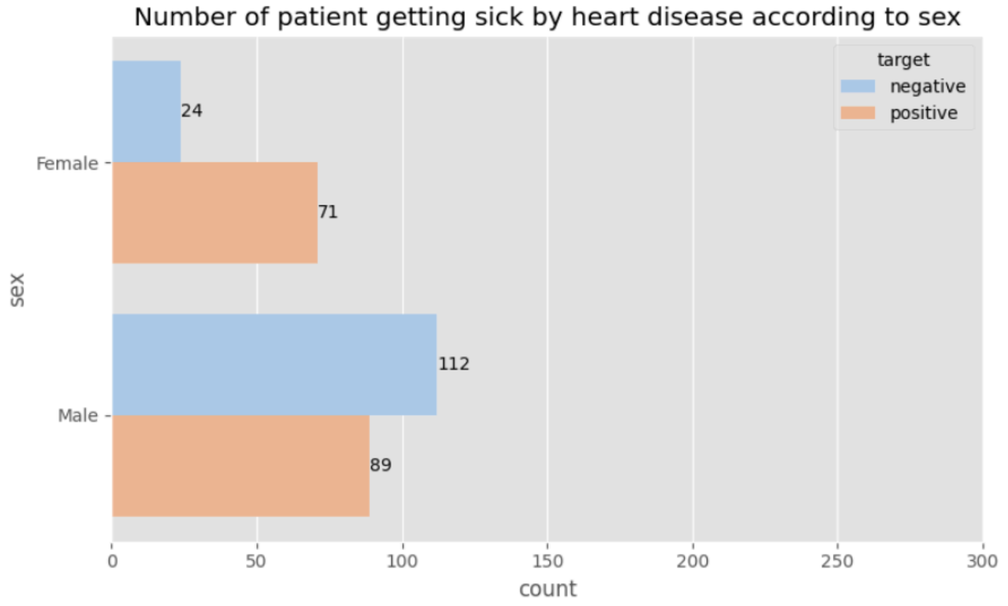
Numerical vs Target

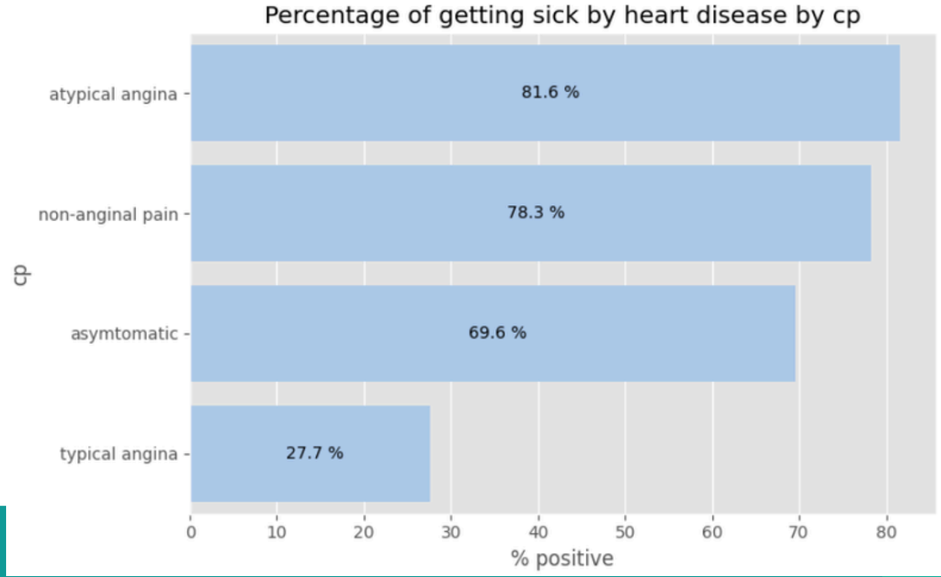
Categorical vs Target

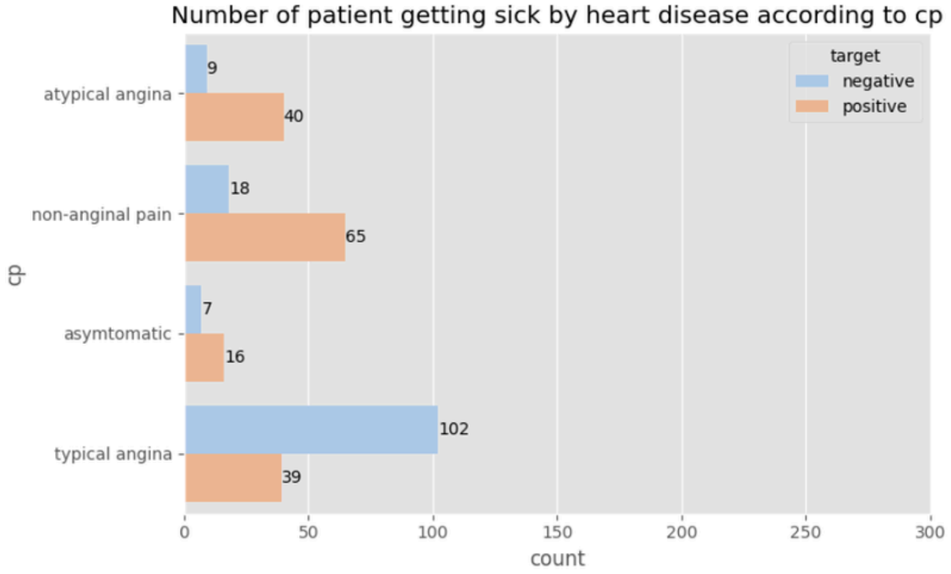


CATEGORICAL FEATURES

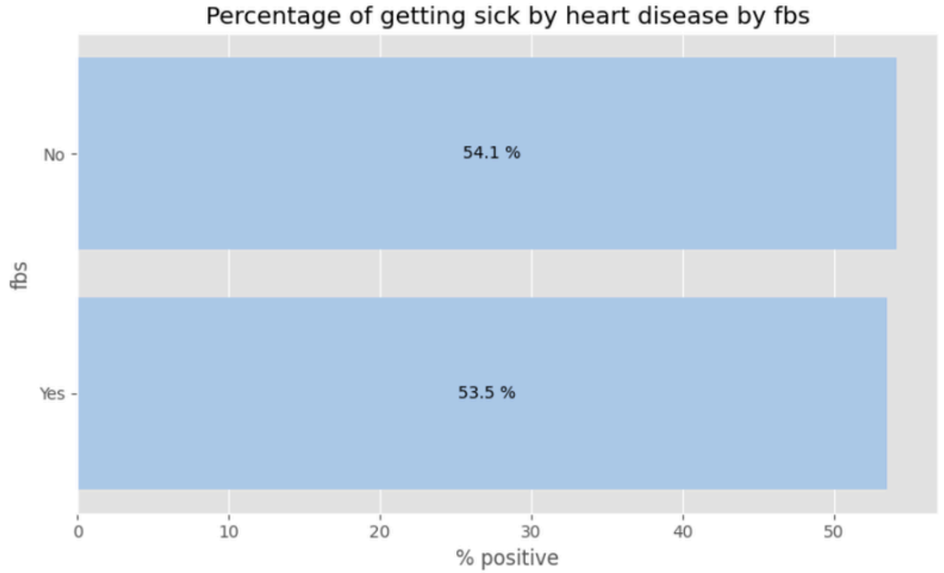


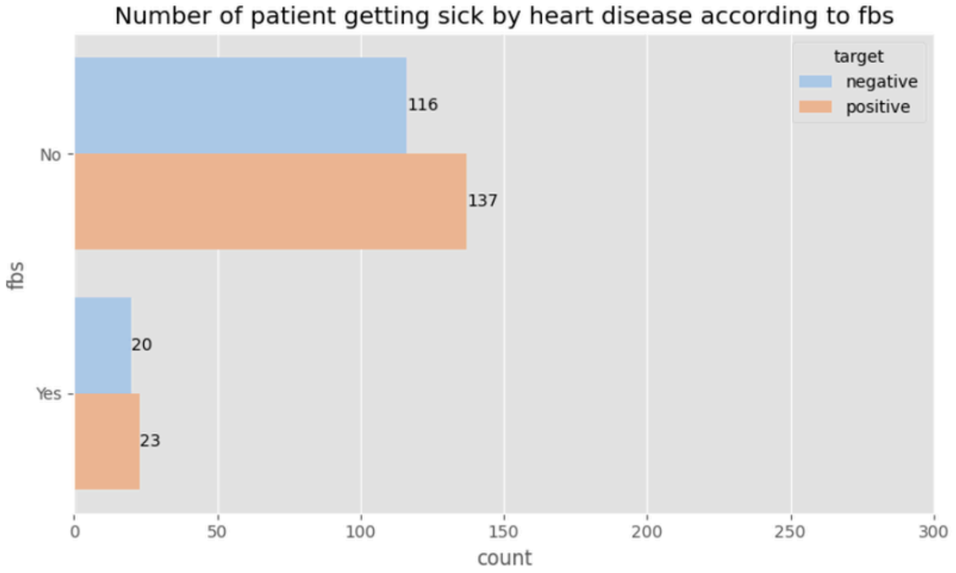


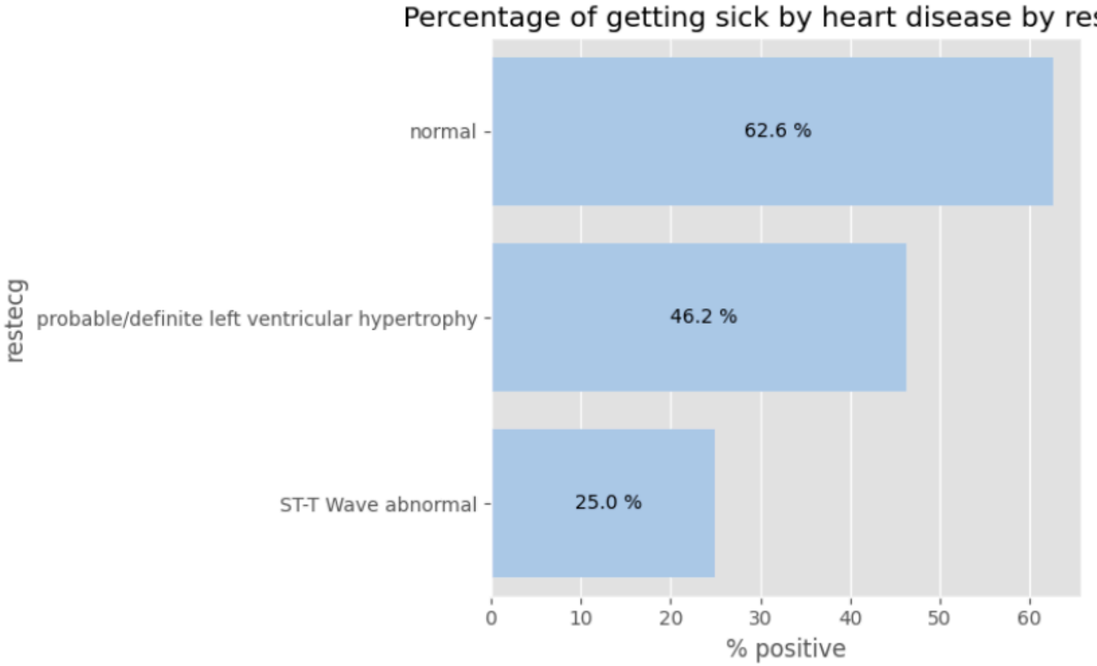


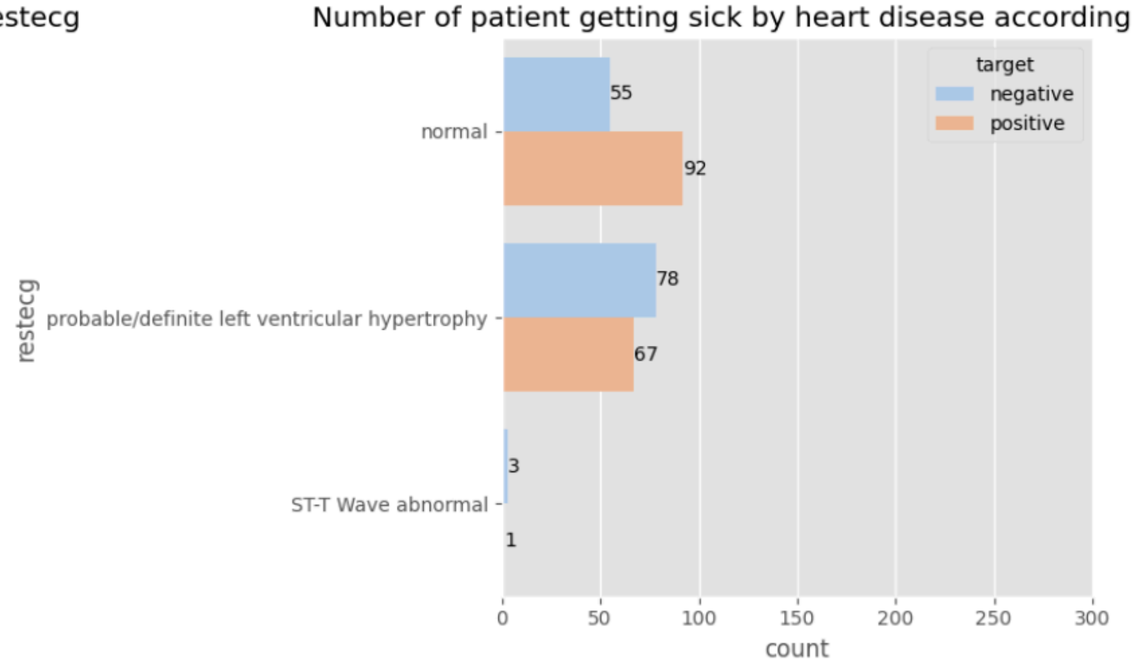


CATEGORICAL FEATURES

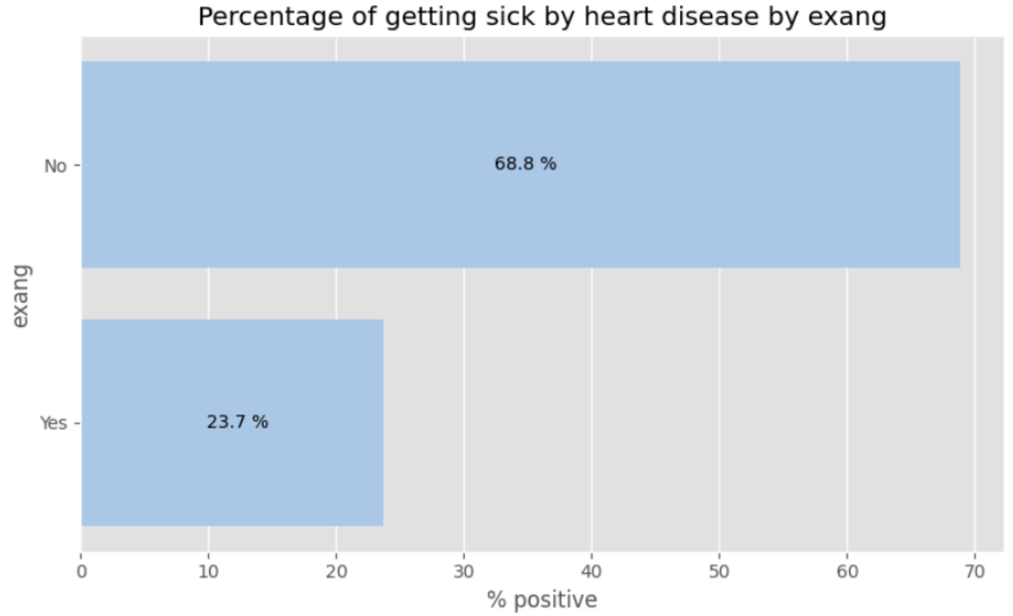


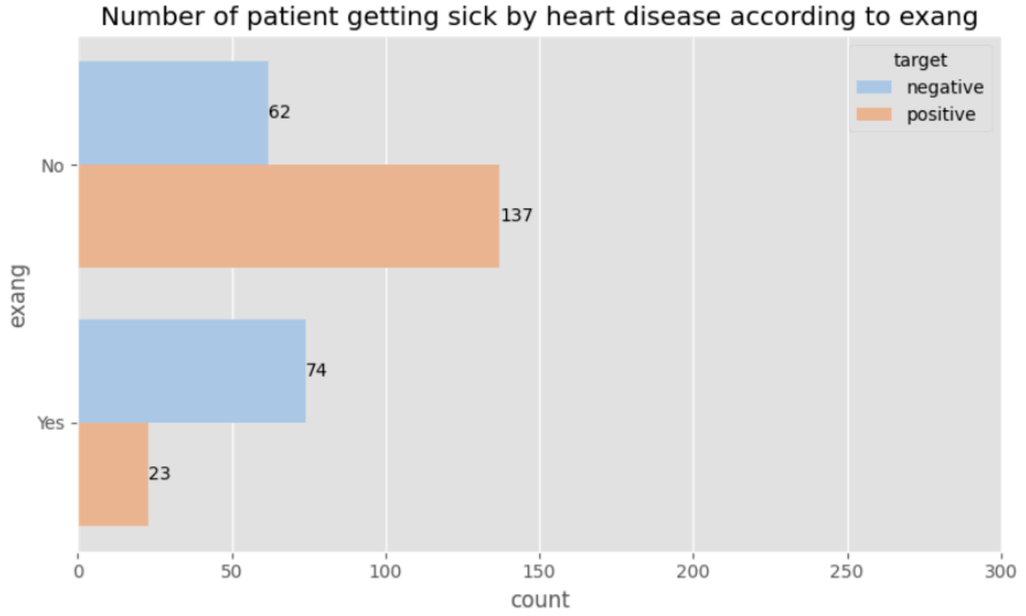


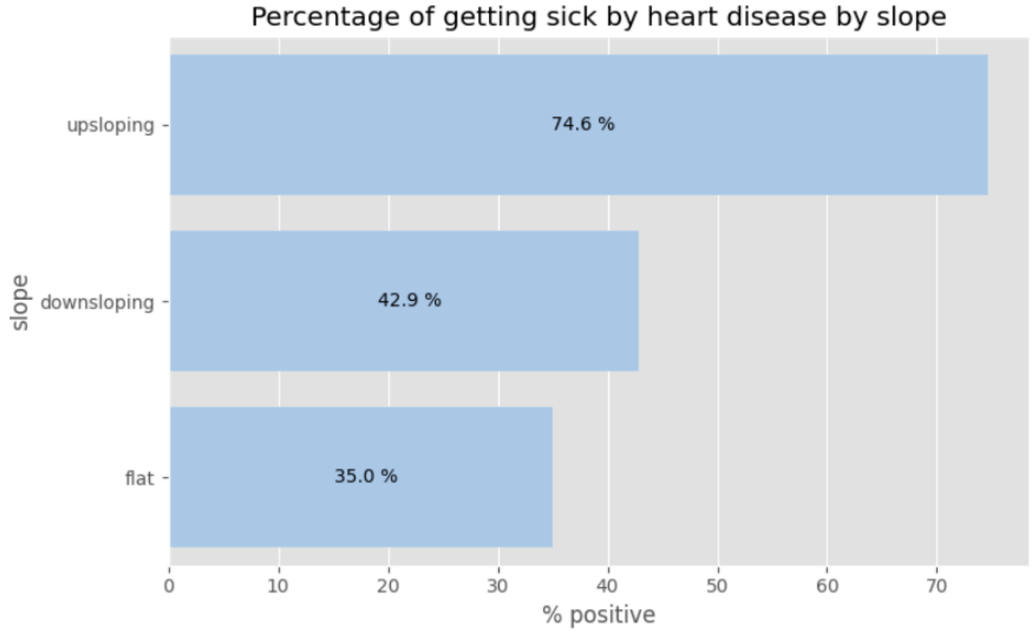


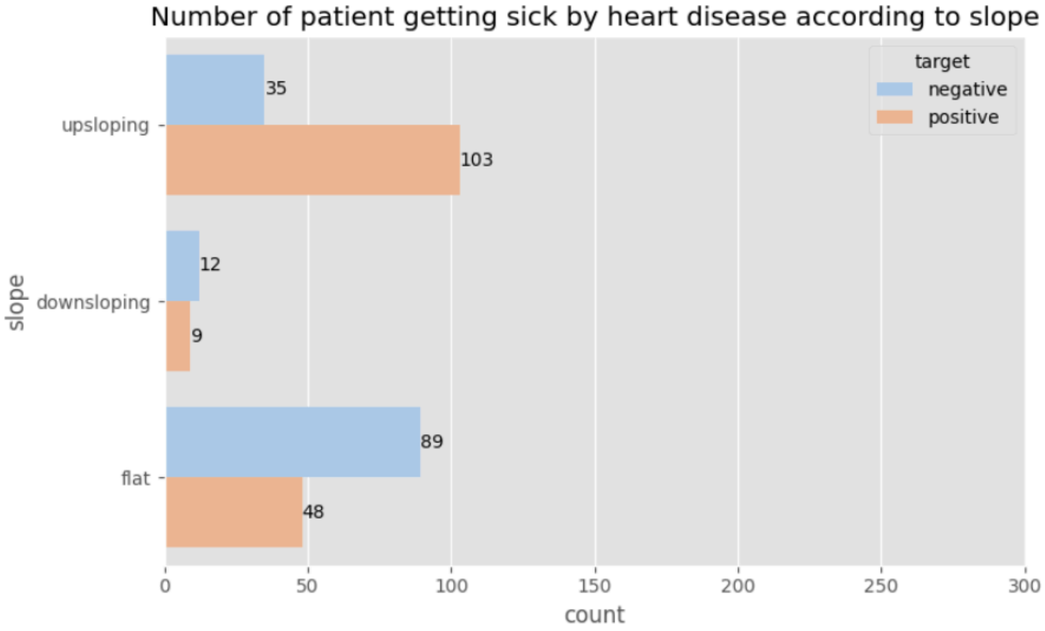


CATEGORICAL FEATURES

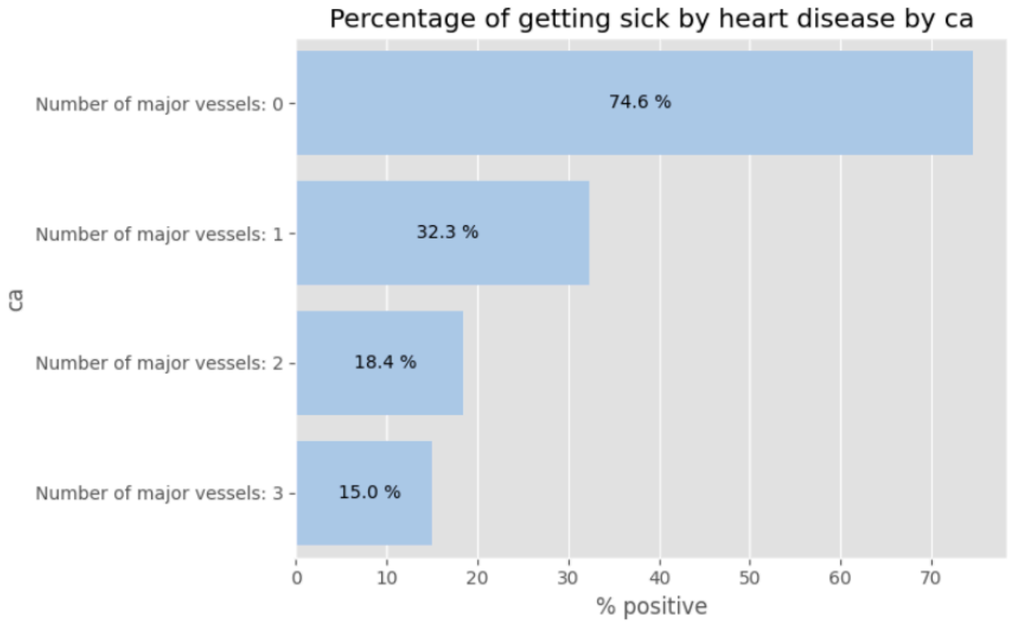


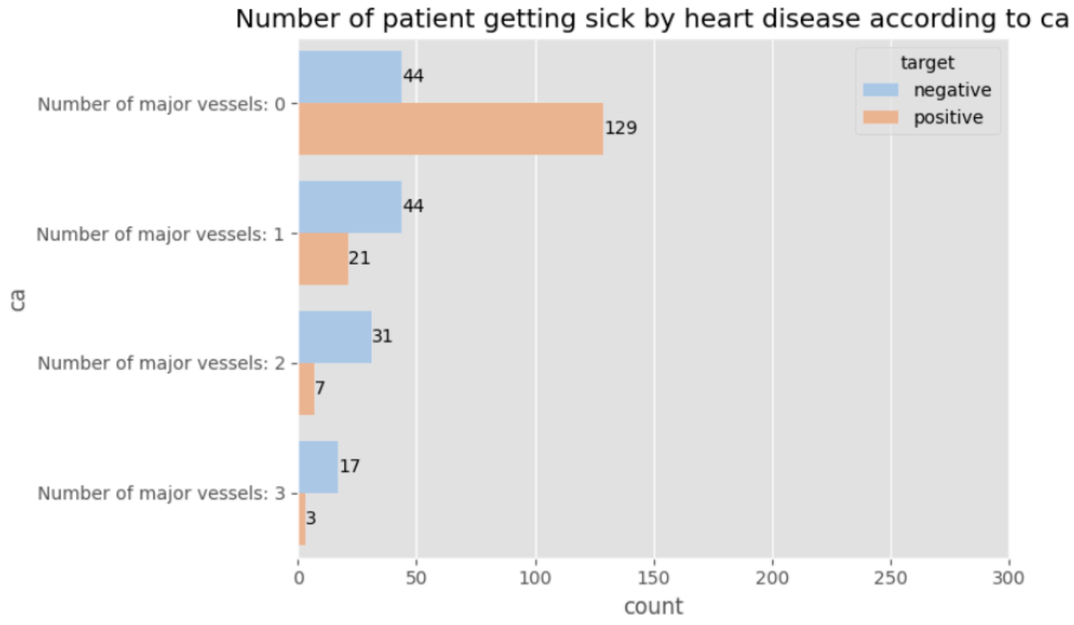


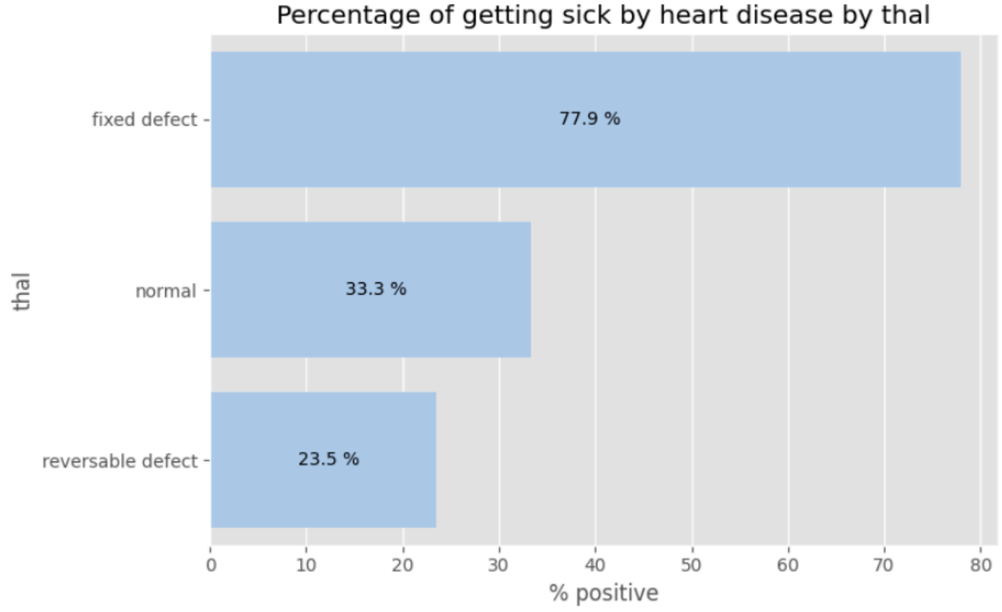


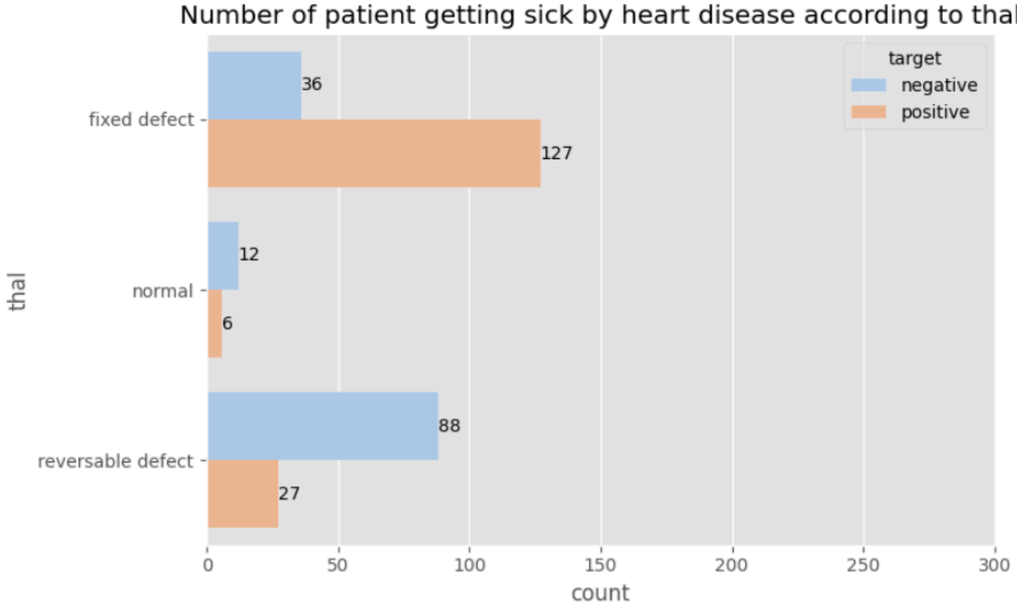


CATEGORICAL FEATURES

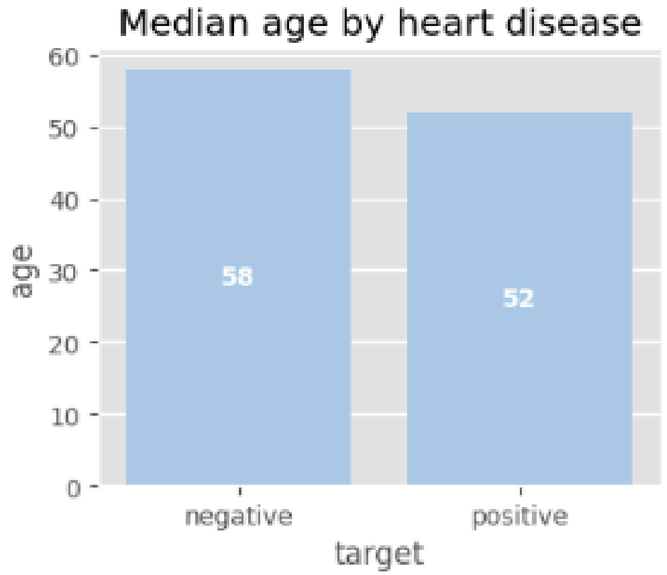


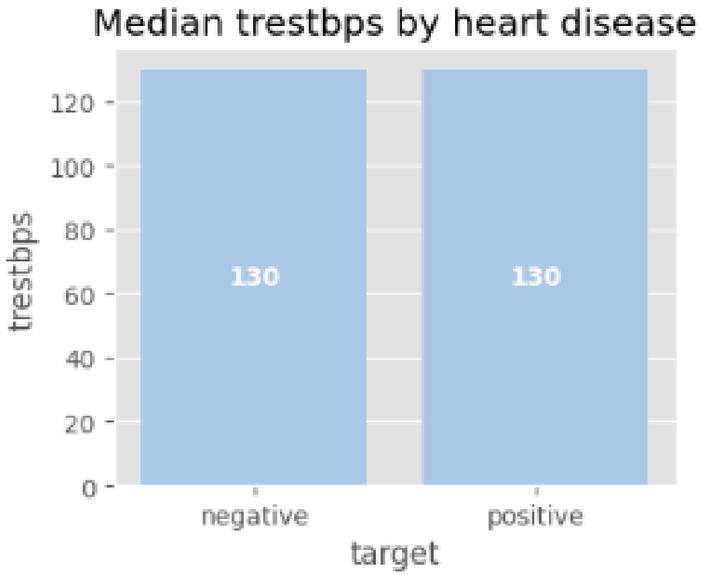


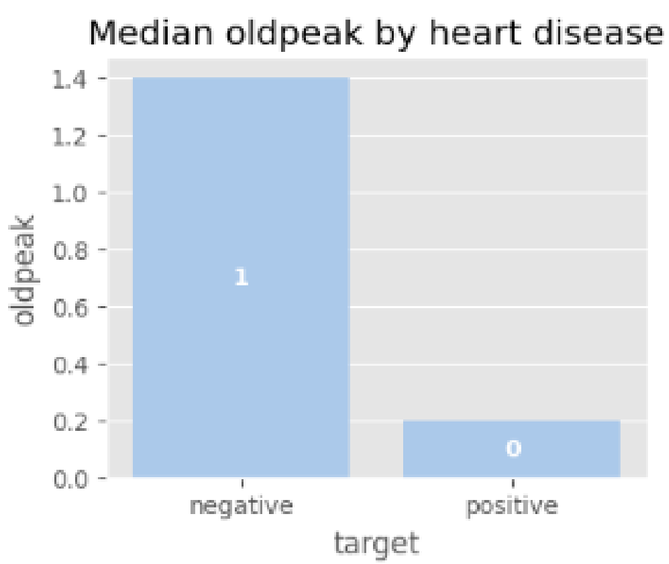


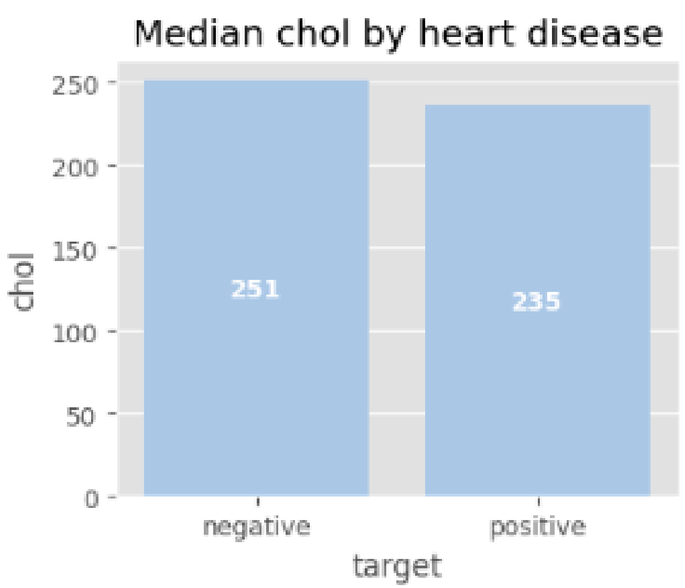


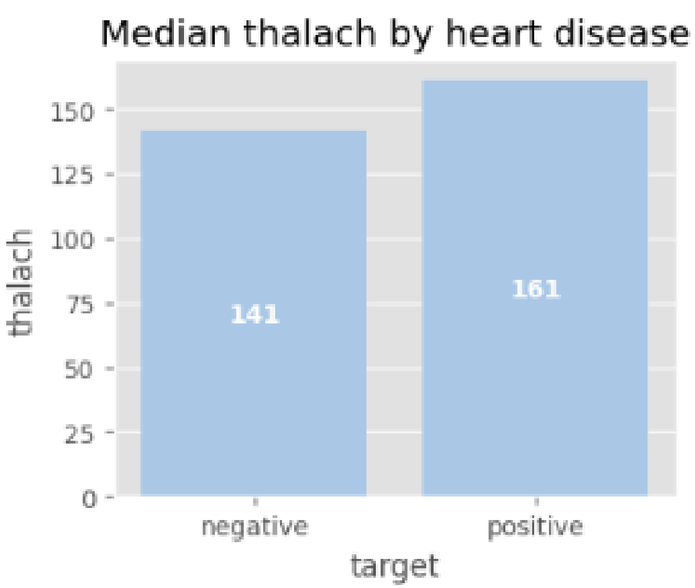
NUMERICAL VS TARGET



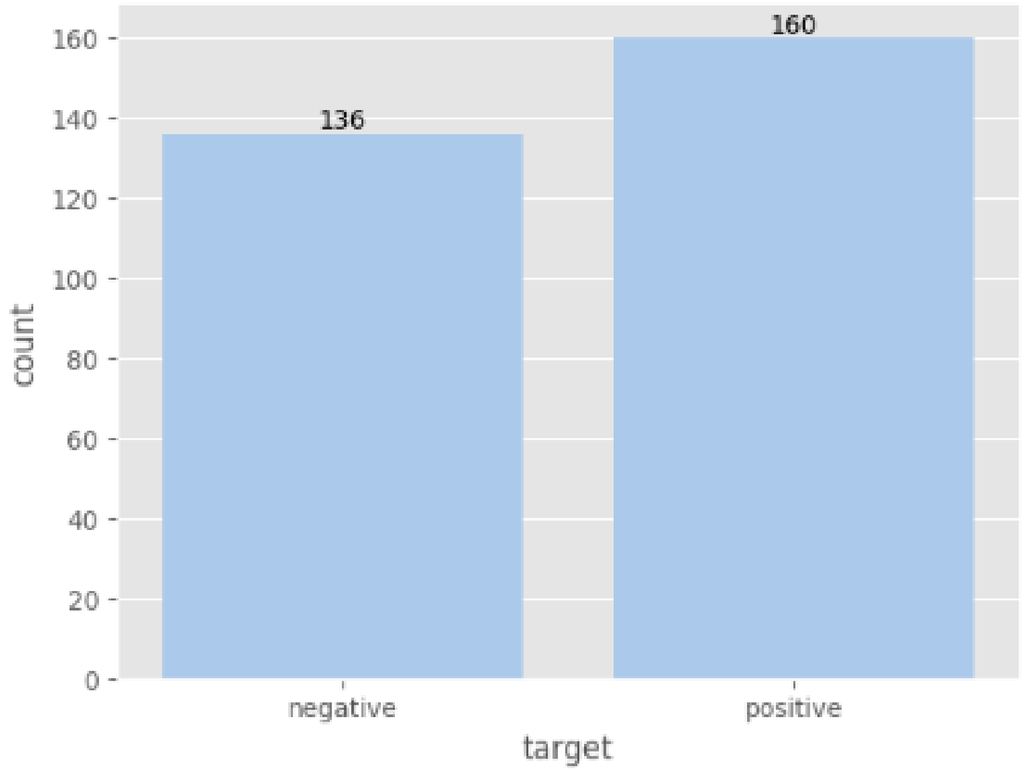








Imbalanced data check



target	proportion
positive	0.540541
negative	0.459459

The data little bit imbalanced, however still categorized as mildly imbalanced.

VALIDATING THE DATA (MEMVALIDASI DATA)

Handling Anomaly

There are some errors in the feature columns:

- Feature 'ca' should have a value between 0-3, but there are some data points with a value of 4.
- Feature 'thal' should have a value between 1-3, but there are some data points with value of 0.

```
# Displaying total unique value from 'ca' column
data['ca'].value_counts().to_frame()
```

ca	count
Number of major vessels: 0	578
Number of major vessels: 1	226
Number of major vessels: 2	134
Number of major vessels: 3	69
4	18

→

ca	count
Number of major vessels: 0	578
Number of major vessels: 1	226
Number of major vessels: 2	134
Number of major vessels: 3	69

```
# Displaying total unique value from 'thal' column
data['thal'].value_counts().to_frame()
```

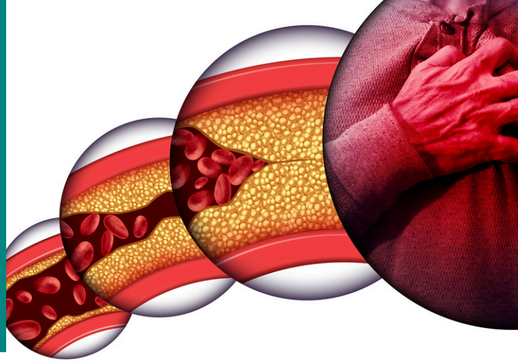
thal	count
fixed defect	544
reversable defect	410
normal	64
0	7

→

thal	count
fixed defect	544
reversable defect	410
normal	64

DETERMINE DATA OBJECT (MENENTUKAN OBJEK DATA)

Selecting object data



🔢 Choose features relevant to the problem:

- Selected features:
- age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal
- Target: num (converted to binary).
- Justification: Features are clinically relevant and frequently used in cardiology.

DATA CLEANING (MEMBERSIHKAN DATA)

Missing values

	Percentage of Missing values
age	0.000000
sex	0.000000
cp	0.000000
trestbps	0.000000
chol	0.000000
fbs	0.000000
restecg	0.000000
thalach	0.000000
exang	0.000000
oldpeak	0.000000
slope	0.000000
ca	1.756098
thal	0.682927
target	0.000000

```
# deleting rows with missing values
data = data.dropna()
```

Duplicated data

```
# Checking duplicated data
data.duplicated().sum()
```

```
np.int64(704)
```

```
# Dropping duplicated data
data.drop_duplicates(keep="first", inplace=True)

# Checking the number of duplicated data after treatment
data.duplicated().sum()
```

```
np.int64(0)
```

Data cleaning

Renaming/replacing labeling

```
# Categorical data labelling
data['sex'] = data['sex'].replace({1: 'Male',
                                   0: 'Female'})
data['cp'] = data['cp'].replace({0: 'typical angina',
                                 1: 'atypical angina',
                                 2: 'non-anginal pain',
                                 3: 'asymtomatic'})
data['fbs'] = data['fbs'].replace({0: 'No',
                                   1: 'Yes'})
data['restecg'] = data['restecg'].replace({0: 'probable/definite left ventricular hypertrophy',
                                           1:'normal',
                                           2: 'ST-T Wave abnormal'})
data['exang'] = data['exang'].replace({0: 'No',
                                       1: 'Yes'})
data['slope'] = data['slope'].replace({0: 'downsloping',
                                       1: 'flat',
                                       2: 'upsloping'})
data['thal'] = data['thal'].replace({1: 'normal',
                                     2: 'fixed defect',
                                     3: 'reversable defect'})
data['ca'] = data['ca'].replace({0: 'Number of major vessels: 0',
                                 1: 'Number of major vessels: 1',
                                 2: 'Number of major vessels: 2',
                                 3: 'Number of major vessels: 3'})
data['target'] = data['target'].replace({0: 'negative',
                                         1: 'positive'})
```

FEATURE ENGINEERING & TRANSFORMATION (MENGKONSTRUKSI DATA)

Feature engineering & Data Transformation

🔢 Choose features relevant to the problem:

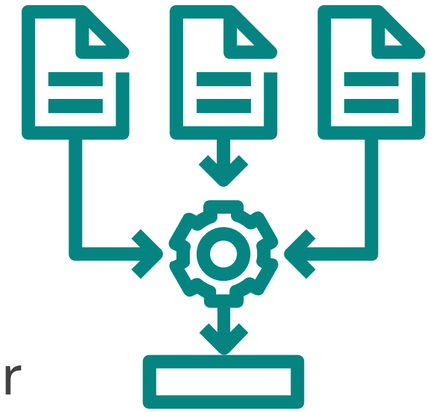
- Transform: num into binary.
- One-hot encode: sex, cp, fbs, restecg, exang, slope.
- Scaling: MinMaxScaler()

Transformer

```
# Column Transformer
transformer = ColumnTransformer([
    ('onehot', OneHotEncoder(drop='first'), ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope',
], remainder='passthrough')
```

MODEL SCENARIO BUILDING (MEMBANGUN SKENARIO MODEL)

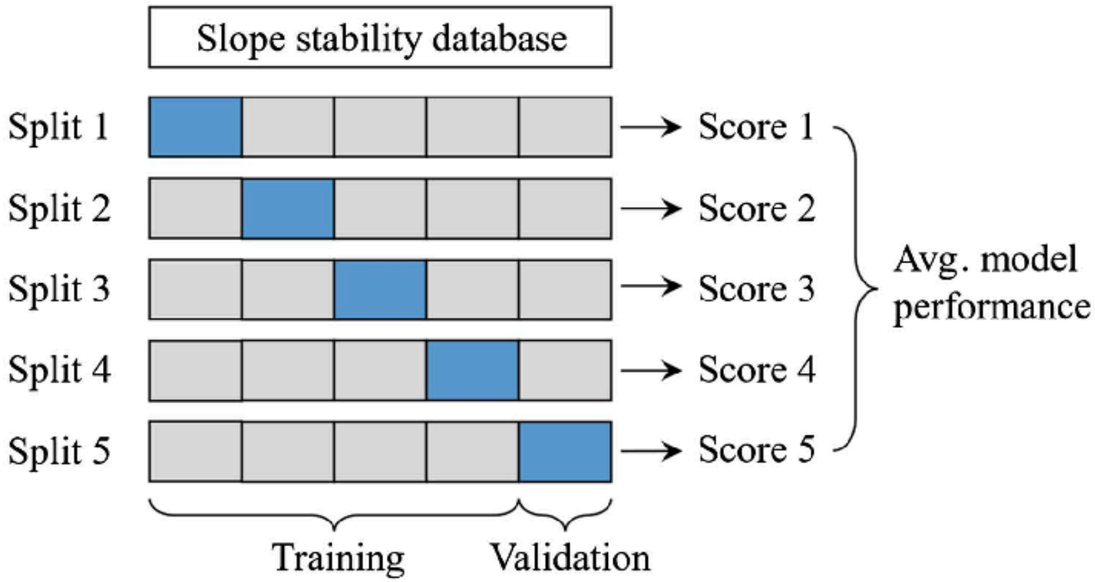
Benchmark Model



01 Logistic Regression
02 KNN Classifier
03 Decision Tree Classifier
04 Random Forest Classifier
05 Adaptive Booster Classifier
06 Gradient Booster Classifier
07 Categorical Booster Classifier
08 XGBoost Classifier
09 LGBM Classifier

Cross validation

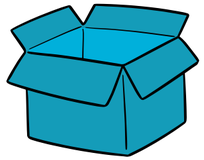
Stratified K-Fold (5 fold)



Preprocessing

Handling Outliers

Winsorization



Encoding

OneHot Encoder

Scaling

MinMaxScaler

Handling Imbalance

Resampling

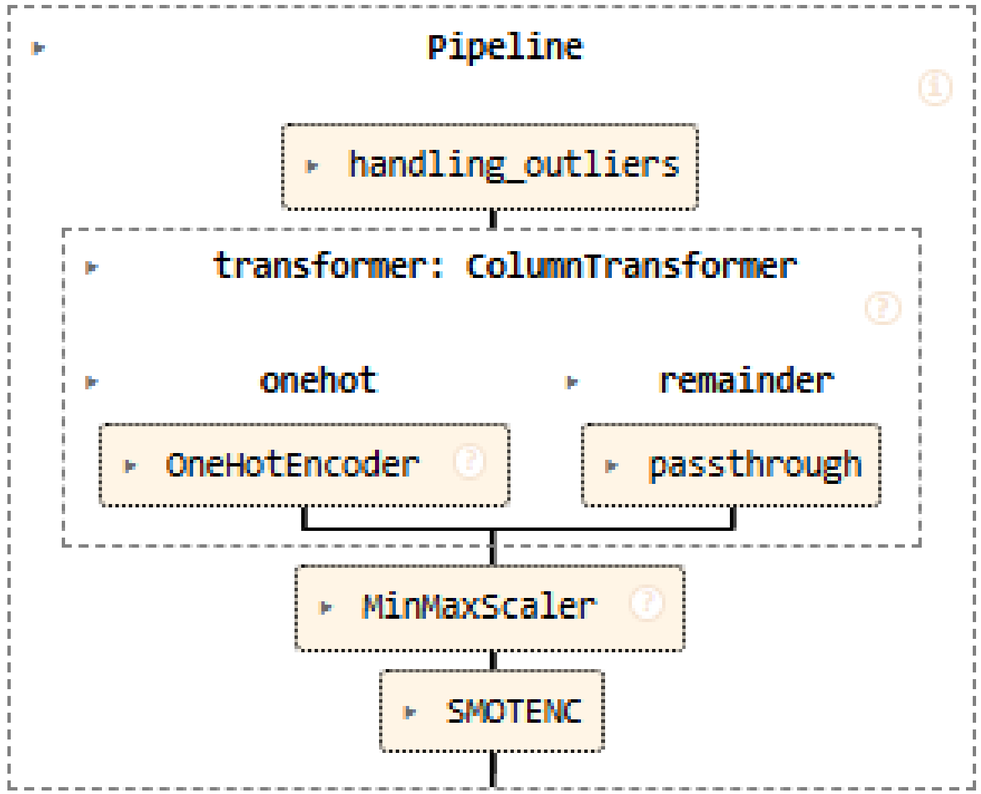
SMOTENC

MODEL BUILDING (MEMBANGUN MODEL)

Modeling

Pipeline

```
model_pipe = Pipeline([
    ('outlier', handling_outliers()),
    ('prep', transformer),
    ('scaler', scaler),
    ('resample', smotenc),
    ('algo', algoritma)
])
```



MODEL EVALUATION (MENGEVALUASI MODEL)

Model Performance

model	mean recall train	mean recall test
KNN	0.851385	0.84375
Logistic Regression	0.851692	0.81250
Random Forest	0.867385	0.81250
LightGBM	0.820308	0.75000
AdaBoost	0.813231	0.71875
GradienBoost	0.828308	0.71875
CatBoost	0.875385	0.71875
XGBoost	0.812308	0.71875
Decision Tree	0.726769	0.68750

KNN

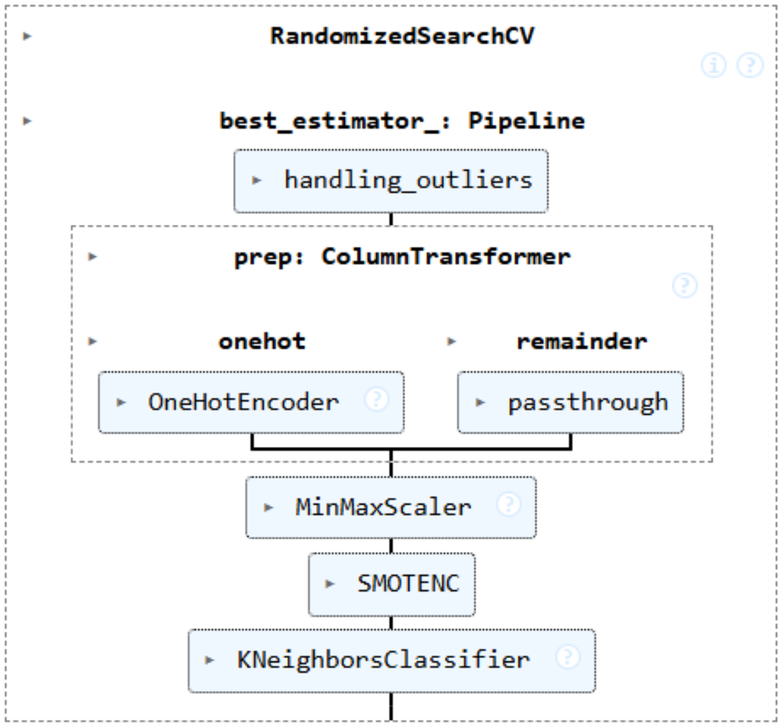
Logistic regression

Random Forest

Hyperparameter Tuning

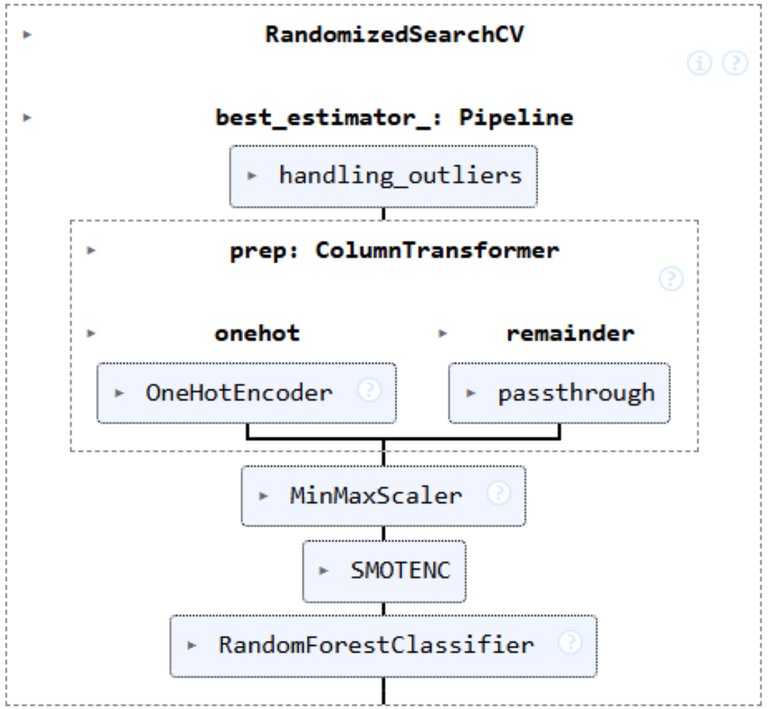
KNN

```
# Hyperparameter space of knn
hyperparam_space_knn_clf = [{
    'model__n_neighbors' : [3, 5, 7],
    'model__weights' : ['uniform', 'distance'],
    'model__metric' : ['euclidean', 'manhattan'],
}]
```



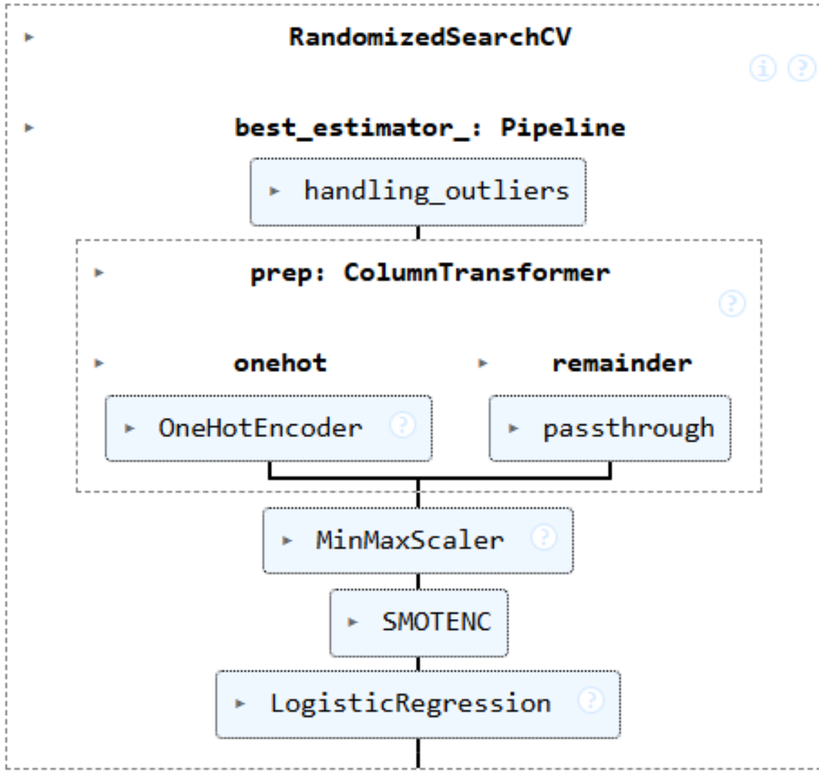
Random Forest

```
hyperparam_space_rf_clf = [{
    'model__n_estimators' : range(10,101,10),
    'model__max_features' : ['sqrt','log2',None],
    'model__max_depth' : range(10,101,10),
    'model__min_samples_split': range(2, 21, 2),
    'model__min_samples_leaf': range(1, 21, 2)
}]
```



Logistic Regression

```
hyperparam_space_logreg_clf = [{
    'model__max_iter' : [30, 50, 100, 150],
    'model__multi_class' : ['auto'],
    'model__solver': ['lbfgs', 'newton-cholesky']
}]
```



Hyperparameter Tuning

Model Performance (Recall score) Before & After Tuning

Model	Conditions	Train score	Test score
KNN Classifier	Before Tuning	0.851	0.844
KNN Classifier	After Tuning	0.891	0.875
Random Forest Classifier	Before Tuning	0.867	0.813
Random Forest Classifier	After Tuning	0.891	0.813
Logistic Regression Classifier	Before Tuning	0.851	0.813
Logistic Regression Classifier	After Tuning	0.851	0.813

Tuned KNN is the best model

Classification report

Random Forest

The Classification Report of Random Forest Classifier

	precision	recall	f1-score	support
0	0.79	0.79	0.79	28
1	0.81	0.81	0.81	32
accuracy			0.80	60
macro avg	0.80	0.80	0.80	60
weighted avg	0.80	0.80	0.80	60

K Nearest Neighbors

The Classification Report of K_Nearest Neighbors Classifier

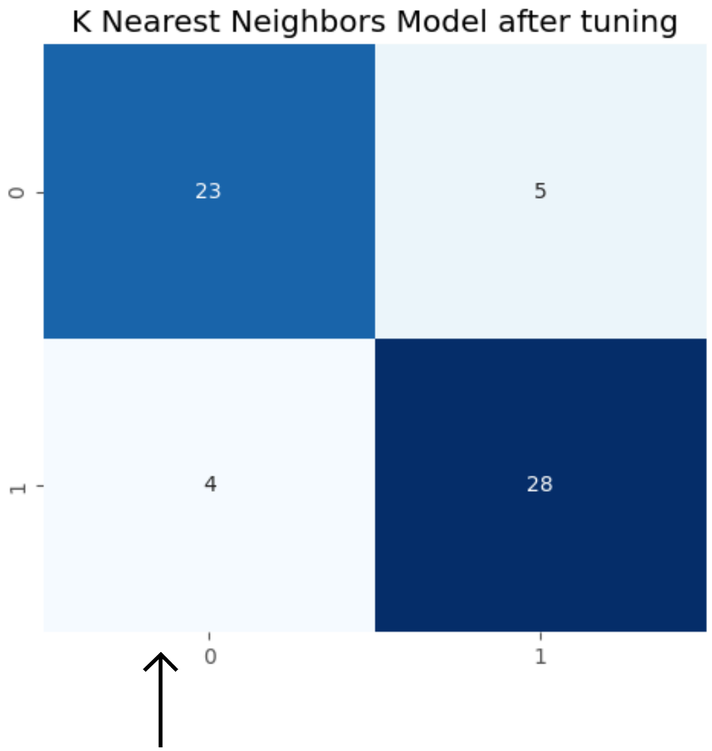
	precision	recall	f1-score	support
0	0.85	0.82	0.84	28
1	0.85	0.88	0.86	32
accuracy			0.85	60
macro avg	0.85	0.85	0.85	60
weighted avg	0.85	0.85	0.85	60

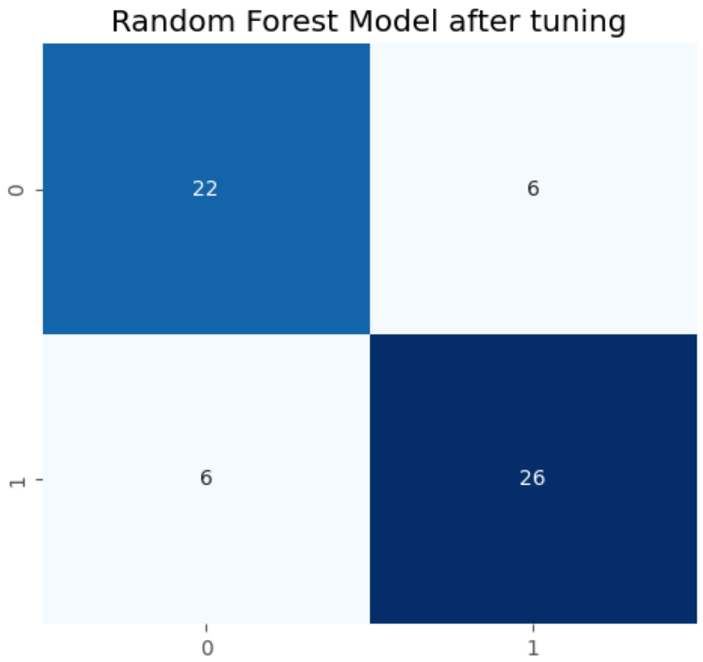
Logistic Regression

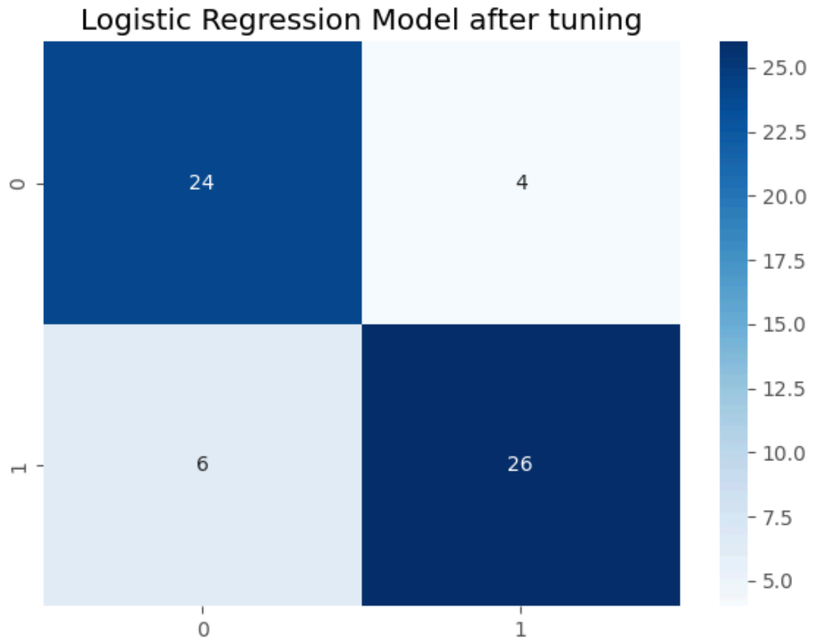
The Classification Report of Logistic Regression Classifier

	precision	recall	f1-score	support
0	0.80	0.86	0.83	28
1	0.87	0.81	0.84	32
accuracy			0.83	60
macro avg	0.83	0.83	0.83	60
weighted avg	0.84	0.83	0.83	60

Confusion matrix

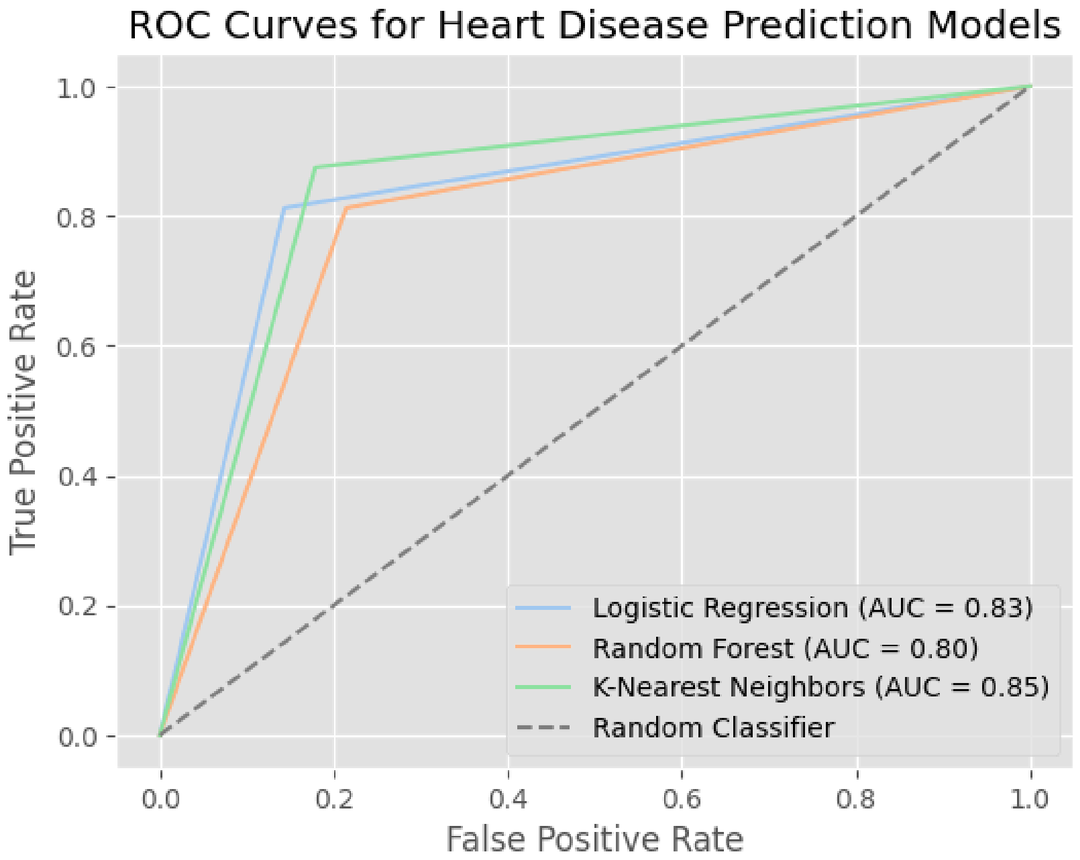






The Lowest False Negative

ROC curves

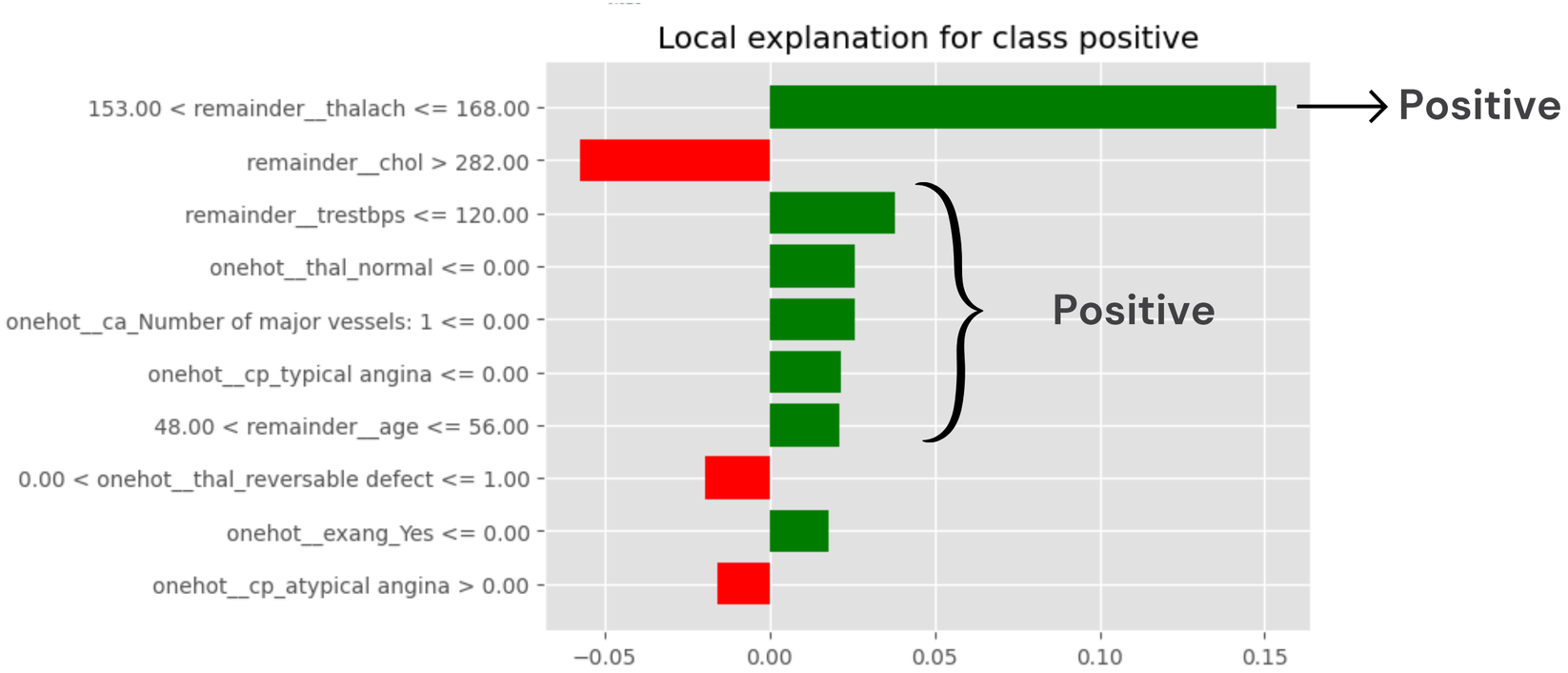


Tuned KNN is the best model

MODEL REVIEW (REVIEW PEMODELAN)

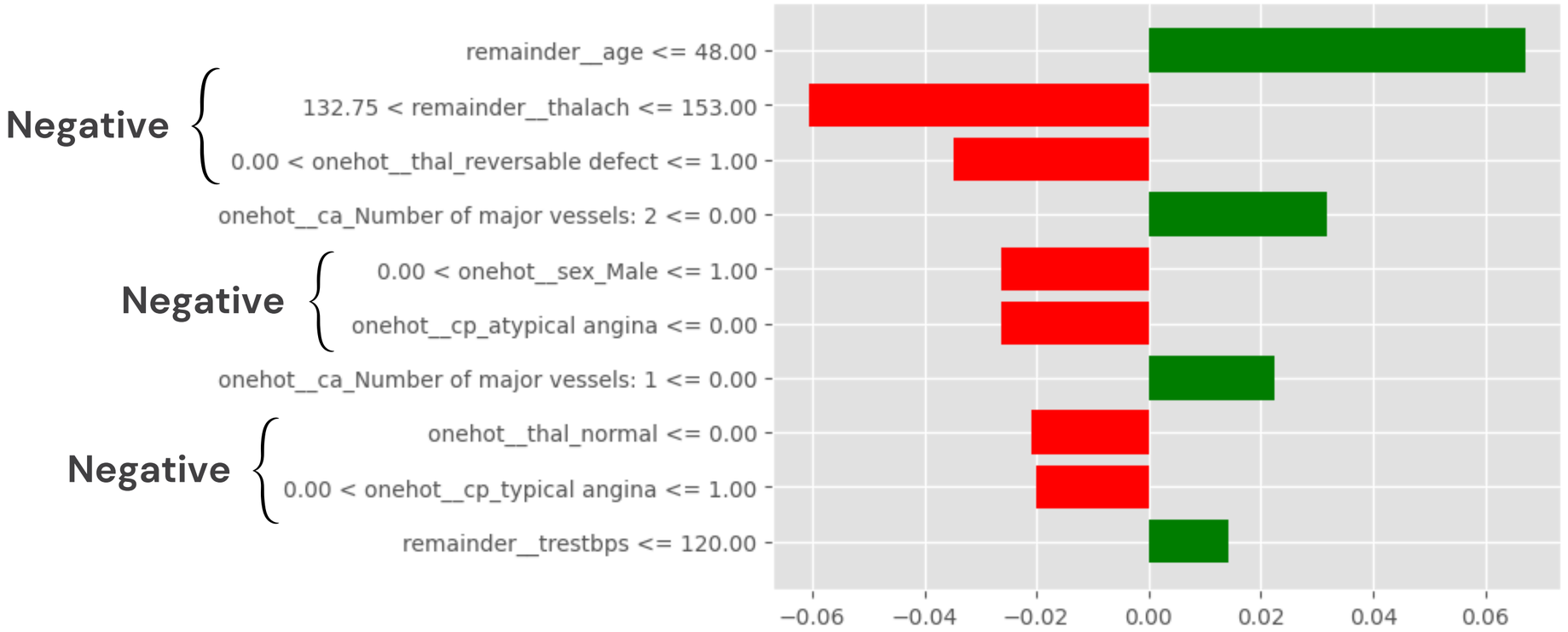
Explainable Model
LIME (Local Interpretable Model-agnostic)

Patient [10] – Positive Heart Diseases



Explainable Model
LIME (Local Interpretable Model-agnostic)

Patient [11] - Negative Heart Diseases



CONCLUSION

Tuned KNN is the best model for predicting heart disease in this scenario, due to its high recall of 0.891, F1-score 0.86 and best ROC AUC score of 0.85.

THANK YOU

My Contact

https://github.com/harishmuh

www.linkedin.com/in/harish-muhammad-7b600b102/

harishmuh@gmail.com

APP DEMO IN STREAMLIT

Upload your input CSV file

Drag and drop file here

Limit 200MB per file • CSV

Browse files

Manual Input

Chest pain type

2

1 4

Type of Chest pain: Atypical angina

Maximum heart rate achieved

80

71 202

Slope of the peak exercise ST segment

1

0 2

ST depression induced

1.00

0.00 6.20

Heart Disease Predictor App

This app predicts symptoms of heart disease

The dataset for this prediction was obtained from the Heart Disease dataset by UC Irvine ML repository .