HEART DISEASE CLASSIFICATION PROBLEM

Prediction with Machine Learning

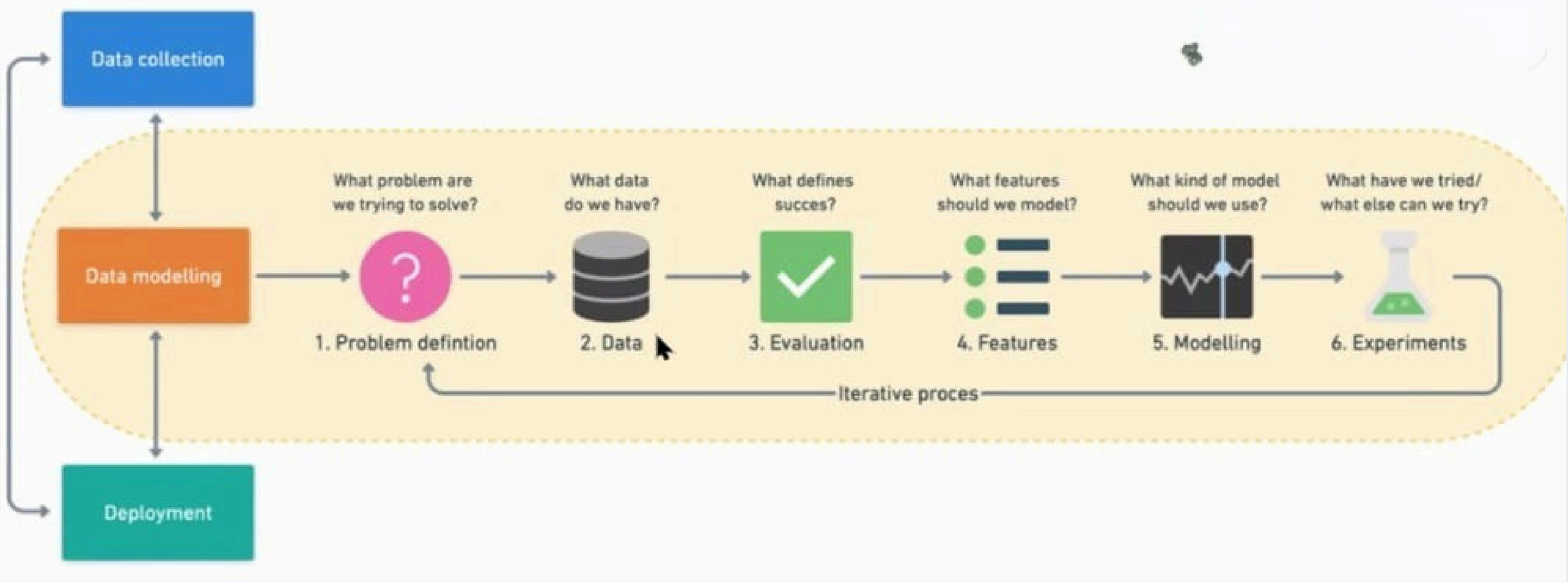
up to 97% Accuracy

by Nadhif Rif'at Rasendriya

DATASET

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

3 STEPS OF MACHINE LEARNING



Data Exploration (exploratory data analysis or EDA)

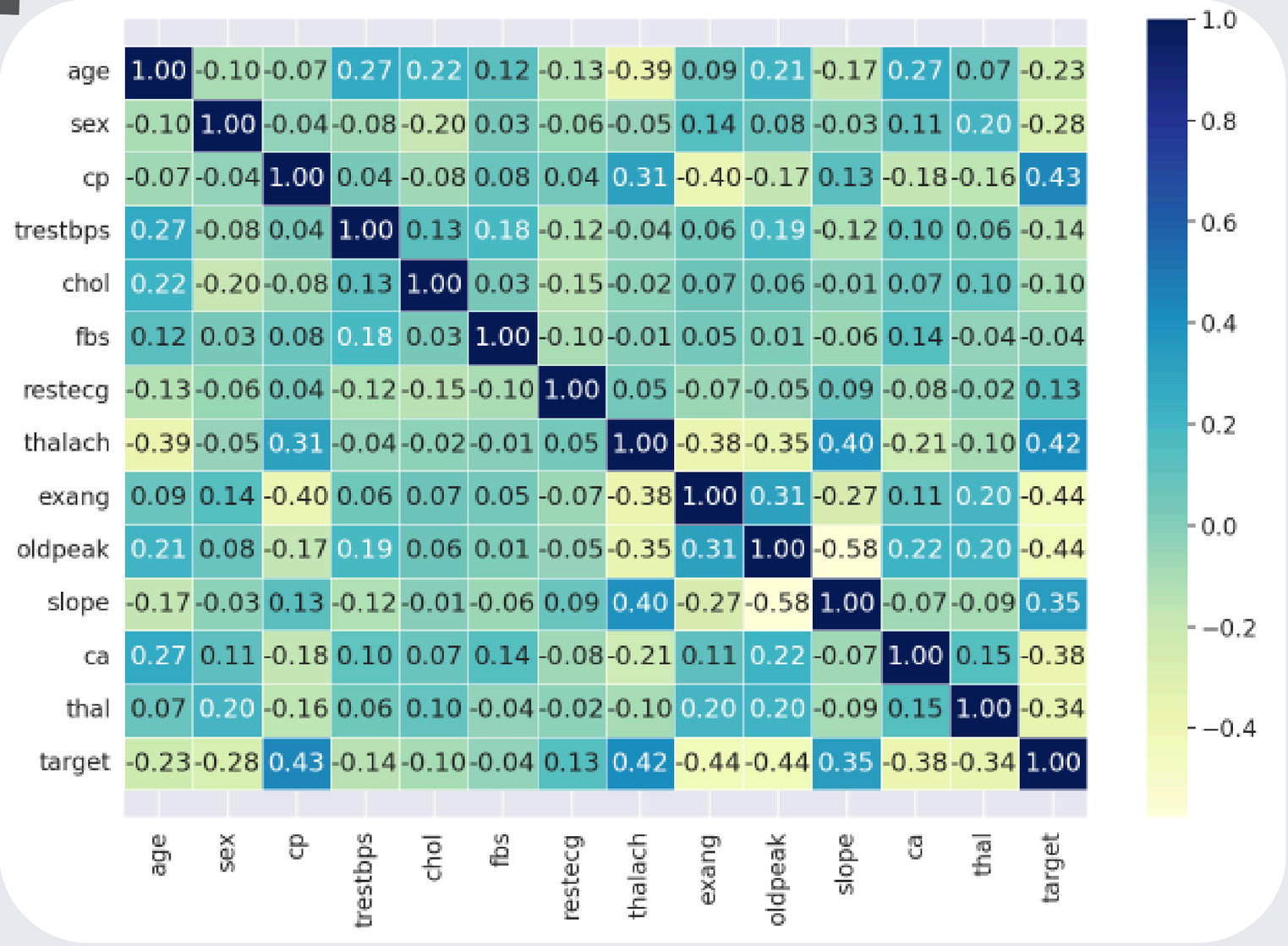
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

The goal here is to find out more about the data and become a subject matter export on the dataset you're working with.

1. What question(s) are you trying to solve?
2. What kind of data do we have and how do we treat different types?
3. What's missing from the data and how do you deal with it?
4. Where are the outliers and why should you care about them?
5. How can you add, change or remove features to get more out of your data?

Correlation Matrix



Yellow indicates the correlation between the 2 variables is low, whereas if the color is closer to blue it indicates the correlation is high

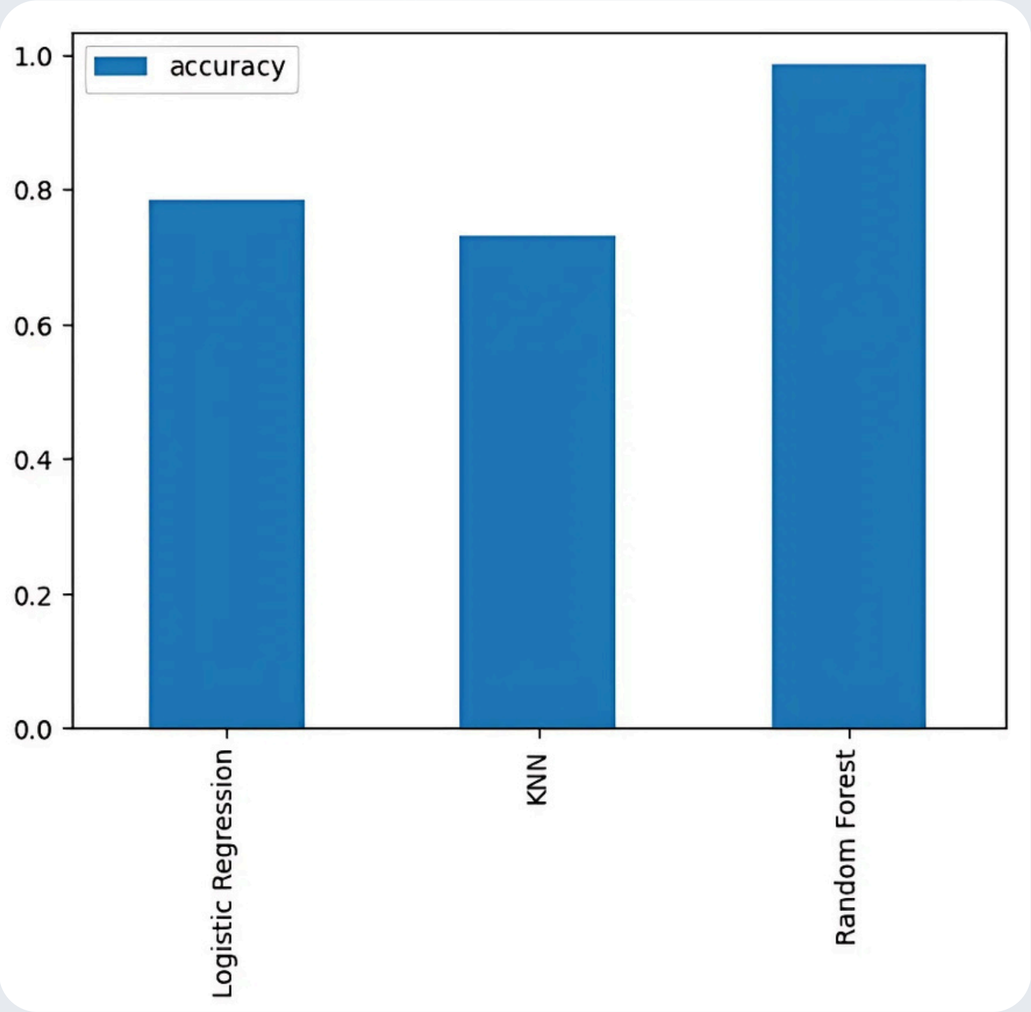
Here there are 3 features that have the most influence:
CP, thalach, and slope

MODELING

Logistic Regression, K-Nearest Neighbors, Random Forest

We'll train it (find the patterns) on the training set.
And we'll test it (use the patterns) on the test set.
We're going to try 3 different machine learning models.

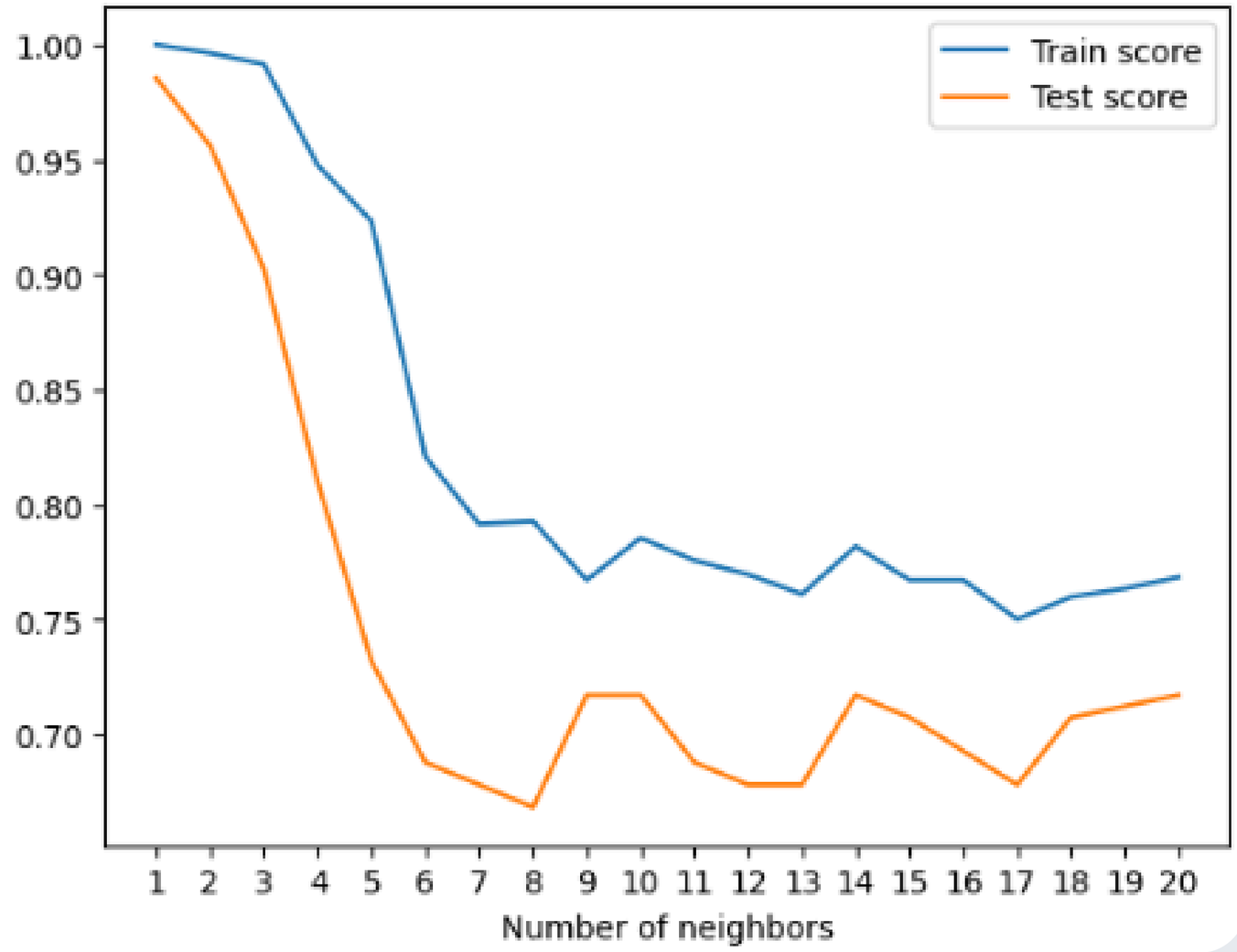
MODEL COMPARISON



Based on the model comparison results, Random Forest has the highest accuracy, which is **1.0 (100%)**. However, since this is just modeling, we still need additional evaluation. We should ask ourselves whether this 1.0 accuracy is truly that good. Therefore, we need to perform validation with **Hyperparameter Tuning**, as there is a possibility of **overfitting** in this model.

Hyperparameter Tuning - Plotting

```
Maximum KNNN score on the test data: 98.54%
```



When plotting using hyperparameter tuning, it was found that the **KNN** score reached a maximum value of **98.54%**, and the graph indicated the possibility of an accuracy of **1.0** (100%). Additionally, I also discovered that if the number of neighbors is below **5**, the model achieves **very high accuracy.**

In the KNN model, when we set n=1, it means the model only considers the nearest data point, making it **highly sensitive** to data variance.

To avoid overfitting, we will apply

Hyperparameter Tuning with RandomizedSearchCV

Note: Based on the Model Comparison in the initial section, we have found that the Random Forest model has the highest accuracy (1.0). Therefore, we will proceed with hyperparameter tuning using RandomizedSearchCV for this model.

Logistic Regression

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
RandomizedSearchCV
best_estimator_: LogisticRegression
LogisticRegression(C=1.623776739188721, solver='liblinear')
LogisticRegression
LogisticRegression(C=1.623776739188721, solver='liblinear')
```

```
[38] rs_log_reg.best_params_
```

```
{'solver': 'liblinear', 'C': 1.623776739188721}
```

```
[39] rs_log_reg.score(X_test, y_test)
```

```
0.7853658536585366
```

Random Forest

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
RandomizedSearchCV
best_estimator_: RandomForestClassifier
RandomForestClassifier(min_samples_split=14, n_estimators=510)
RandomForestClassifier
RandomForestClassifier(min_samples_split=14, n_estimators=510)
```

```
[41] # Find the best hyperparameters
     rs_rf.best_params_
```

```
{'n_estimators': 510,
 'min_samples_split': 14,
 'min_samples_leaf': 1,
 'max_depth': None}
```

```
[42] rs_rf.score(X_test, y_test) # score untuk test
```

```
0.926829268292683
```

Confusion Matrix

In classification problems, there are four types of predictions:

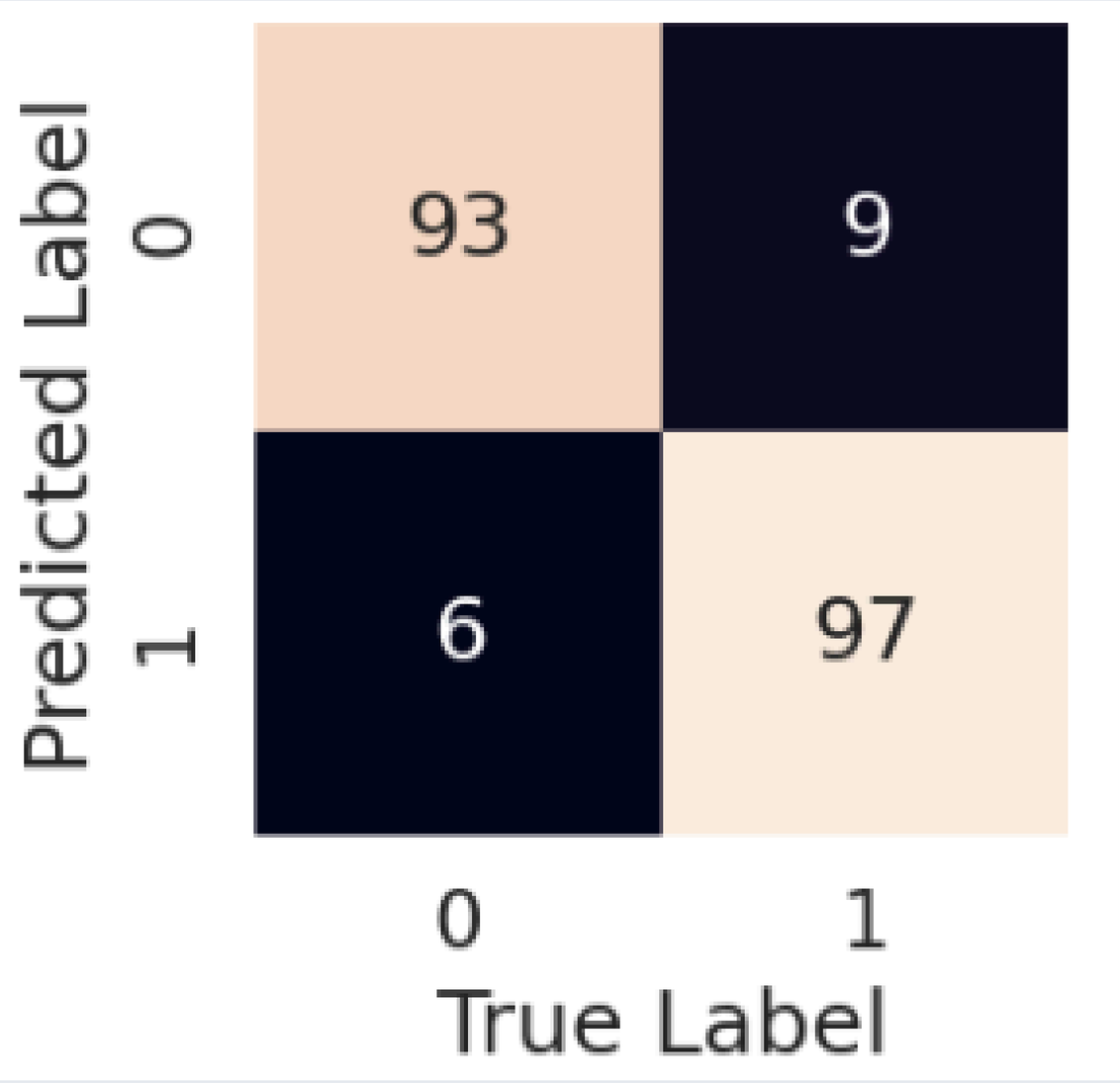
1. True Positive (TP) – The model correctly predicts a positive class when the actual class is also positive.
2. True Negative (TN) – The model correctly predicts a negative class when the actual class is negative.
3. False Positive (FP) – The model incorrectly predicts a positive class when the actual class is negative (also known as a Type I error).
4. False Negative (FN) – The model incorrectly predicts a negative class when the actual class is positive (also known as a Type II error).

1 -> P = T, A = T -> TP

2 -> P = T, A = F -> TN

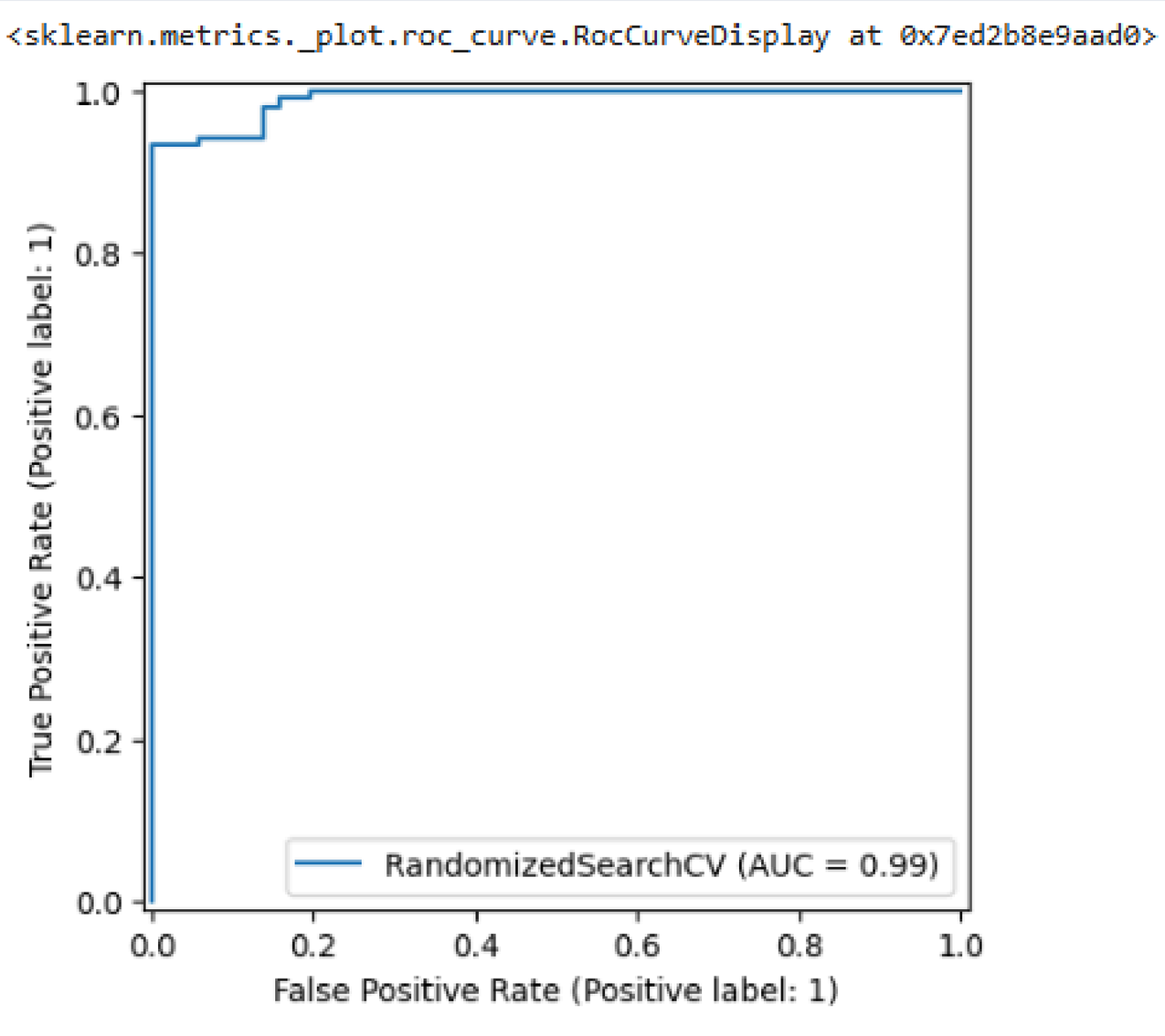
3 -> P = F, A = F -> FP

4 -> P = F, A = T -> FN



Confusion Matrix

So, in the confusion matrix visualization, there are **15** data (6+9) that were misclassified.



ROC Curve Display

Here, we can see that the AUC is **0.99**
When the AUC is above 0.90, it indicates that the model is good.
The area under the curve (error area) is small.

Conclusion

This project focuses on predicting heart disease using machine learning, achieving an accuracy of up to 97%. Among the tested models, Random Forest provided the highest accuracy of 100%, but further evaluation is needed to prevent overfitting.
Hyperparameter tuning results show that the K-Nearest Neighbors (KNN) model also performed well, with a maximum accuracy of 98.54%. The evaluation using the confusion matrix revealed 15 misclassified data points. Additionally, the AUC value of 0.99 indicates that the model performs exceptionally well in distinguishing between positive and negative classes.

Model Potential

Overall, the developed model has the potential to assist in the early detection of heart disease. However, further validation is required before it can be applied in real-world scenarios.

Project Link

Github

https://github.com/nadhif-royal/HeartDiseasePredictionML

Dataset

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

Thankyou

Nadhif Rif’at Rasendriya

linkedin.com/in/royalnadhif50/

instagram.com/royal_nadhif/