Heart Disease Prediction

OUR TEAM

Fatmanur Karabacak

Zeynep Vural

Muhteber Aydın

PURPOSE OF PROJECT

This project focuses on developing a model using machine learning techniques to diagnose heart disease. The dataset used in this project contains various clinical characteristics related to heart disease, including age, gender, symptoms, blood pressure, cholesterol levels and other clinical measurements. Our goal is to build a model to accurately classify whether a person has heart disease using machine learning algorithms on this dataset.

DATASET

There is a total of 920 patient data.

Each patient has 14 features.

It includes data from patients in 4 different regions: Hungarian, Zurich, Cleveland, Long Beach.

THE FEATURES IN DATASET

01 **AGE :** Age

02 **SEX :** Sex

03 **CP:** Type of chest pain.

04 **TRESTBPS:** The patient's blood pressure at rest.

05 **CHOL:** The patient's cholesterol level.

06 **FBS:** The patient's fasting blood glucose level.

07 **RESTECG:** Refers to electrocardiographic results at rest.

08 **THALACH:** Maximum heart rate refers to the highest heart rate an individual can achieve during exercise or under stress.

09 **EXANG:** The presence of exercise-related angina.

10 **OLDPEAK:** Refers to exercise-induced ST segment depression .

11 **SLOPE:** The slope of the ST segment after exercise.

12 **CA:** The number of major vessels colored by fluoroscopy.

13 **THAL:** This characteristic is an important indicator in determining the type and severity of heart disease.

14 **NUM** (Heart Disease Diagnosis): Refers to the patient's diagnosis of heart disease.

PREPROCESSING

```
age           0
sex           0
cp            0
trestbps     59
chol         30
fbs          90
resteccg      2
thalach      55
exange       55
oldpeak      62
slope       309
ca          611
thall       486
num           0
dtype: int64
```

Removing missing values

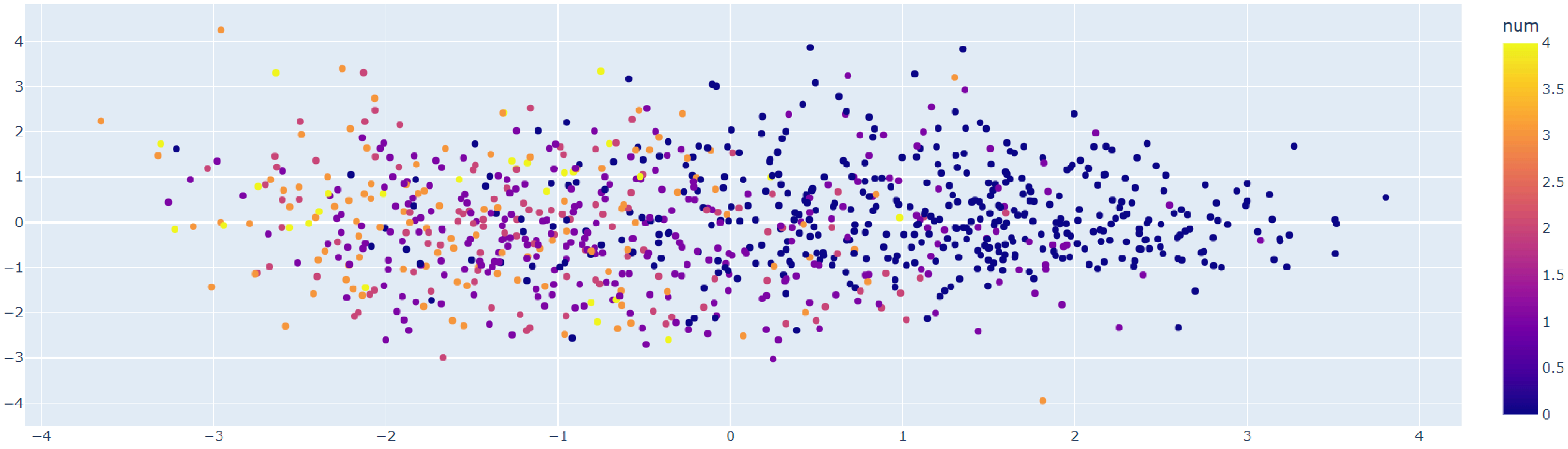
We removed classes with too many NaN values.

Filling empty values

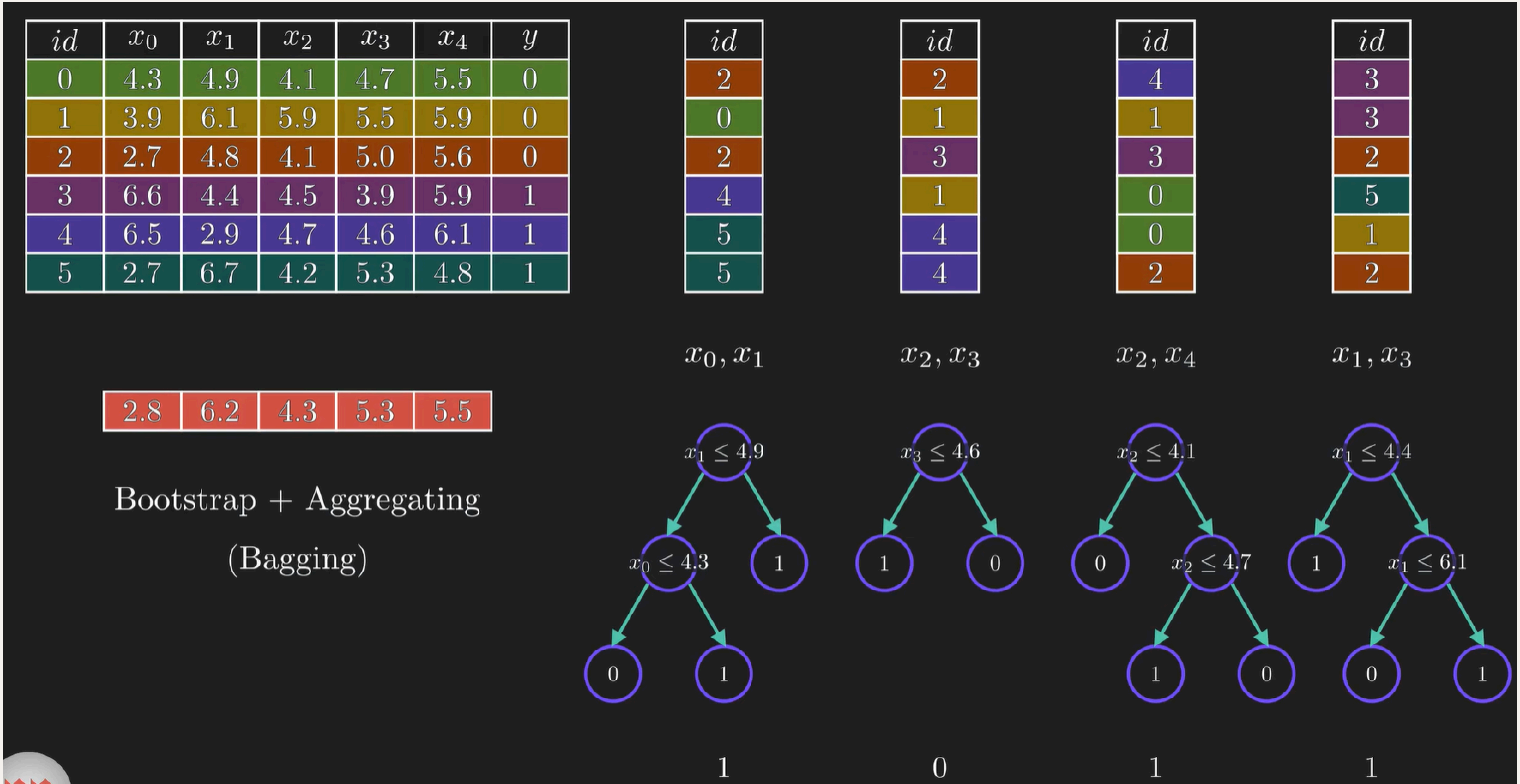
Using the SimpleImputer class, missing values in the dataset are filled with average values. This step is important for cleaning the dataset and making it suitable for machine learning models.

VISUALIZATION

PCA (Principal Component Analysis) method is used to reduce multidimensional data into two dimensions and to visualize this reduced data.



RANDOM FOREST CLASSİFİER



RANDOM FOREST CLASSİFİER

```
RandomForestClassifier Sınıflandırma Raporu:
              precision    recall  f1-score   support

           0       0.72      0.87      0.78        84
           1       0.55      0.53      0.54        59
           2       0.50      0.24      0.32        21
           3       0.23      0.18      0.20        17
           4       0.00      0.00      0.00         3

    accuracy                           0.61       184
   macro avg       0.40      0.36      0.37       184
weighted avg       0.58      0.61      0.59       184
```

IT WAS LİKELY DUE TO THE UNBALANCED CLASS DİSTRİBUTİON PROBLEM

THE NUMBER OF EXAMPLES İN THE CLASSES İS;

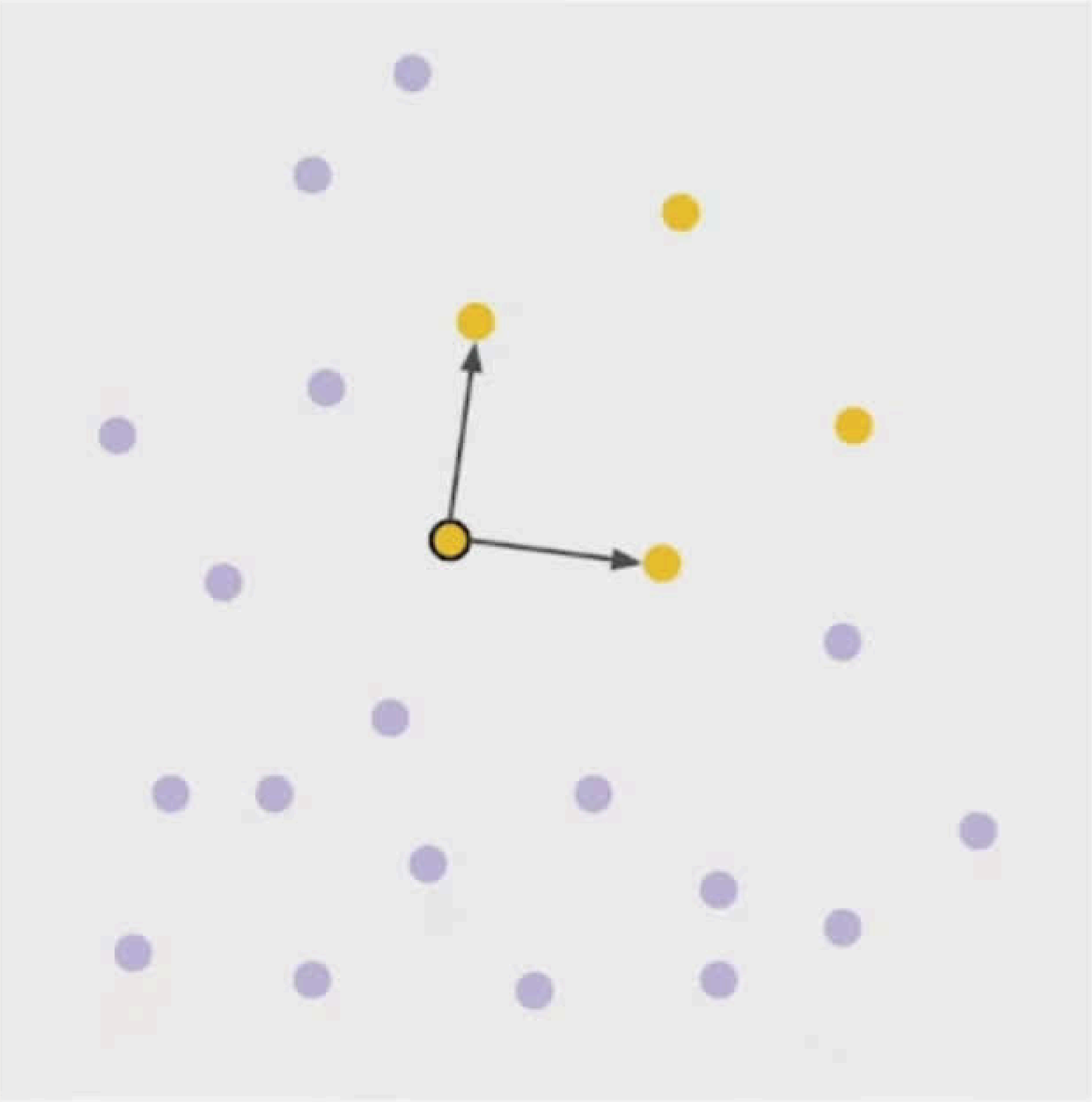
num	
0	411
1	265
2	109
3	107
4	28

BEFORE

num	
0	411
2	411
1	411
3	411
4	411

AFTER

SMOTE (Synthetic Minority Oversampling Technique)



The SMOTE Algorithm

FOR OVERSAMPLING THE MINORITY CLASS

1. Identify a data point from the minority class.
2. Find its k nearest neighbor. Here, k is 2.

RANDOM FOREST CLASSİFİER AFTER SMOTE

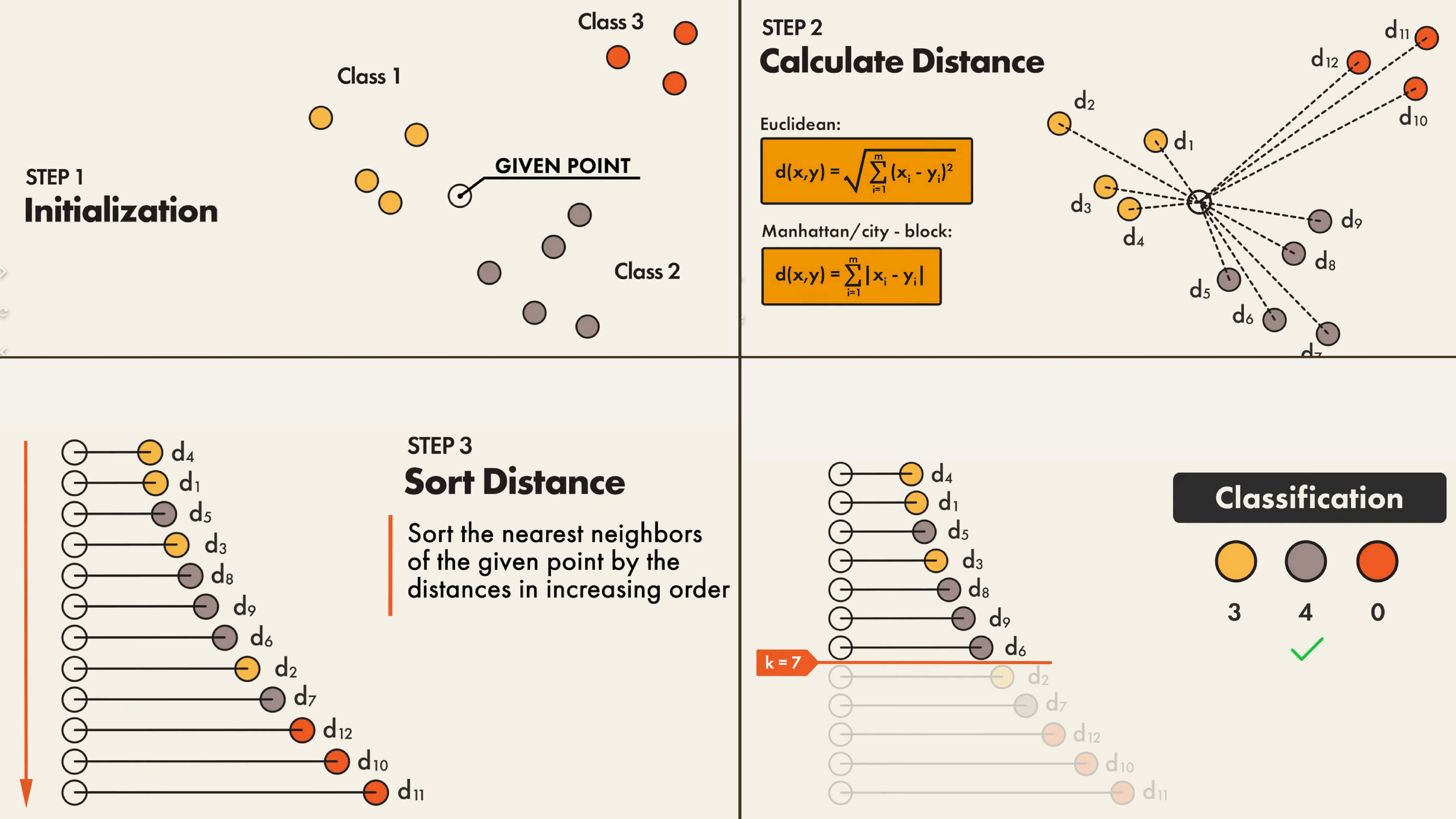
SMOTE işleminden sonra RandomForestClassifier Sınıflandırma Raporu:

	precision	recall	f1-score	support
0	0.82	0.73	0.78	83
1	0.65	0.65	0.65	82
2	0.82	0.80	0.81	82
3	0.79	0.82	0.80	82
4	0.90	1.00	0.95	82
accuracy			0.80	411
macro avg	0.80	0.80	0.80	411
weighted avg	0.80	0.80	0.80	411

AFTER APPLYİNG HYPERPARAMETER TUNİNG OPERATİON TO RFC

```
En iyi Random Forest Classifier Sınıflandırma Raporu:
              precision    recall  f1-score   support

           0       0.82      0.73      0.78        83
           1       0.68      0.66      0.67        82
           2       0.80      0.80      0.80        82
           3       0.78      0.83      0.80        82
           4       0.92      1.00      0.96        82

    accuracy                           0.81       411
   macro avg       0.80      0.81      0.80       411
weighted avg       0.80      0.81      0.80       411
```

```
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['auto', 'sqrt', 'log2']
}
```



STEP 1
Initialization
Class 1
Class 2
Class 3
GIVEN POINT
STEP 2
Calculate Distance
Euclidean:
$d(x,y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$
Manhattan/city - block:
$d(x,y) = \sum_{i=1}^{m} |x_i - y_i|$
d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9 d_{10} d_{11} d_{12}
STEP 3
Sort Distance
Sort the nearest neighbors of the given point by the distances in increasing order
d_4 d_1 d_5 d_3 d_8 d_9 d_6 d_2 d_7 d_{12} d_{10} d_{11}
k = 7
Classification
3
4
0

K-NEİGHBORS CLASSİFİER

KNeighborsClassifier Sınıflandırma Raporu:

	precision	recall	f1-score	support
0	0.78	0.64	0.70	83
1	0.62	0.41	0.50	82
2	0.73	0.88	0.80	82
3	0.73	0.79	0.76	82
4	0.82	1.00	0.90	82
accuracy			0.74	411
macro avg	0.74	0.74	0.73	411
weighted avg	0.74	0.74	0.73	411

After Applying Hyperparameter Tuning Operation to KNN

```
En iyi parametrelerle KNeighbors Classifier Sınıflandırma Raporu:
              precision    recall  f1-score   support

           0       0.83      0.65      0.73        83
           1       0.75      0.68      0.71        82
           2       0.85      0.94      0.89        82
           3       0.84      0.93      0.88        82
           4       0.91      1.00      0.95        82

    accuracy                           0.84       411
   macro avg       0.84      0.84      0.83       411
weighted avg       0.84      0.84      0.83       411
```

```
param_grid = {
    'n_neighbors': [3, 5, 7, 9],  # Komşu sayısı için aralık
    'weights': ['uniform', 'distance'],  # Ağırlık fonksiyonu için seçenekler
    'metric': ['euclidean', 'manhattan']  # Uzaklık metriği seçenekleri
}
```

SUPPORT VECTOR MACHINE

Purpose: is to find a hyperplane that separates data points between classes as clearly as possible.

SVC Sınıflandırma Raporu:

	precision	recall	f1-score	support
0	0.72	0.72	0.72	75
1	0.49	0.39	0.43	85
2	0.60	0.75	0.67	81
3	0.63	0.47	0.54	80
4	0.82	0.97	0.89	90
accuracy			0.66	411
macro avg	0.65	0.66	0.65	411
weighted avg	0.65	0.66	0.65	411

We used linear kernel

SUPPORT VECTOR MACHINE

```
Polinomiyal SVM Sınıflandırma Raporu:
              precision    recall  f1-score   support

           0       0.77      0.64      0.70        83
           1       0.64      0.56      0.60        82
           2       0.77      0.78      0.78        82
           3       0.73      0.88      0.80        82
           4       0.93      1.00      0.96        82

    accuracy                           0.77       411
   macro avg       0.77      0.77      0.77       411
weighted avg       0.77      0.77      0.77       411
```

```
# Polinomiyal çekirdek kullanarak SVM modelini oluşturma
svm_model_poly = SVC(kernel='poly', degree=5, coef0=1)
svm_model_poly.fit(x_train, y_train)
```

As a result of the parameter setting process with GridSearchCV, we observed that SVM with polynomial kernel achieved a higher accuracy rate.

DESICION TREE CLASSIFIER

1. Split Root Node

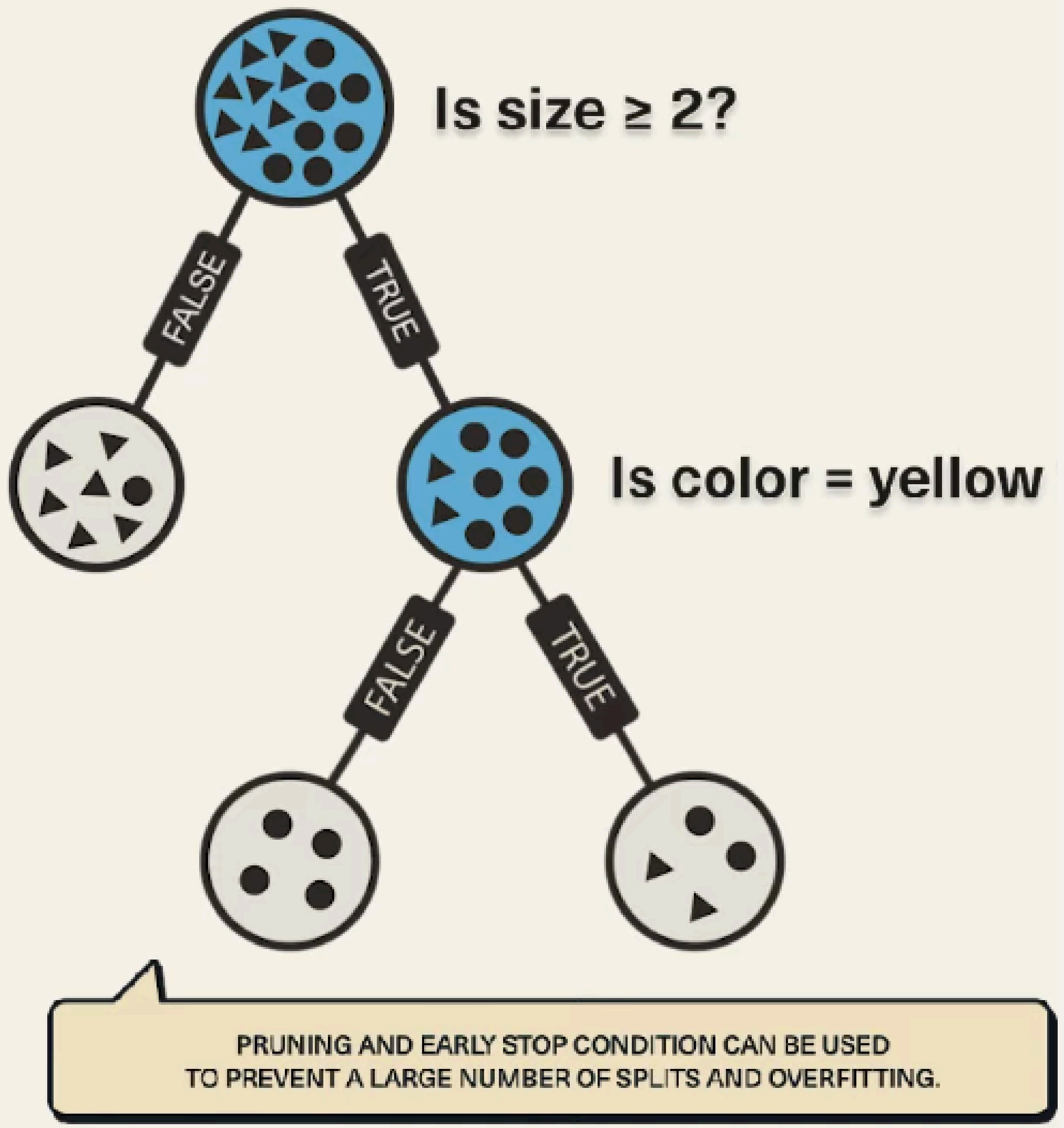
Select the decision rule which has the largest information gain to split the current node.

2. Split Child Node

Only consider decision rule never selected before in the current branch.

3. Stop Splitting

Stop splitting if there are no decision rules left or the group impurity = 0.



DESICION TREE CLASSIFIER

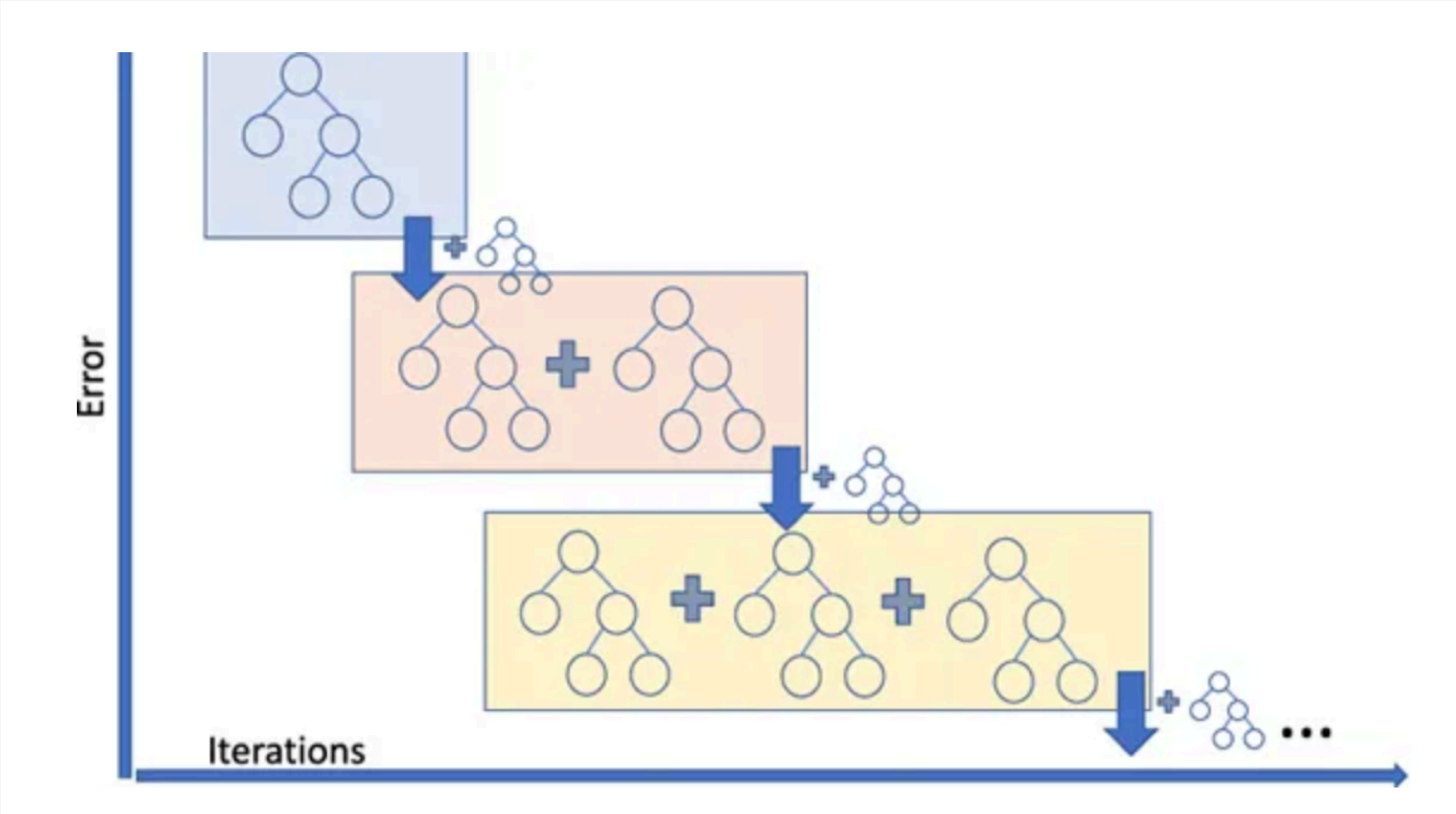
DecisionTreeClassifier Sınıflandırma Raporu:

	precision	recall	f1-score	support
0	0.66	0.67	0.66	75
1	0.57	0.58	0.57	85
2	0.69	0.74	0.71	81
3	0.61	0.51	0.56	80
4	0.85	0.90	0.88	90
accuracy			0.68	411
macro avg	0.68	0.68	0.68	411
weighted avg	0.68	0.68	0.68	411

```
decision_tree_model = DecisionTreeClassifier(class_weight='balanced')
```

GRADIENT BOOSTING CLASSIFIER

Purpose: This is because each successive estimator focuses on correcting the errors of previous estimators..



GRADIENT BOOSTING CLASSIFIER

GradientBoostingClassifier Sınıflandırma Raporu:

	precision	recall	f1-score	support
0	0.75	0.72	0.73	75
1	0.47	0.45	0.46	85
2	0.70	0.74	0.72	81
3	0.63	0.57	0.60	80
4	0.80	0.89	0.84	90
accuracy			0.68	411
macro avg	0.67	0.67	0.67	411
weighted avg	0.67	0.68	0.67	411

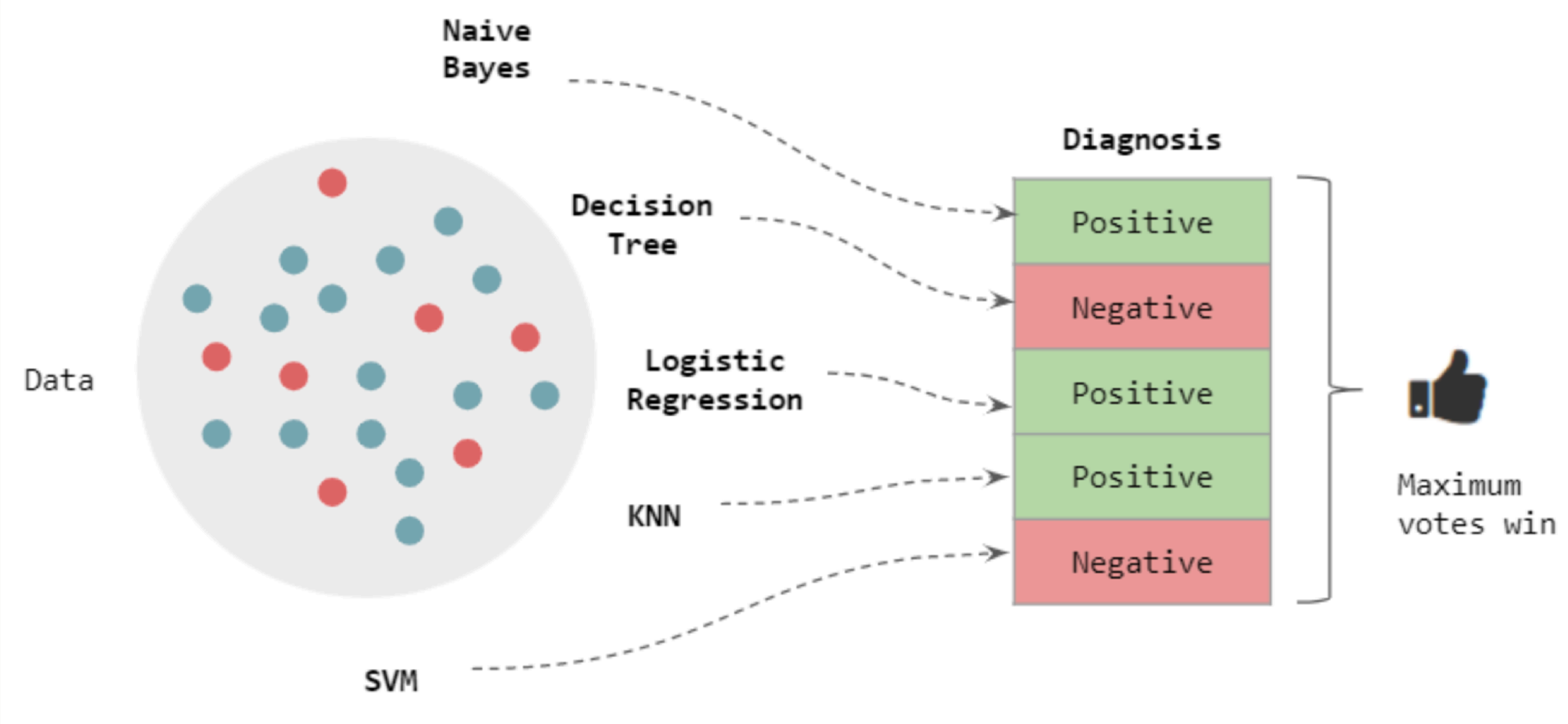
GRADIENT BOOSTING CLASSIFIER

```
En iyi parametreler: {'learning_rate': 0.5, 'max_depth': 7, 'n_estimators': 200}
En iyi skor: 0.7548661800486617
GradientBoostingClassifier Sınıflandırma Raporu:
              precision    recall  f1-score   support

           0       0.75      0.71      0.73        83
           1       0.62      0.59      0.60        82
           2       0.75      0.76      0.75        82
           3       0.76      0.77      0.76        82
           4       0.91      0.98      0.94        82

    accuracy                           0.76       411
   macro avg       0.76      0.76      0.76       411
weighted avg       0.76      0.76      0.76       411
```

```python
# GridSearchCV için parametrelerin belirlenmesi
param_grid = {
    'n_estimators': [50, 100, 200],
    'learning_rate': [0.01, 0.1, 0.5],
    'max_depth': [3, 5, 7]
}
```

ENSEMBLE METHODS

Purpose: Creating a stronger model by combining multiple classifier models.



ENSEMBLE METHODS

Classifiers:

1. **Random Forest Classifier:**
   - Parametreler: n_estimators=100, random_state=42
2. **Extra Trees Classifier:**
   - Parametreler: n_estimators=100, random_state=42
3. **K-Nearest Neighbors Classifier:**
   - Parametreler: n_neighbors=3, weights='distance', metric='manhattan'
4. **Support Vector Classifier:**
   - Parametreler: kernel='linear', probability=True

ENSEMBLE METHODS

```
ensemble_model = VotingClassifier(estimators=[
    ('rfc', rfc),
    ('et', et),
    ('knn', knn),
    ('svc', svc)
], voting='hard')
```

Creating a Model with Voting:

Classifiers are combined using the VotingClassifier class.

Classifiers were combined with hard voting method

```
Ensemble Model Doğruluk Skoru: 0.8321167883211679
```

ENSEMBLE METHODS

```
ensemble_model = VotingClassifier(estimators=[
    ('rfc', rfc),
    ('et', et),
    ('knn', knn),
    ('svc', svc)
], voting='soft')
```

Creating a Model with voting:

- Classifiers are combined with the soft voting method..

```
Ensemble Model Doğruluk Skoru: 0.8564476885644768
```

AUTOML

It is an approach that automates the process of developing machine learning models. This approach makes it faster and easier for users to build machine learning models by automating steps such as data preprocessing, feature selection, model selection, hyperparameter tuning, and evaluation of results.

AutoGluon Sınıflandırma Raporu:

	precision	recall	f1-score	support
0	0.70	0.89	0.79	84
1	0.55	0.46	0.50	59
2	0.31	0.19	0.24	21
3	0.29	0.24	0.26	17
4	1.00	0.33	0.50	3
accuracy			0.60	184
macro avg	0.57	0.42	0.46	184
weighted avg	0.57	0.60	0.58	184

THANK YOU

HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRİVE/1OLJCRQ74NS8J5İOYEQFF
TVNUGOX6JVBJ?USP=SHARİNG