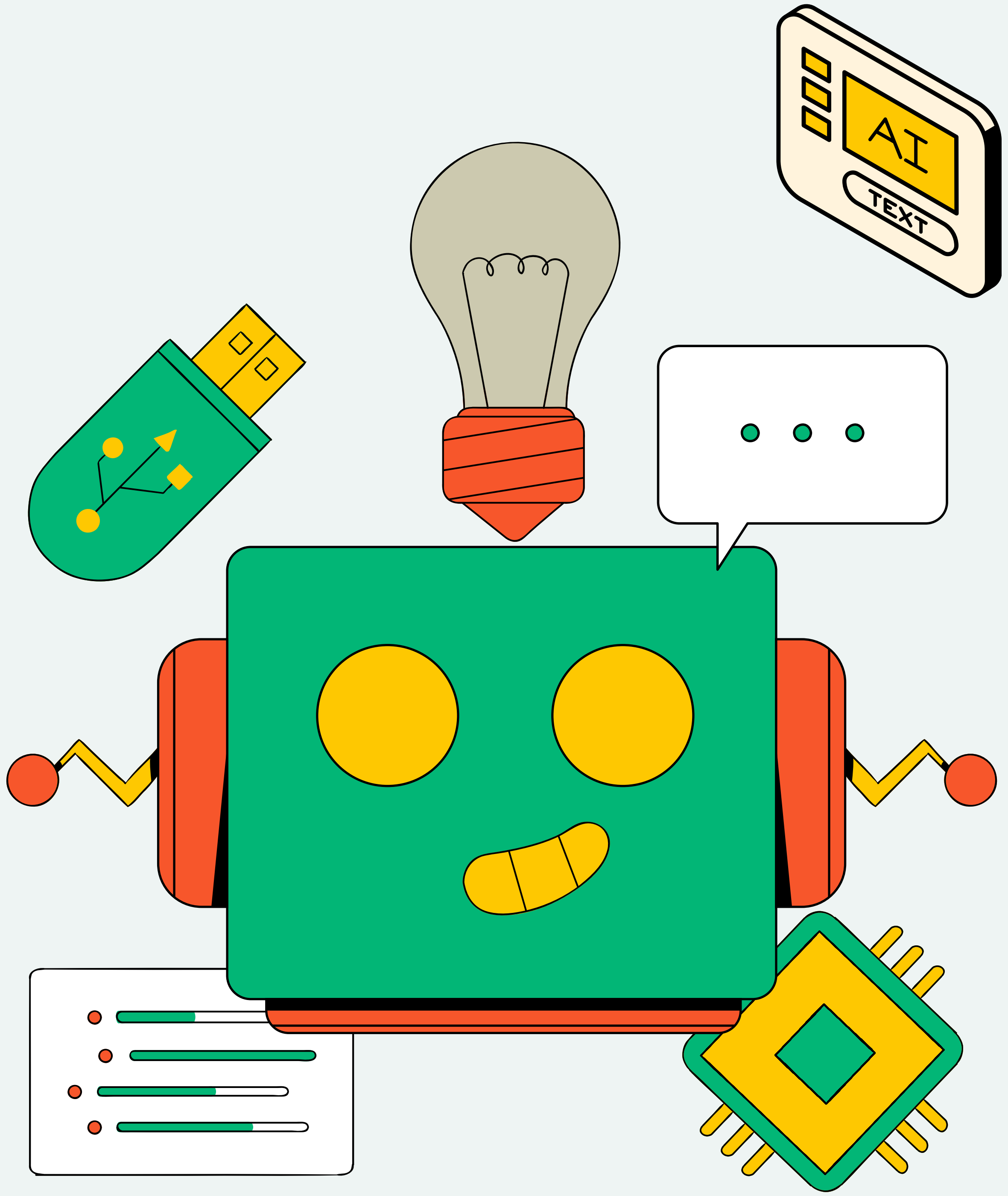
THYNK UNLIMITED

WE LEARN FOR THE FUTURE

ADVANCED MACHINE LEARNING

PRESENTATION



PRESENTED BY

1-MARTINA MOUSA

2-SALMA ANWER

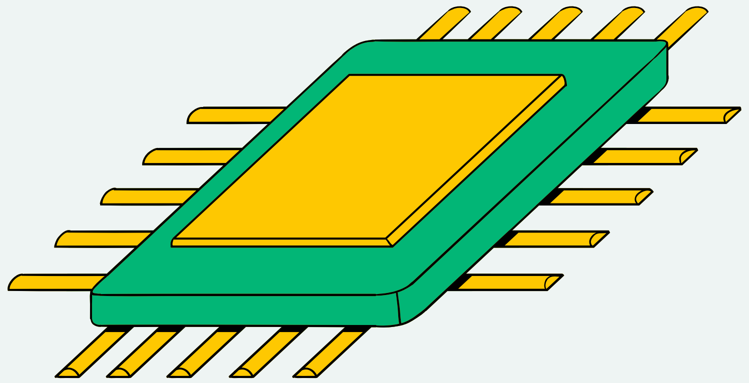
3-BASEL WAEL

4-SAIF MAHMOUD

5-OMAR AHMED

6-ZAYED TAMER

7-TAHA MOHAMED



PRESENTATION OUTLINE

- INTRODUCTION TO MOBILE PRICE CLASSIFICATION DATASET
- INFORMATION ABOUT MOBILE PRICE CLASSIFICTION DATASET
- ANN MODEL INFORMATION
- ANN MODEL RESULTS
- DECISION TREE MODEL INFORMATION
- DECISION TREE MODEL RESULTS
- INTRODUCTION TO INSURANCE DATASET
- INFORMATION ABOUT INSURANCE DATASET
- SVM MODEL INFORMATION
- SVM MODEL RESULTS

INTRODUCTION TO MOBILE PRICE CLASSIFICATION DATASET

The Mobile Price Classification dataset is a collection of attributes associated with mobile phones, ranging from technical specifications to features that influence pricing. It serves as a valuable resource for machine learning tasks, particularly classification, where the goal is to predict the price range of a mobile device based on its attributes

- The dataset encompasses various features such as battery power, Bluetooth, clock speed, dual SIM support, front and primary camera specifications, 4G support, internal memory, screen resolution, RAM, and other hardware specifications.
- Each mobile phone entry is labeled with one of four price ranges: low, medium, high, or very high, enabling supervised learning algorithms to train and predict accordingly.

OBJECTIVE

The primary objective of this dataset is to develop predictive models capable of accurately determining the price category of a mobile phone based solely on its technical attributes. This task is crucial for both consumers and manufacturers alike, offering insights into market segmentation and consumer preferences.

INFORMATION ABOUT MOBILE PRICE CLASSIFICTION DATASET

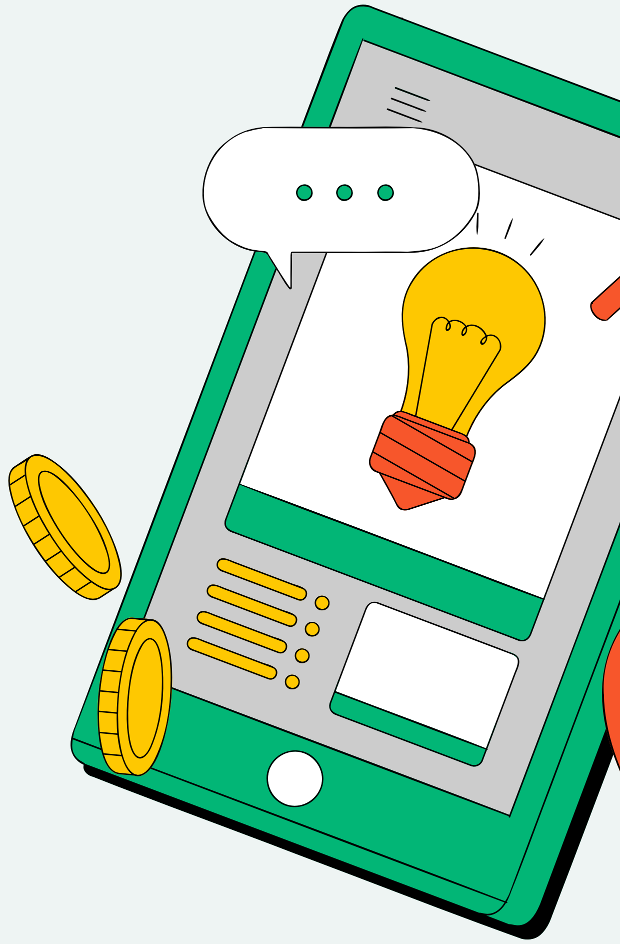
- Type of Dataset: Numerical Dataset
- Number of Classes: 4
  - Class 0: Low Price
  - Class 1: Medium Price
  - Class 2: High Price
  - Class 3: Very High Price

- Total Number of Samples: 2000
- Split of Samples:
  - Training: 80% of the dataset (1600 samples)
  - Testing: 20% of the dataset (400 samples)

ANN MODEL INFORMATION

Feature Extraction Phase

we not used explicit feature extraction mentioned. Instead, the dataset is split into input features (X) and target variable (Y), followed by feature scaling using ` StandardScaler()` .



- **Number of Features**

Before Scaling: Number of Features: 4

Dimension of Features: (2000, 4)

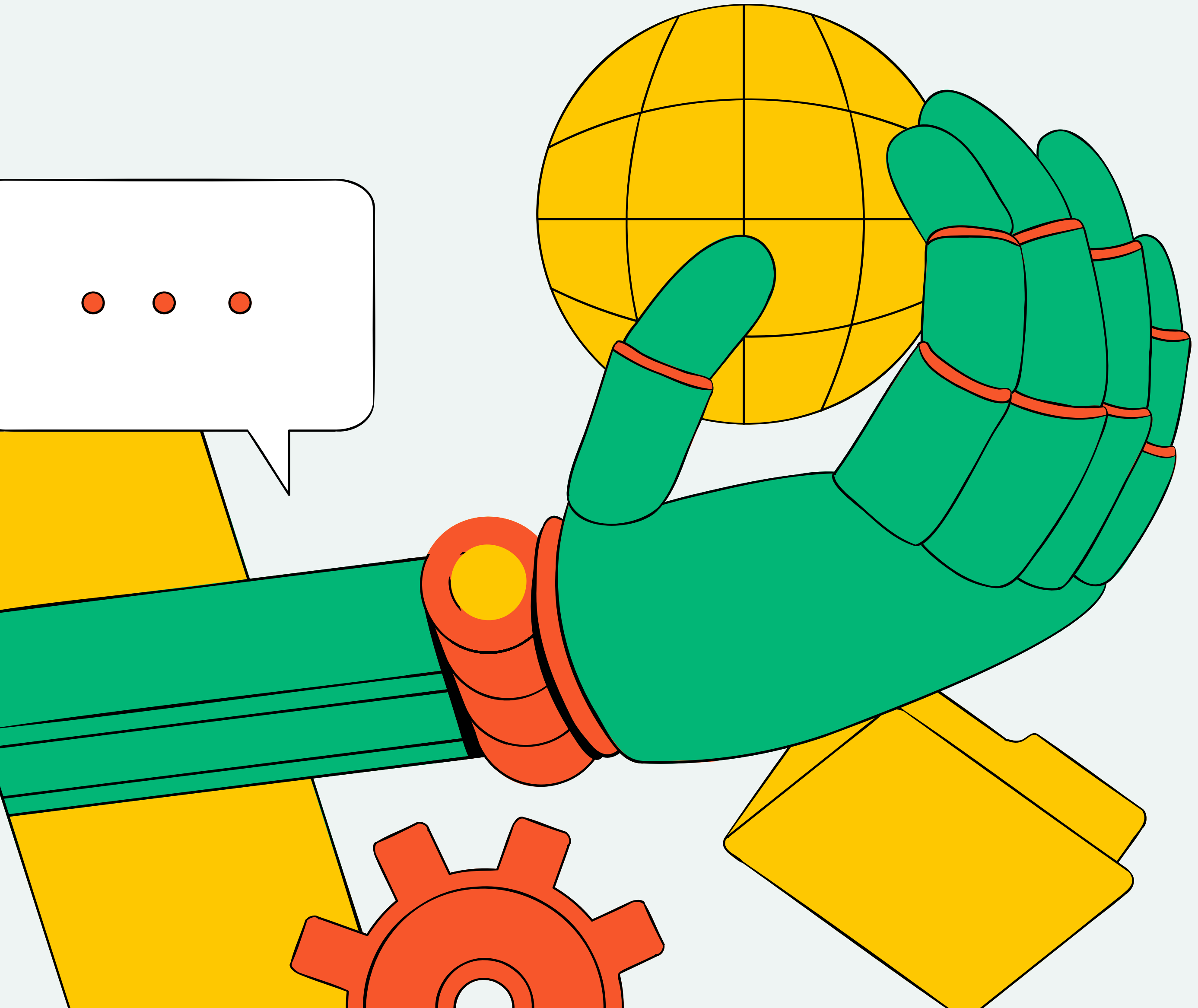
After Scaling:Number of Features: 4

Dimension of Features (Train Data): (1600, 4)

- **Feature Names**

['battery_power', 'px_height', 'px_width', 'ram']

• CROSS-VALIDATION



Cross-validation is used in the implemented model with the **validation_split=0.2** parameter in the **model.fit()** function. This parameter specifies that 20% of the training data will be used as a validation set during training.

HYPERPARAMETERS

OPTIMIZER:

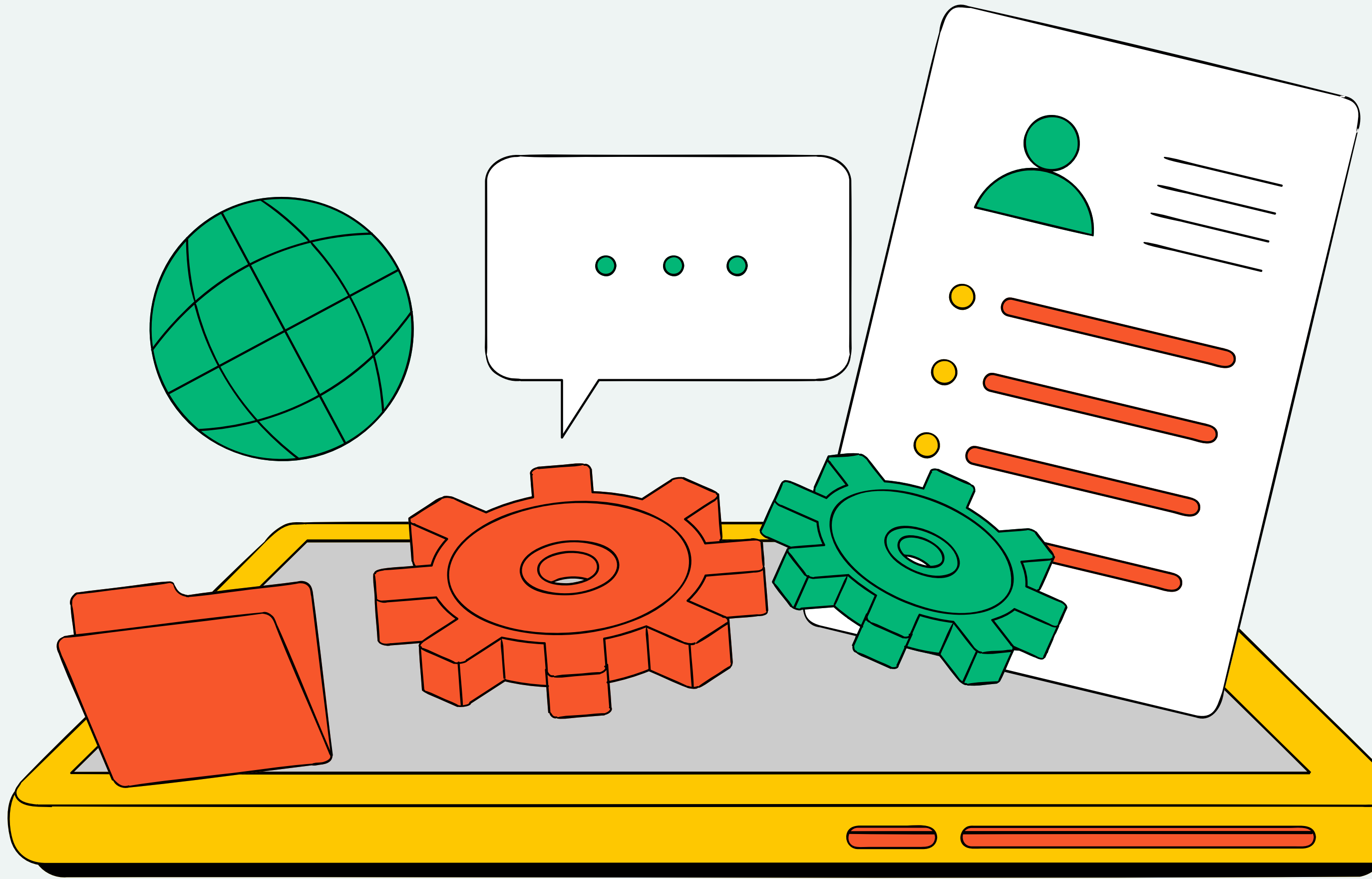
DEFINITION: THE OPTIMIZER IS THE ALGORITHM USED TO UPDATE THE PARAMETERS OF THE MODEL DURING TRAINING TO MINIMIZE THE LOSS FUNCTION.
VALUE: ADAM OPTIMIZER IS USED WITH DEFAULT PARAMETERS
LOSS=SPARSE_CATEGORICAL_CROSSENTROPY

BATCH SIZE:

DEFINITION: BATCH SIZE IS THE NUMBER OF SAMPLES USED IN EACH ITERATION OF TRAINING.
VALUE: BATCH_SIZE=32 IS USED DURING TRAINING.

NUMBER OF EPOCHS:

DEFINITION: EPOCHS REFER TO THE NUMBER OF TIMES THE ENTIRE DATASET IS PASSED FORWARD AND BACKWARD THROUGH THE NEURAL NETWORK DURING TRAINING.
VALUE: EPOCHS=20 IS USED FOR TRAINING THE MODEL.

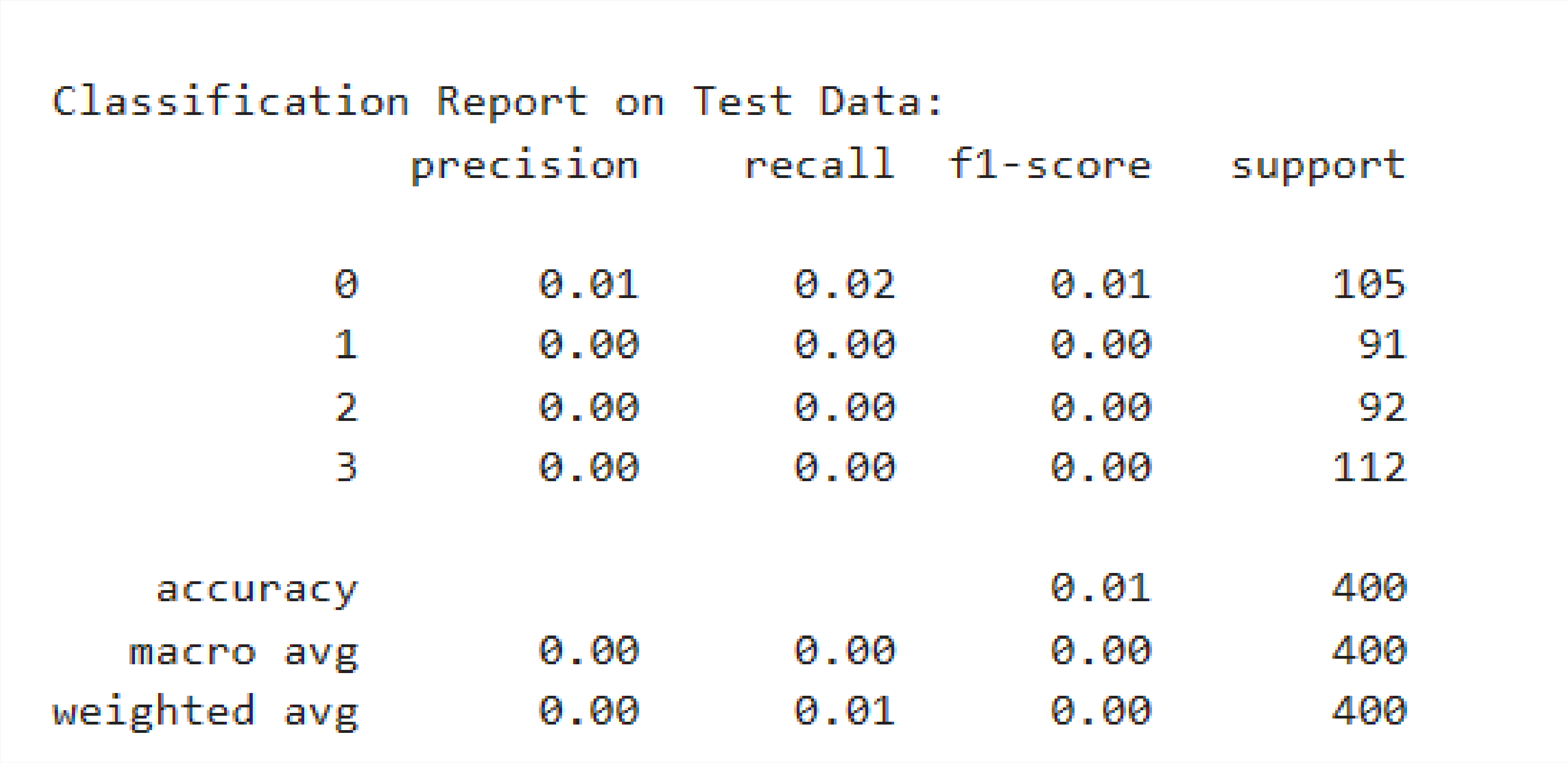


ANN MODEL RESULTS

- ACCURACY

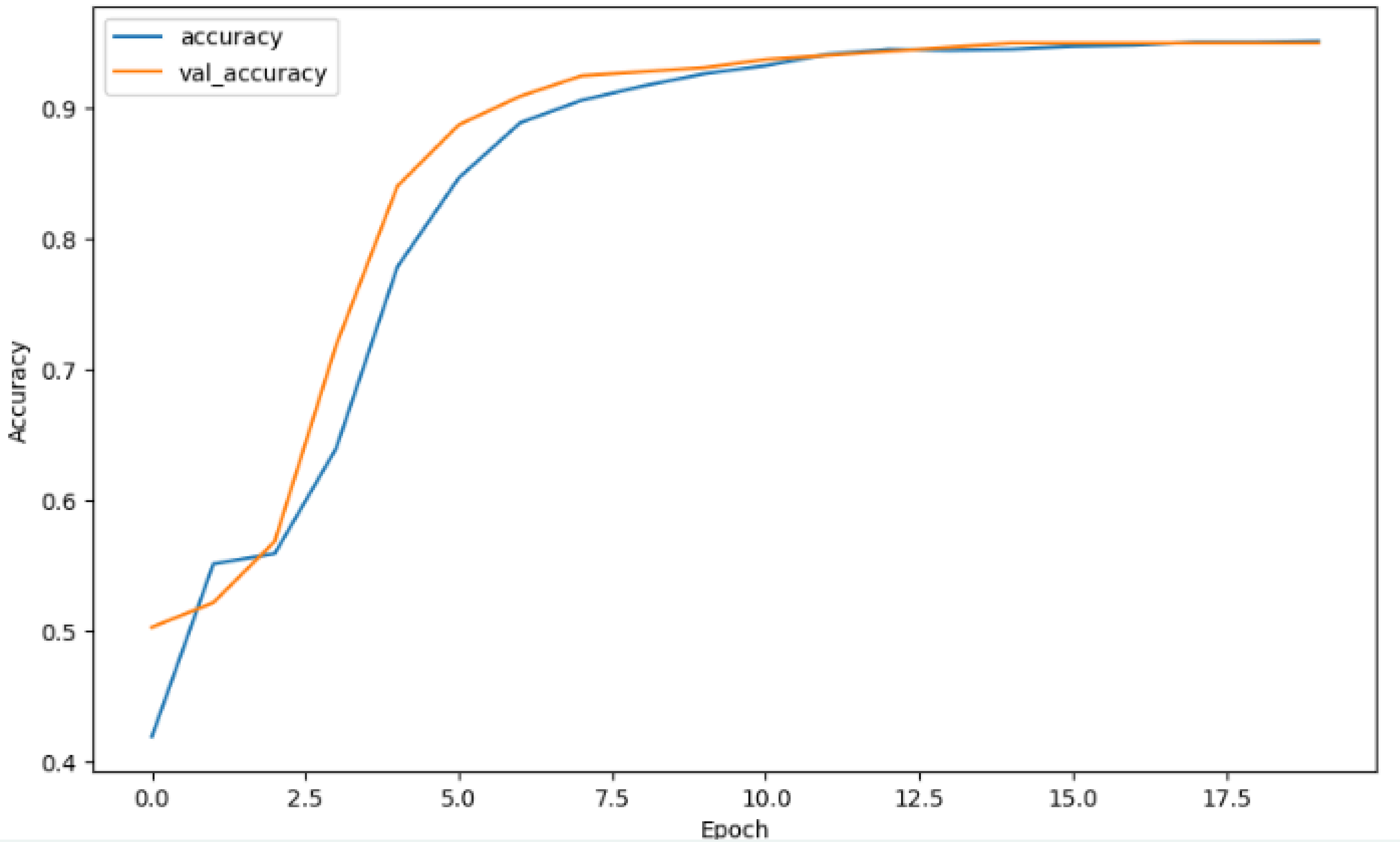
TEST ACCURACY: 0.9524999856948853

- CLASSIFICATION REPORT DIGRAM

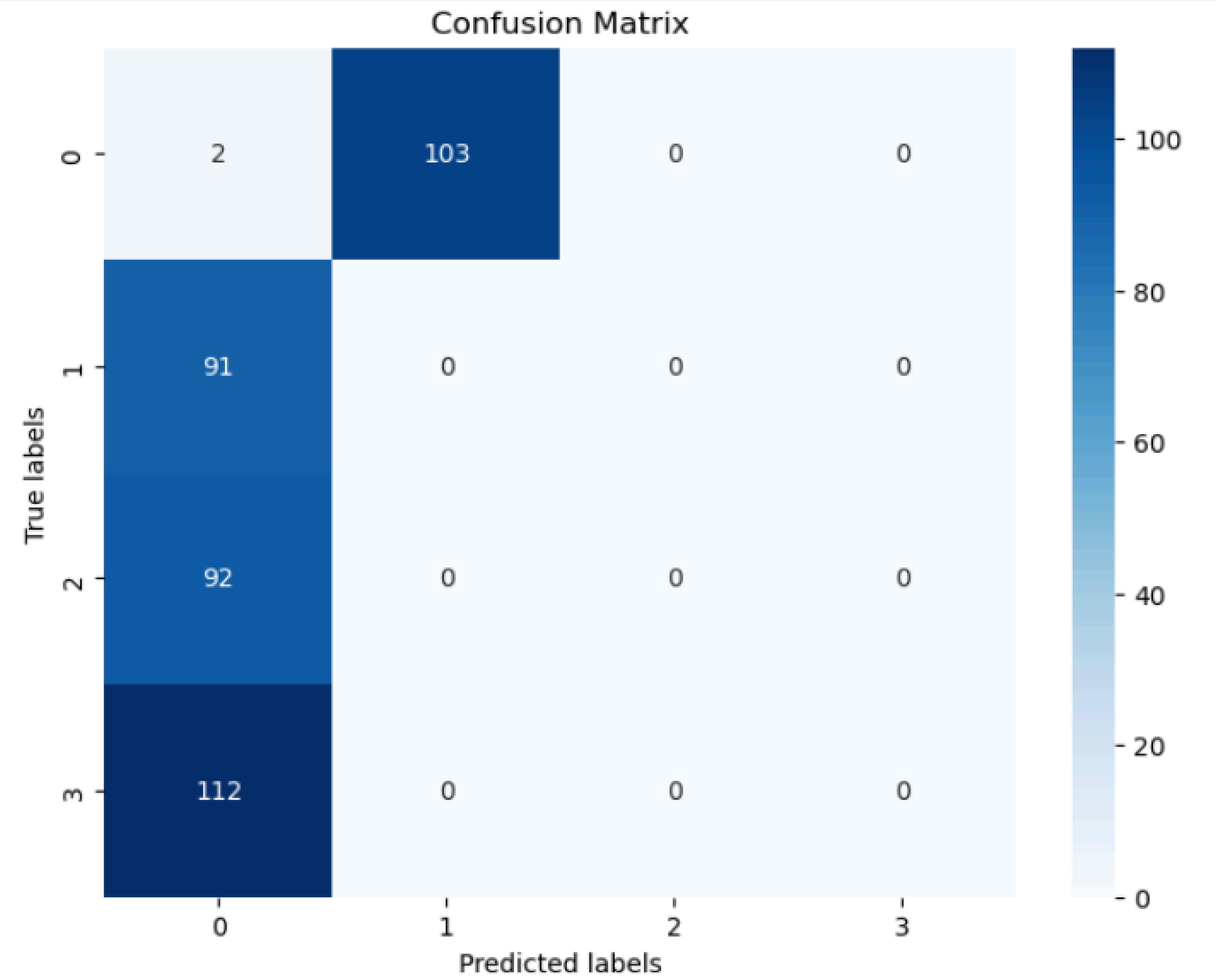


```
Classification Report on Test Data:
              precision    recall  f1-score   support

           0       0.01      0.02      0.01       105
           1       0.00      0.00      0.00        91
           2       0.00      0.00      0.00        92
           3       0.00      0.00      0.00       112

    accuracy                           0.01       400
   macro avg       0.00      0.00      0.00       400
weighted avg       0.00      0.01      0.00       400
```

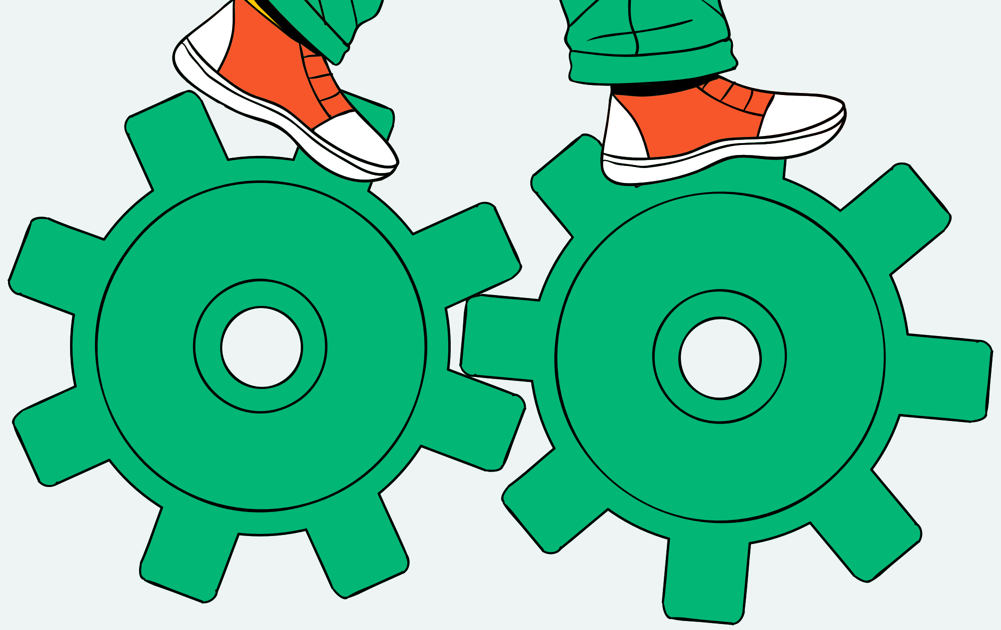
PLOT TRAINING HISTORY



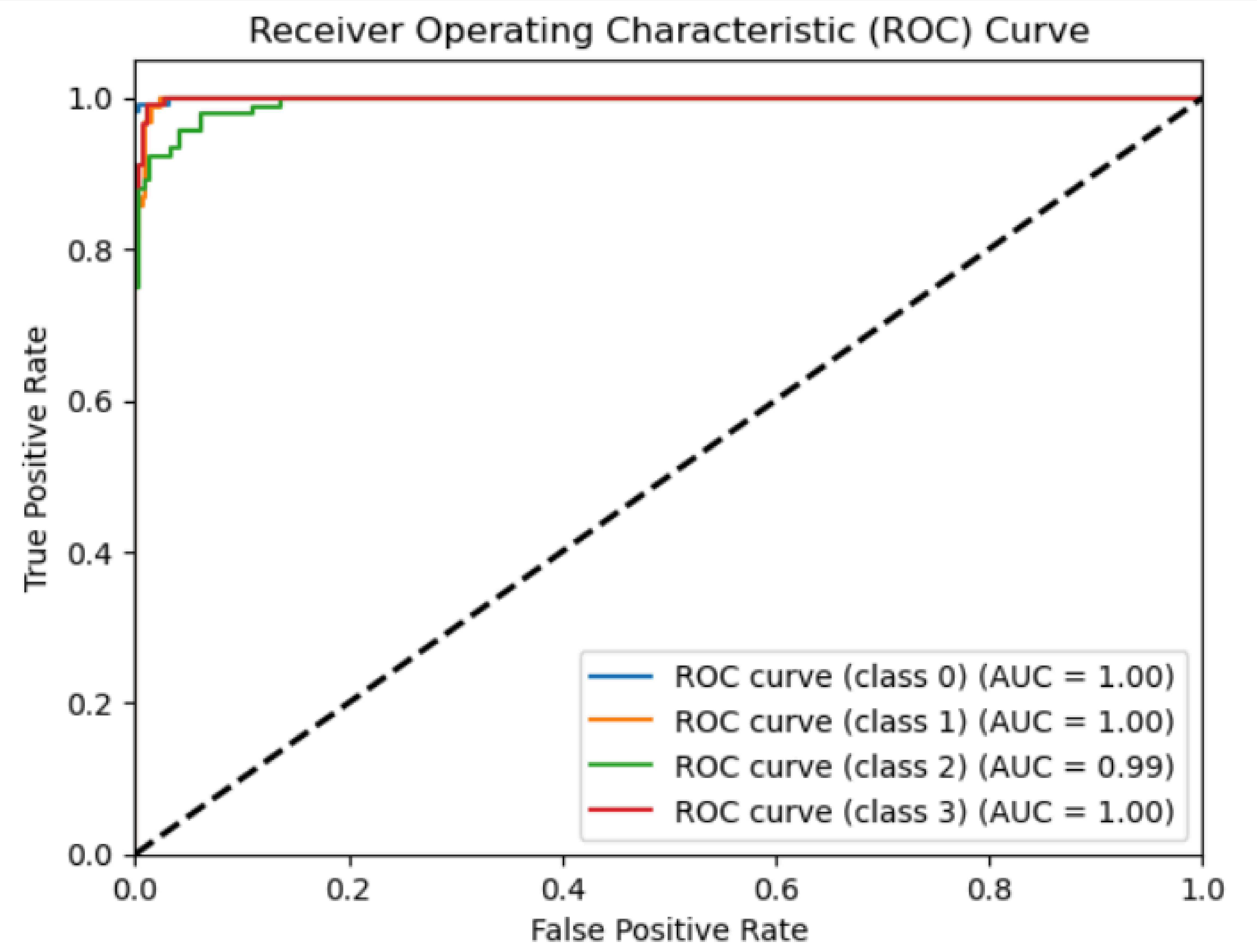
PLOT CONFUSION MATRIX



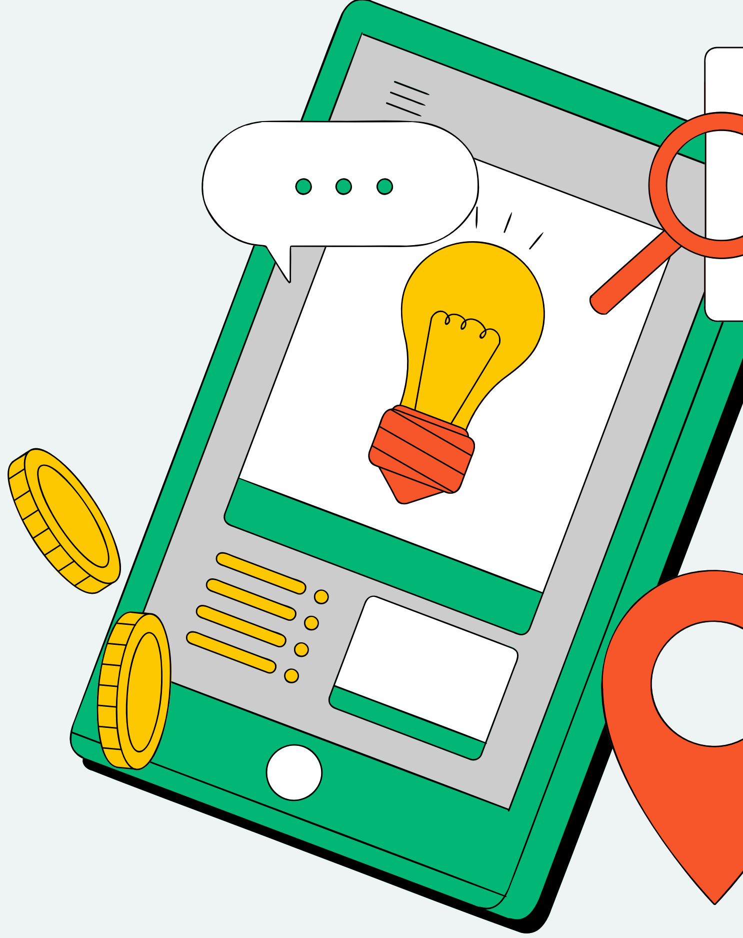
	0	1	2	3
0	2	103	0	0
1	91	0	0	0
2	92	0	0	0
3	112	0	0	0



PLOT ROC CURVE



DECISION TREE MODEL INFORMATION



- ## Feature Extraction Phase

no explicit feature extraction phase is mentioned. The features are simply split into input features (X) and target variable (Y), and then they are used directly for modeling.

- ## Number of Features

The number of features extracted corresponds to the number of columns in the DataFrame **data** after drop unimportant columns is equal 4

- ## THE DIMENSION OF THE RESULTED FEATURES

can be obtained from the shape of the DataFrame X is equal 2000

- ## Feature Names

['battery_power', 'px_height', 'px_width', 'ram']

CROSS-VALIDATION

Individual Scores: Each fold represents a separate evaluation of the model on a different subset of the data. These scores indicate the accuracy achieved by the model on each fold.
Mean Score: The average accuracy across all folds, providing an overall assessment of the model's performance.
Standard Deviation of Scores: Indicates the variability or spread of the accuracy scores around the mean. A lower standard deviation suggests more consistent performance across folds.

k=5 folds

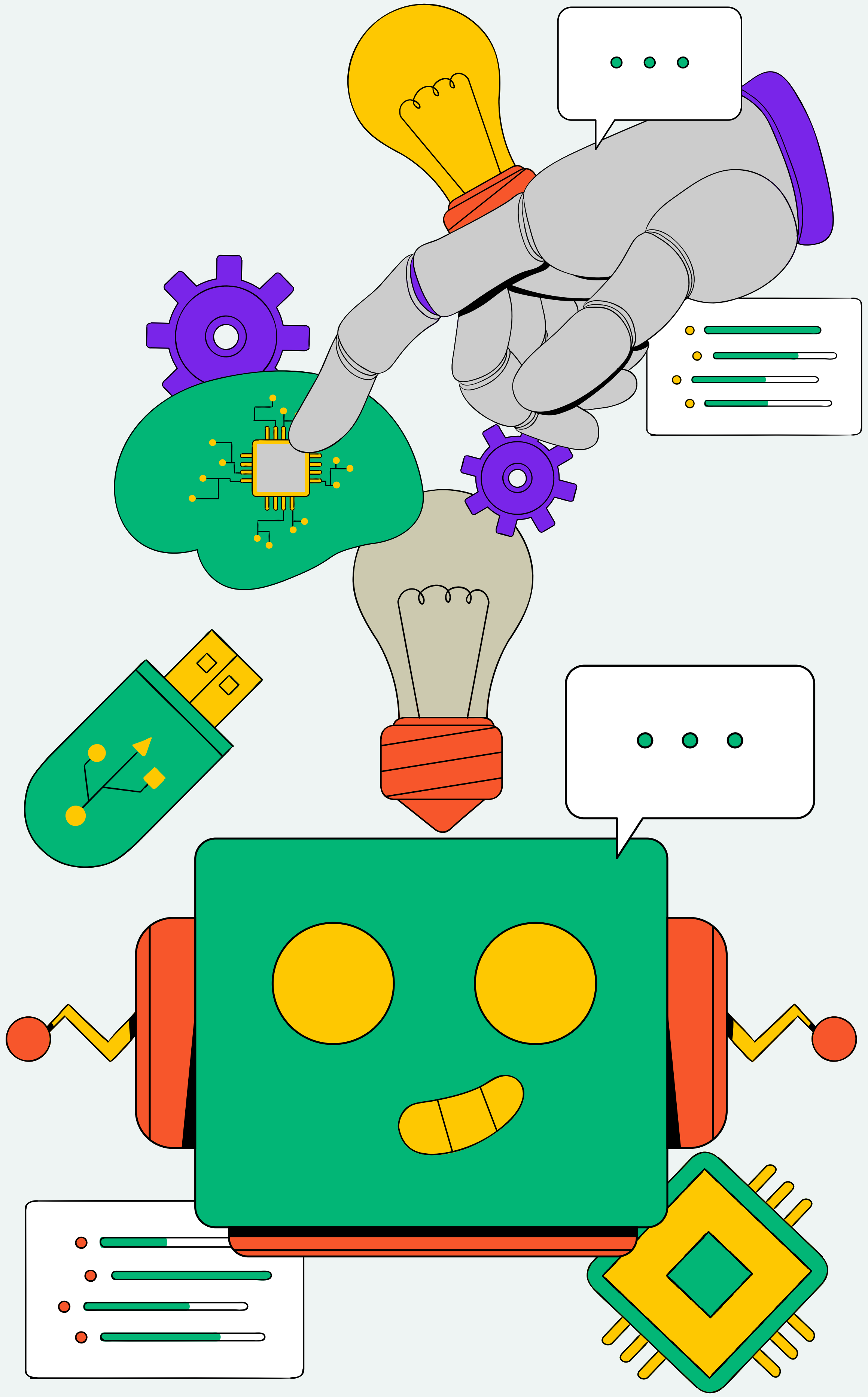
Individual Scores:
Fold 1: 0.85
Fold 2: 0.82
Fold 3: 0.88
Fold 4: 0.87
Fold 5: 0.84
Mean Score: 0.852
Standard Deviation of Scores: 0.022

HYPERPARAMETERS

- ## DECISION TREE HYPERPARAMETERS

  - **Criterion**: entropy
  - **Min Samples Split**: 10
  - **Min Samples Leaf**: 4
  - **Splitter**: 'best'

These hyperparameters are used in the DecisionTreeClassifier() initialization.



MODEL EVALUATION HYPERPARAMETERS

- **Metric**: Accuracy score is used for model evaluation.

PLOTTING HYPERPARAMETERS

MAX DEPTH VALUES: A RANGE FROM 1 TO 21 IS EXPLORED TO PLOT THE DECISION TREE'S LOSS CURVE.

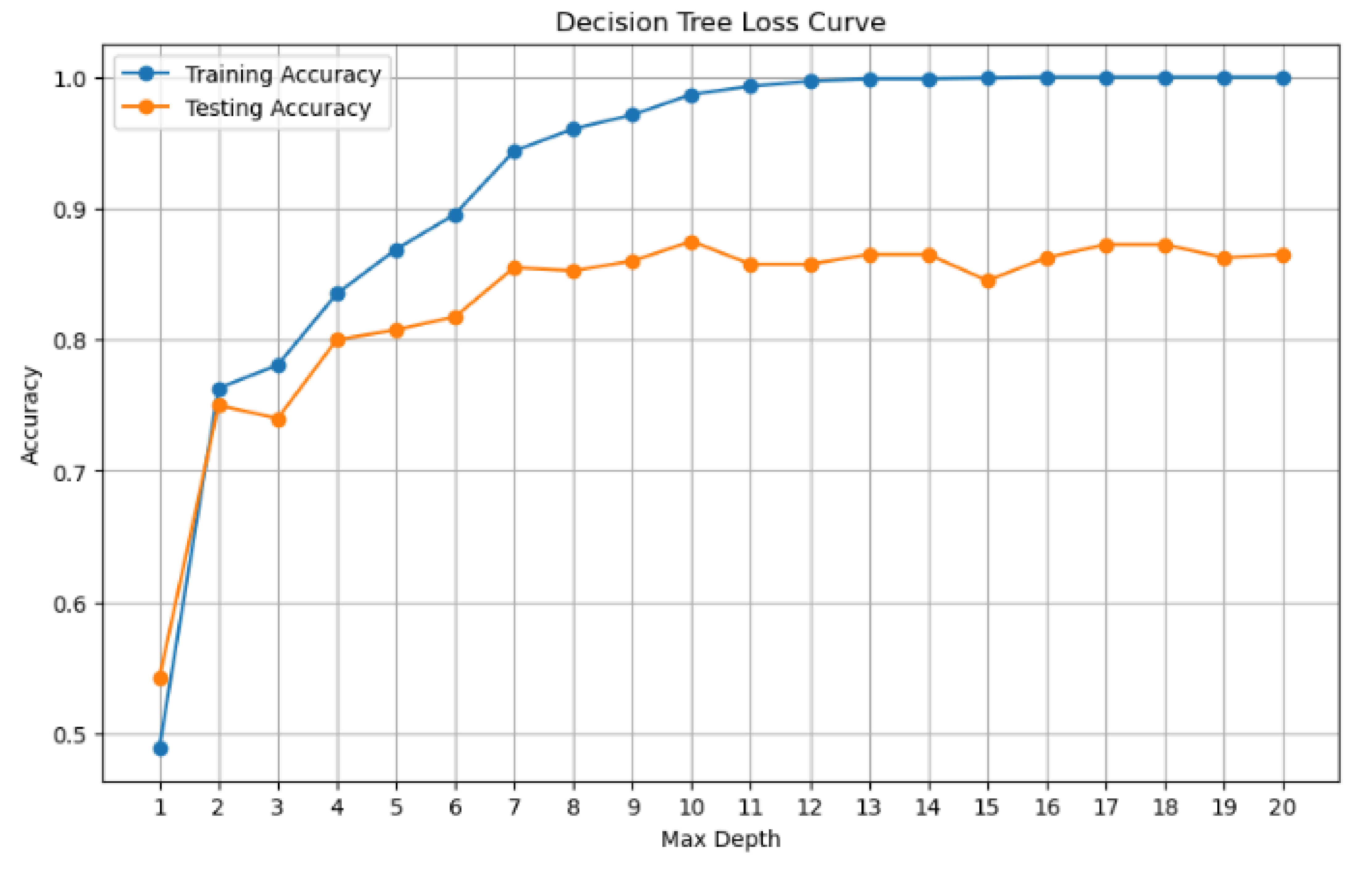
DECISION TREE MODEL RESULTS

- ACCURACY

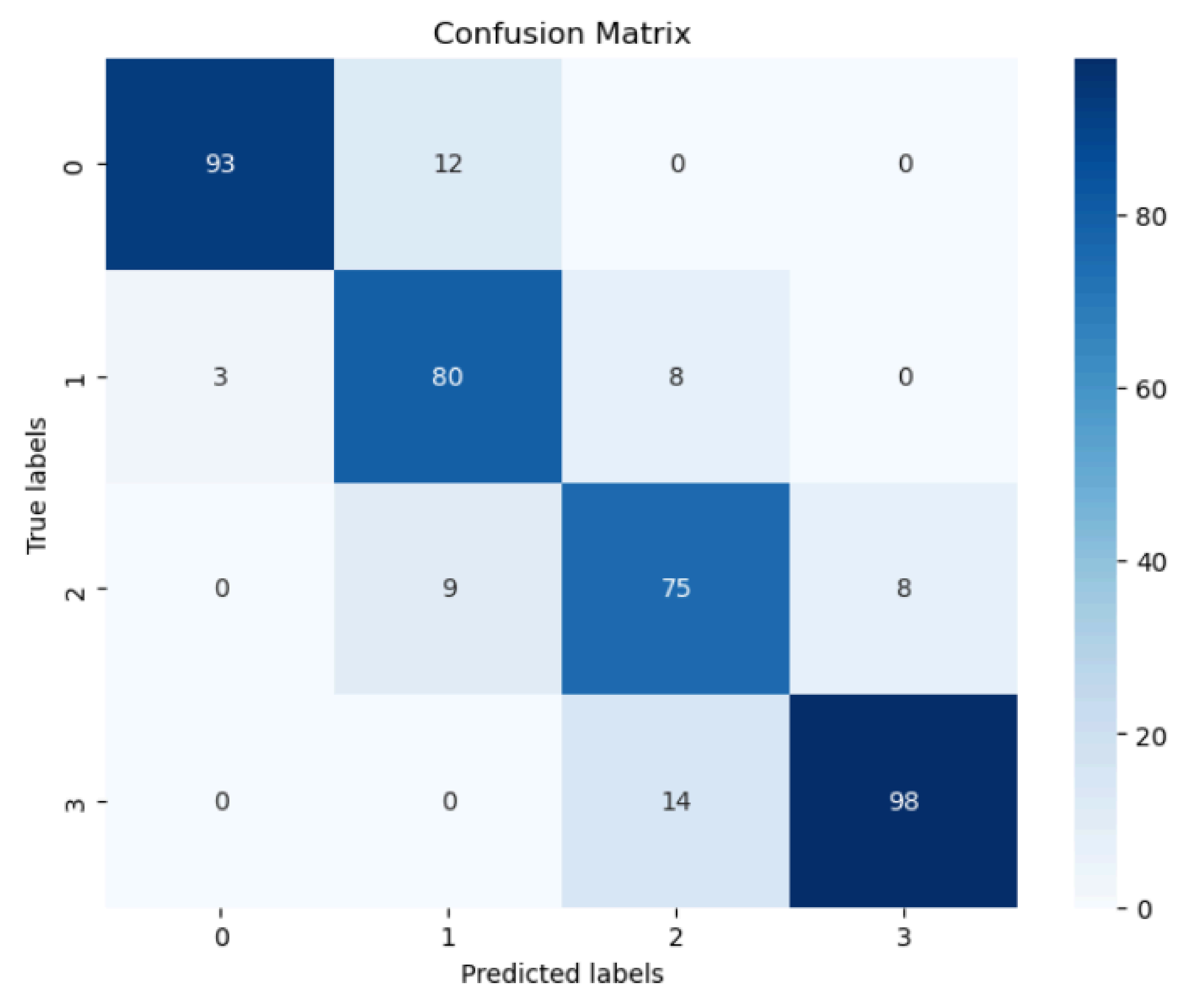
Training-set accuracy score: 0.9625

Test-set accuracy score: 0.8575

- **PLOT THE LOSS CURVE**



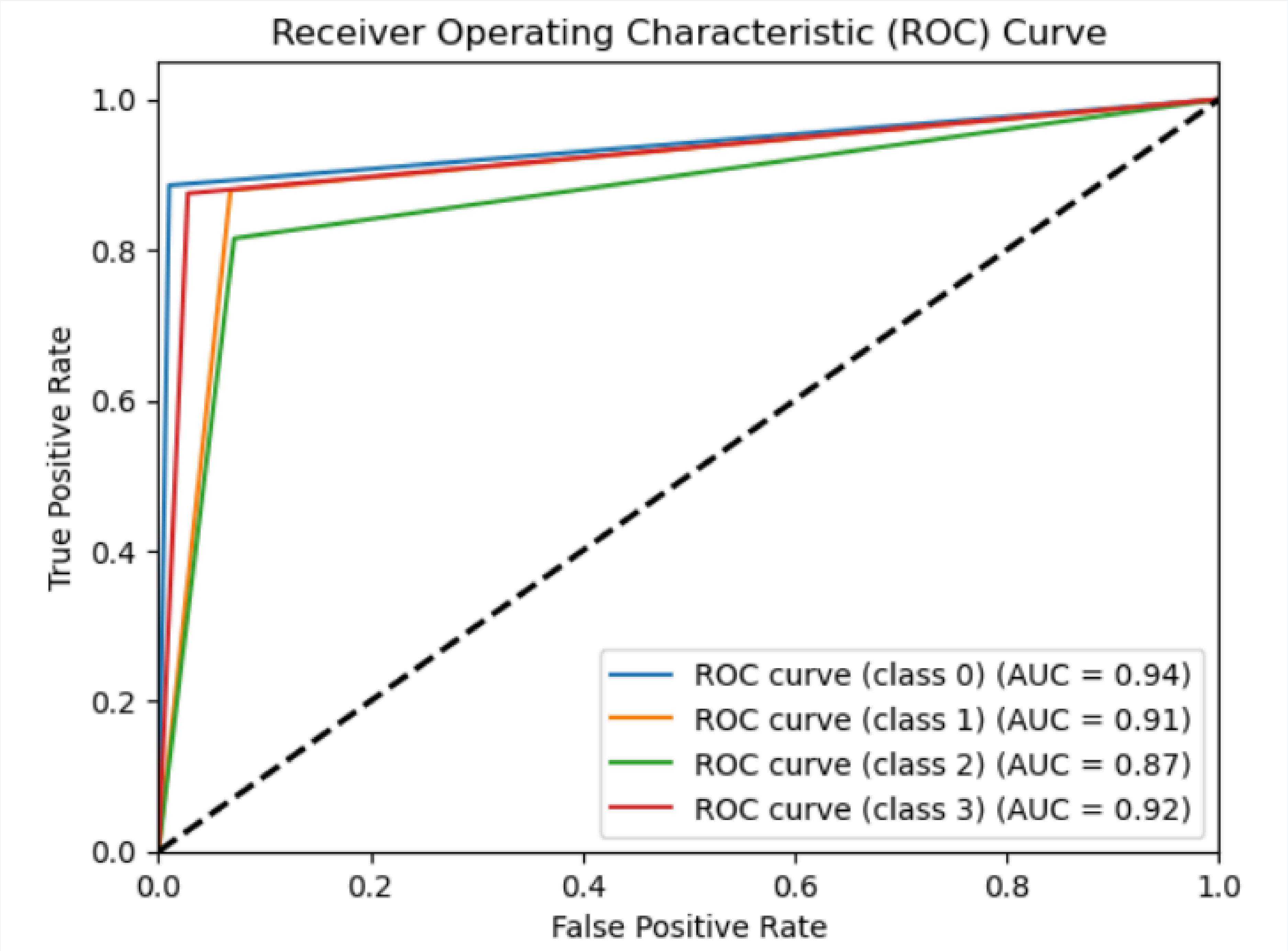
- **CONFUSION MATRIX**



Confusion Matrix

True labels \ Predicted labels	0	1	2	3
0	93	12	0	0
1	3	80	8	0
2	0	9	75	8
3	0	0	14	98

- **PLOT ROC_CURVE**



INTRODUCTION TO INSURANCE DATASET

This application aims to predict insurance costs for individuals based on various factors such as age, sex, BMI (Body Mass Index), number of children, smoking habits, and region. The prediction is made using a trained Support Vector Machine (SVM) regression model.

The dataset underpinning this predictive system encapsulates a wealth of information on individuals and their corresponding insurance costs. Each meticulously curated data point encompasses critical attributes, including:

- **Age**: The individual's chronological age.
- **Sex**: Gender categorization (Male/Female).
- **BMI**: The Body Mass Index, a key health indicator derived from height and weight.
- Children: The number of dependent children.
- **Smoker**: Binary indication of smoking habits (Yes/No).
- **Region**: Geographic classification of the individual's residence (Southeast, Southwest, Northeast, Northwest).

OBJECTIVE

The primary objective of the Insurance Cost Prediction system is to accurately estimate insurance costs for individuals based on their demographic and health-related attributes. This system serves various stakeholders, including insurance companies, policyholders, and healthcare providers,

INFORMATION ABOUT INSURANCE DATASET

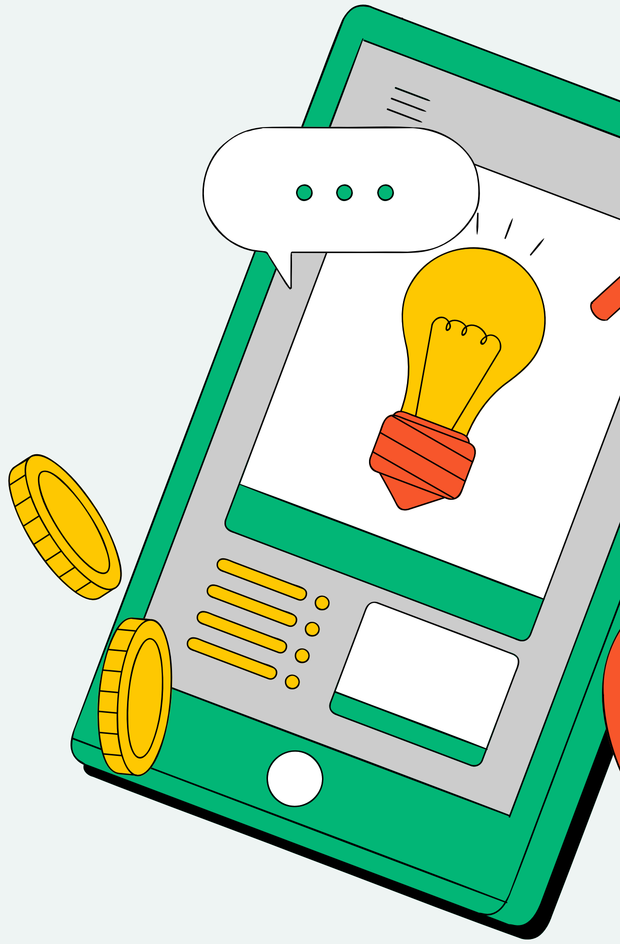
- **Type of Dataset: Numerical Dataset**
- **Total Number of Samples: 1338**

- **Split of Samples:**
  - **Training: 80% of the dataset (1070 samples)**
  - **Testing: 20% of the dataset (268 samples)**

SVM MODEL INFORMATION

Feature Extraction Phase

The number of features extracted, their names, and the dimension of the resulting features can be obtained from the shape of the input dataset (X) before and after feature scaling. Before scaling, the number of features and their names are determined by the columns of the dataset. After scaling, the dimension of the resulting features is obtained from the shape of the scaled dataset (X_train_scaled or X_test_scaled).



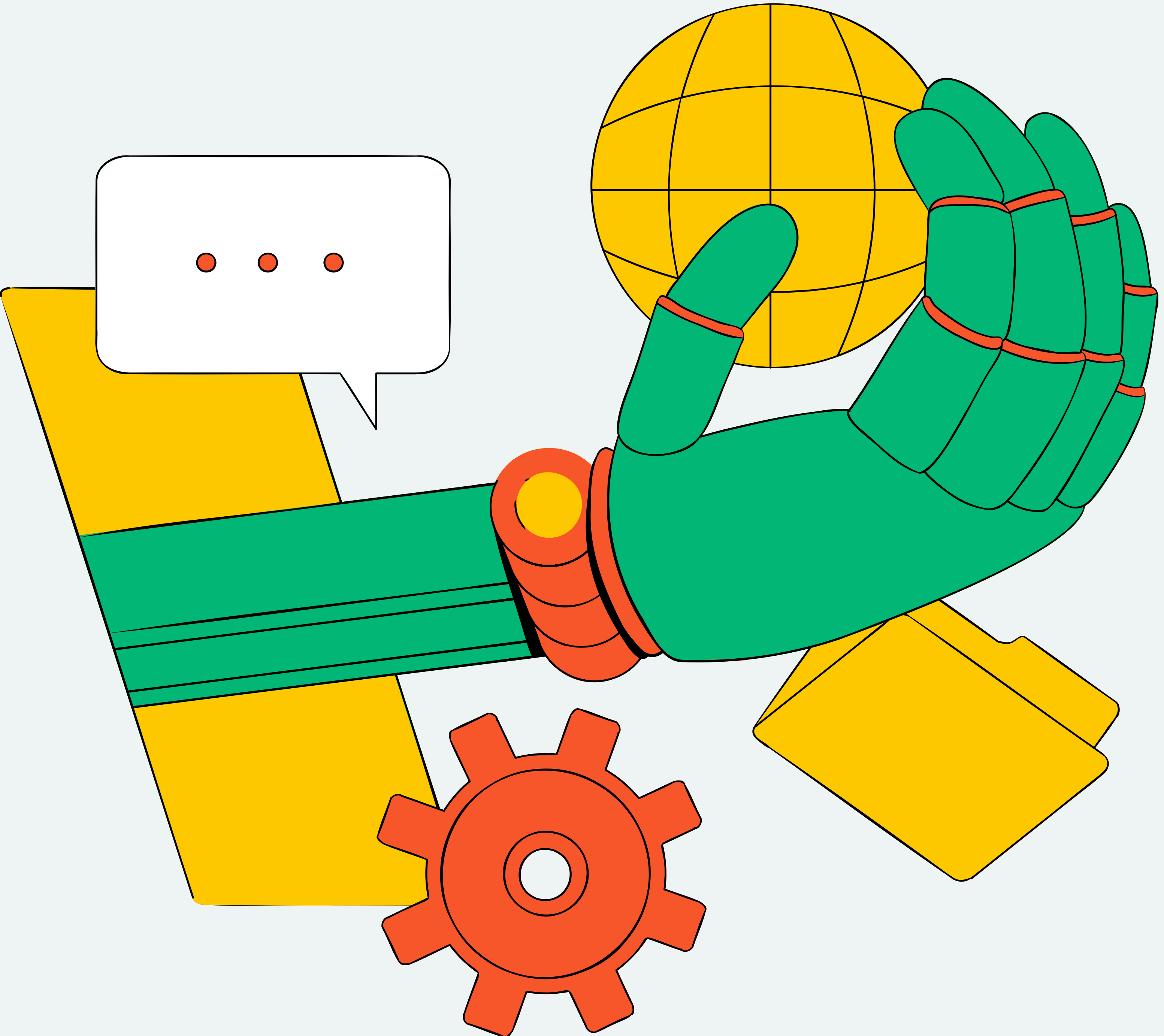
- **Number of Features**

Before Scaling: Number of Features: 6
Dimension of Features: (1338, 6)
After Scaling:Number of Features: 6
Dimension of Features (Train Data): (1070,6)

- **Feature Names**

['age', 'sex', 'bmi', 'children', 'smoker', 'region']

• CROSS-VALIDATION

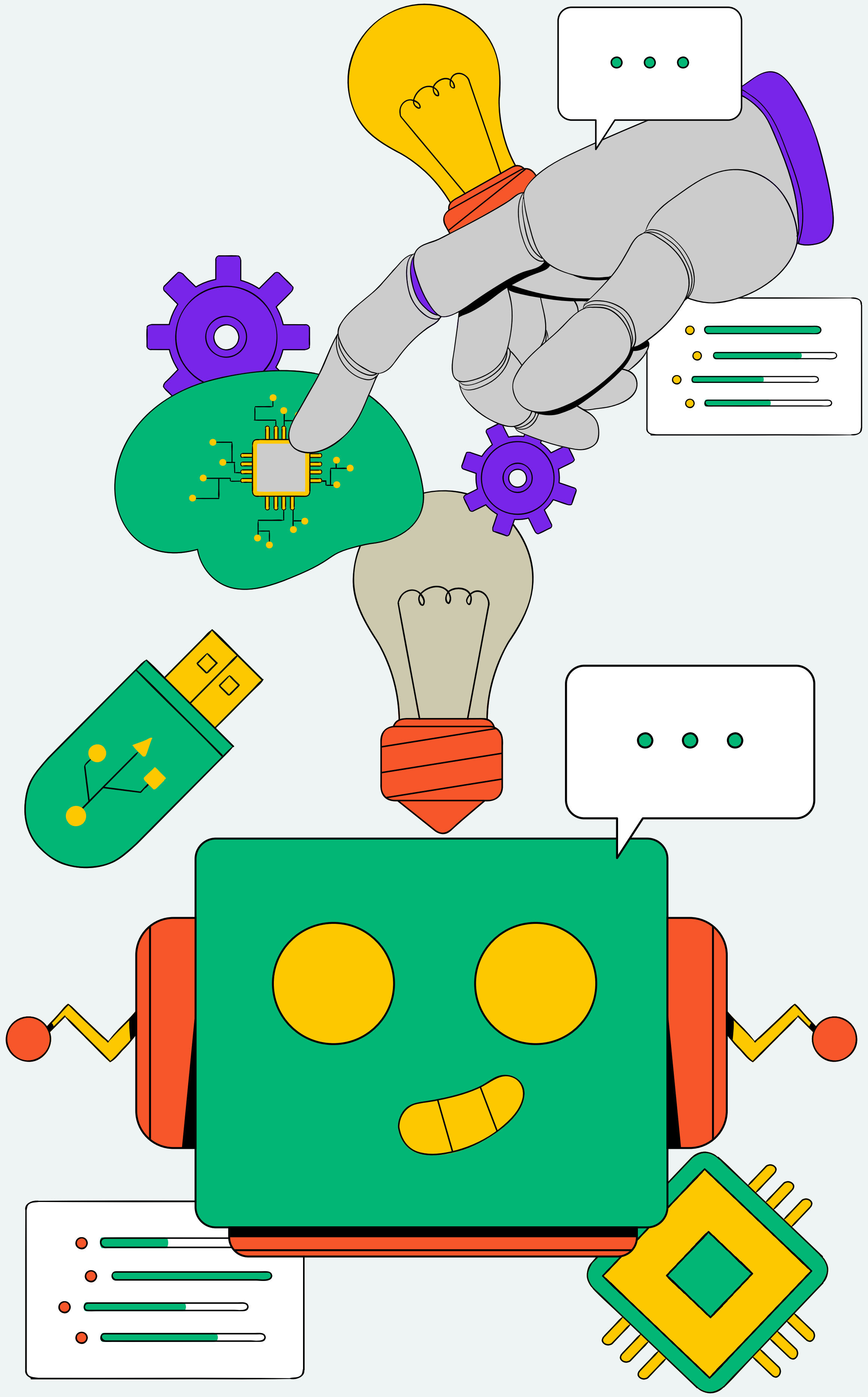


Cross-Validation: Cross-validation is used to evaluate the performance of the Support Vector Regression (SVR) model. It helps assess the model's generalization ability and reduce overfitting. In the provided code, 5-fold cross-validation is implemented (**cv=5**), meaning the dataset is divided into 5 equal parts, and the model is trained and evaluated 5 times, each time using a different part as the validation set.

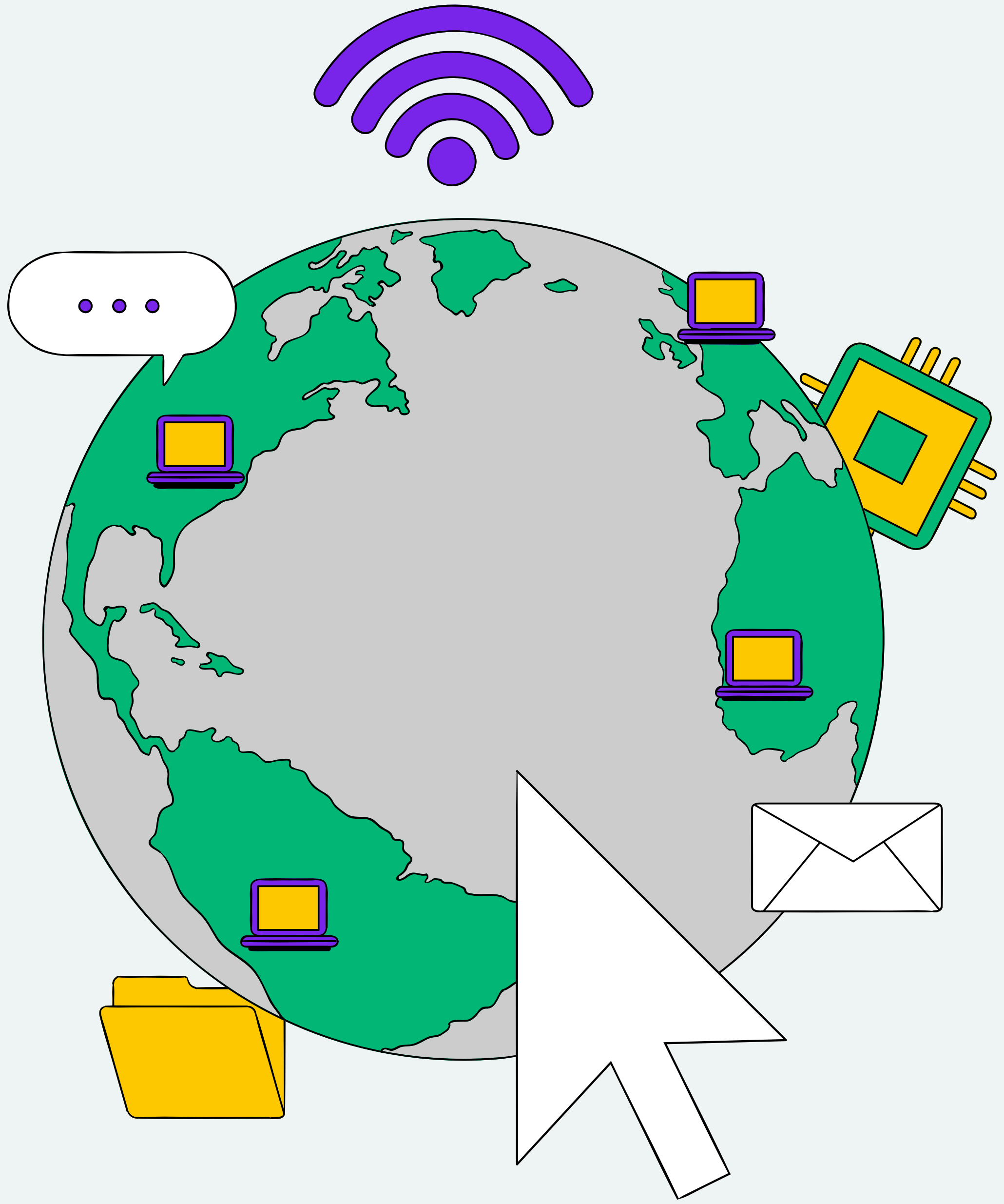
```
Cross-validation scores: [0.80883247
0.88107672 0.7878651  0.74174881 0.84779605]
Mean CV R-squared: 0.8134638303705287
```

HYPERPARAMETERS

- Kernel: 'rbf' (Radial Basis Function)
- Regularization parameter (C): 3000
- Kernel coefficient for 'rbf' (gamma): 0.1



EVALUATION METRICS



- R-squared (R2) value is used as an evaluation metric to assess the goodness-of-fit of the model to the training and test data. It measures the proportion of the variance in the dependent variable that is predictable from the independent variables.

- Mean Squared Error (MSE) and Mean Absolute Error (MAE) are calculated to quantify the prediction errors of the model on the test data.

SVM RESULTS

- ACCURACY

R squared value for training data: 0.8258480280176116

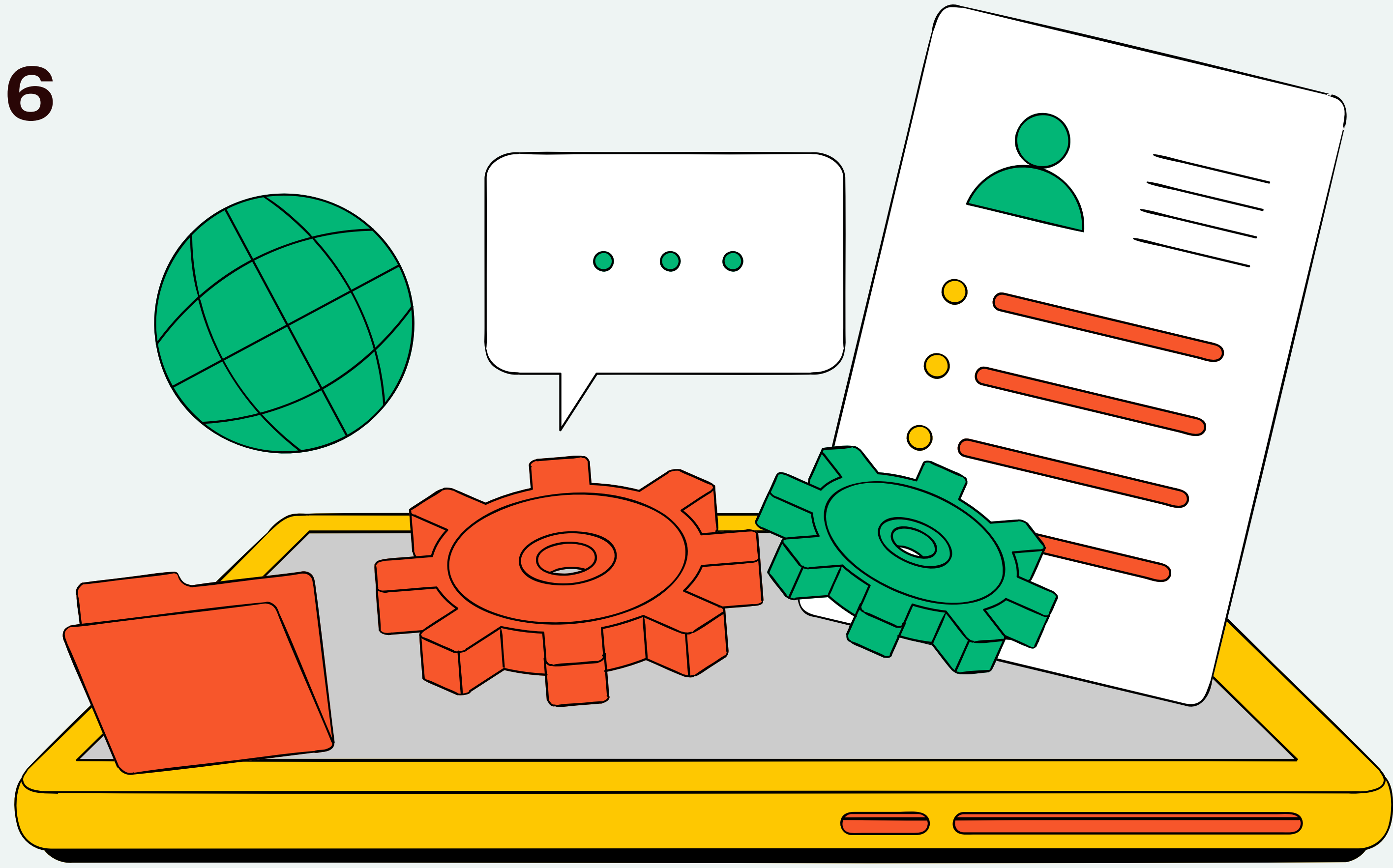
R squared value for test data: 0.8502438984589014

- MEAN SQUARED ERROR

Mean Squared Error (MSE): 23249444.797408793

- MEAN ABSOLUTE ERROR

MEAN ABSOLUTE ERROR (MAE): 1920.5303423097982



THANKS