THYROID DISEASE DETECTION

Aditya Marne
Soham Jagtap
Adnan Sheikh
Sufiyan Shaikh

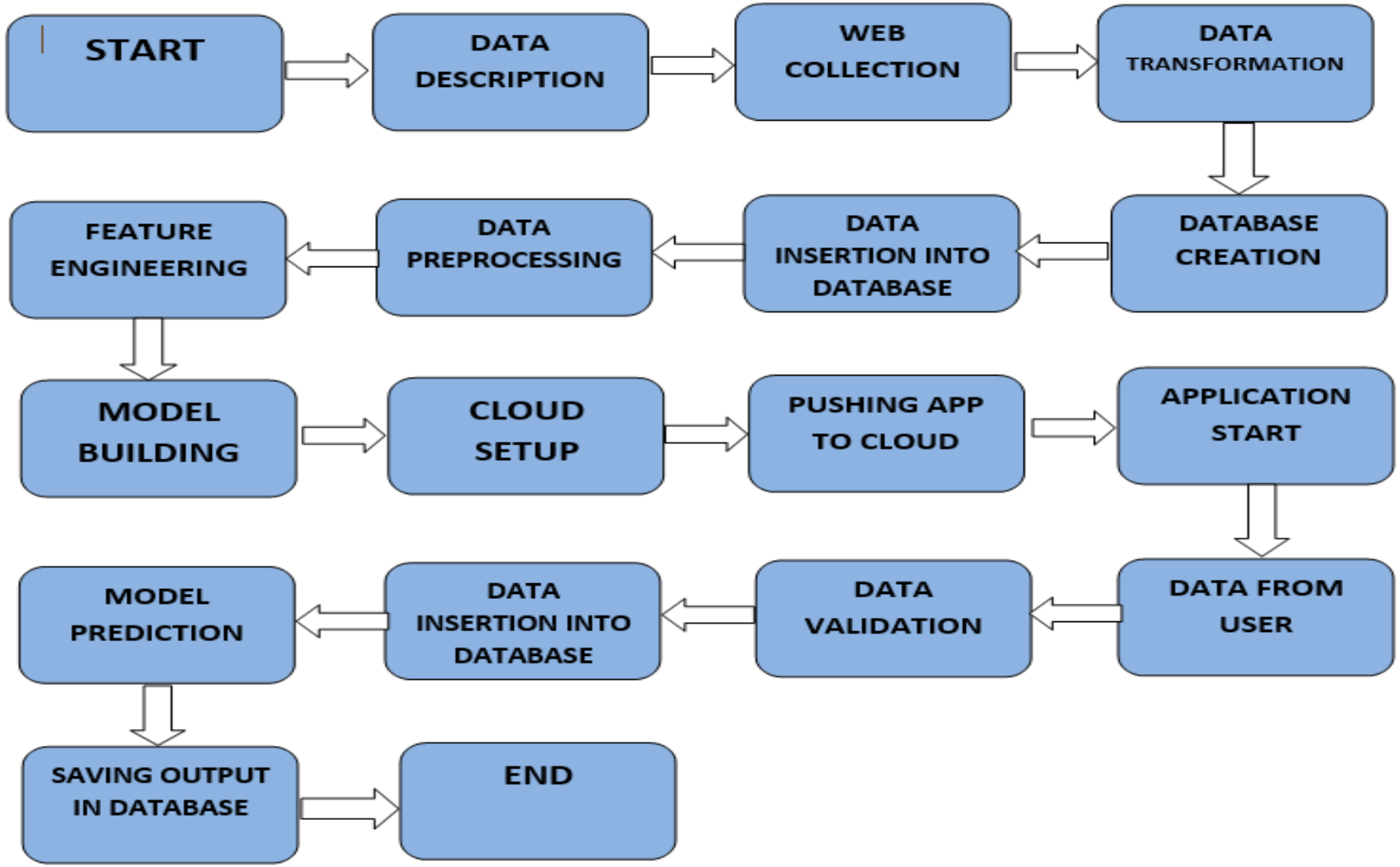
INTRODUCTION

Thyroid disease a very common problem in India, more than one crore people are suffering with the disease every year. Especially it is more common in female. Hyperthyroidism and hypothyroidism are the most two common diseases caused by irregular function of thyroid gland. Thyroid disorder can speed up or slow down the metabolism of the body. In the world of rising new technology and innovation, healthcare industry is advancing with the role of Artificial Intelligence. Machine learning algorithms can help to early detection of the disease and to improve the quality of the life. This study demonstrates the how different classification algorithms can forecasts the presence of the disease. Different classification algorithms such as Logistic regression, Random Forest, Decision Tree, Naïve Bayes, Support Vector Machine have been tested and compared to predict the better outcome of the model.

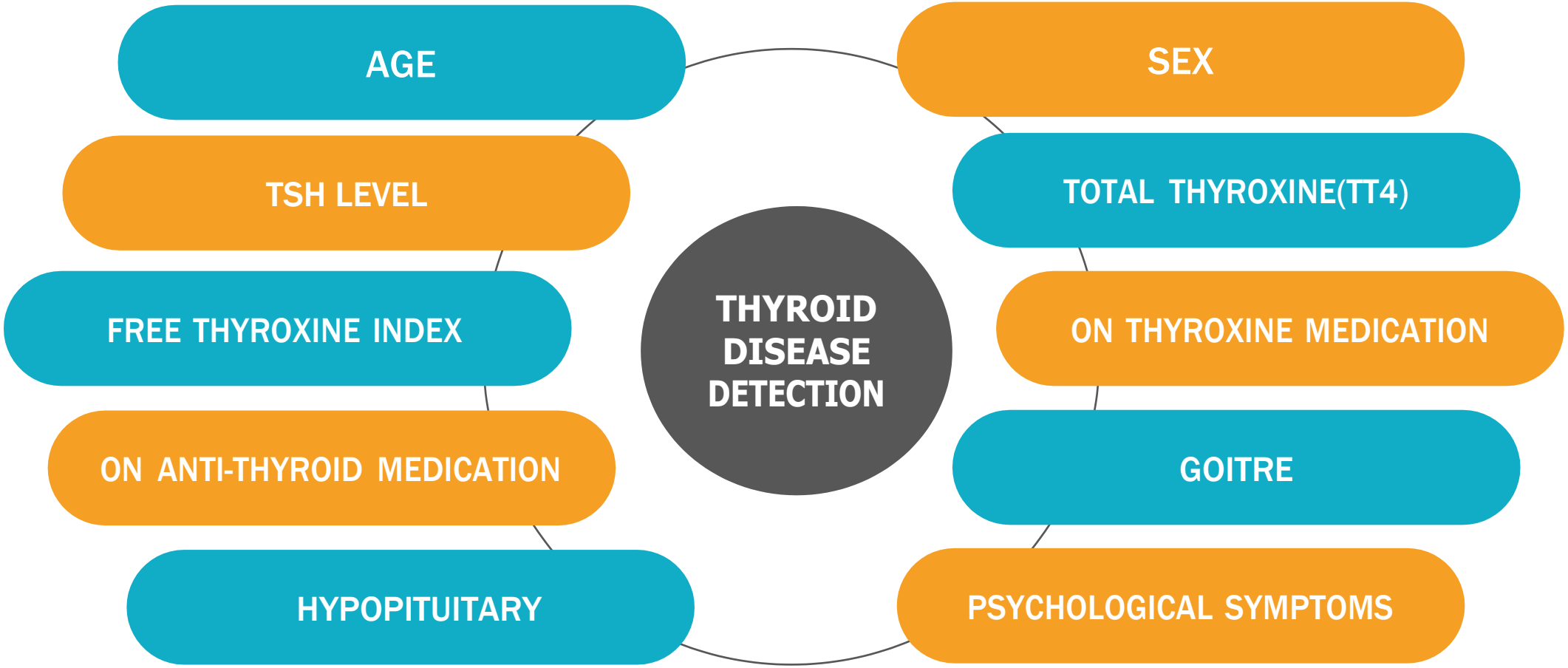
OBJECTIVE

The main goal of this project is to predict the risk of hyperthyroid and hypothyroid based on various factors of individuals. Thyroid disease is a common cause of medical diagnosis and prediction, with an onset that is difficult to forecast in medical research. It will play a decisive role in order to early detection, accurate identification of the disease and helps the doctors to make proper decisions and better treatment.

ARCHITECTURE



DATASET


AGE
SEX
TSH LEVEL
TOTAL THYROXINE(TT4)
FREE THYROXINE INDEX
THYROID DISEASE DETECTION
ON THYROXINE MEDICATION
ON ANTI-THYROID MEDICATION
GOITRE
HYPOPITUITARY
PSYCHOLOGICAL SYMPTOMS

DATA ANALYSIS STEPS

DATA COLLECTION

In step 1, we collect data which is generally present in a database or on internet.

DATA PREPROCESSING

In step 2, we preprocess the data which involves data cleaning by handling outliers, null values etc.

EXPLORATORY DATA ANALYSIS

In step 3, we explore the data by performing univariate and bivariate analysis on the features.

FEATURE SELECTION

In step 4, we use feature selection techniques to filter out the most important features to perform model creation

MODEL CREATION AND EVALUATION

In step 5, we finally build models on our dataset and choose the model which gives the best accuracy.

Decision Tree Algorithm

INTRODUCTION

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.
- It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. It's gives us high accuracy as 99%.

Decision Tree Algorithm

ADVANTAGES

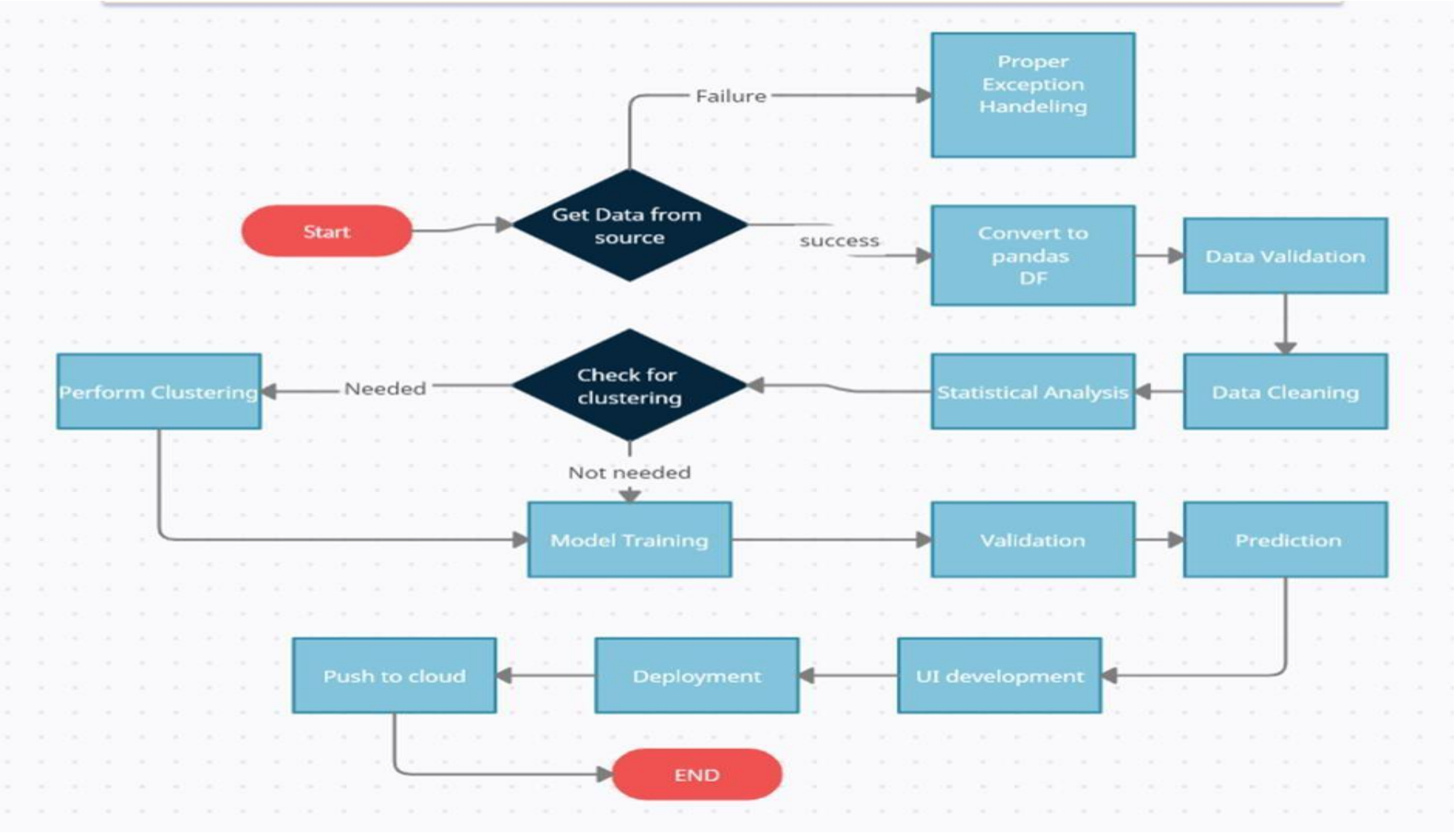
- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Decision Tree Algorithm

DISADVANTAGES

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.
- For more class labels, the computational complexity of the decision tree may increase

MODEL TRAINING AND VALIDATION WORKFLOW

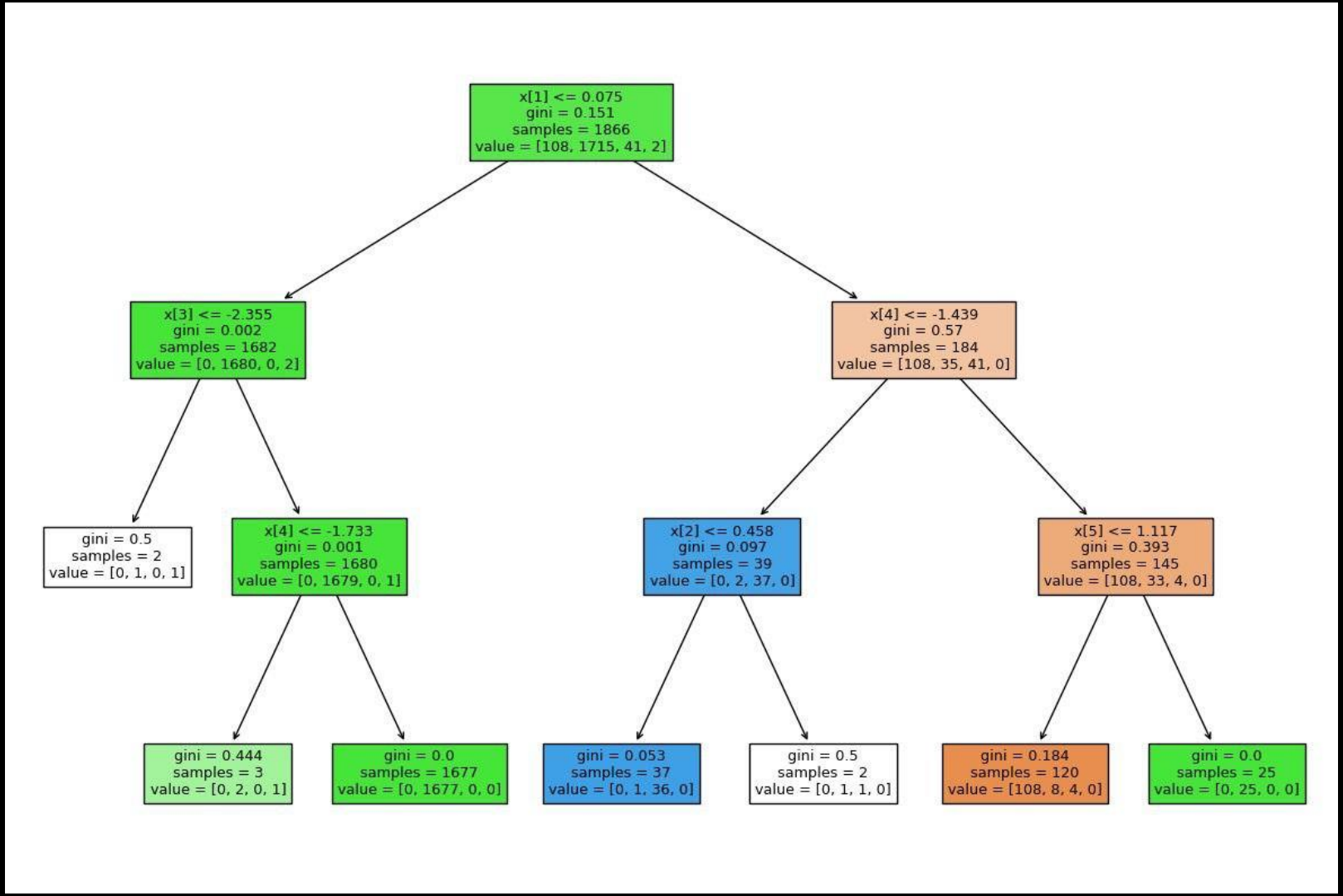


MODEL PREDICTION RESULTS ON TRAINING DATASET

```python
from sklearn.metrics import accuracy_score,classification_report
# Model training score
print(f"Model Score : {model_2.score(X_train,y_train)}")

#model prediction
y_pred = model_2.predict(X_test)

# Accuacy Score
print(f"Accuracy Score : {accuracy_score(y_test,y_pred)}")

#Classification Report
print(f"Classification Report : \n {classification_report(y_test,y_pred)}")
```

```
Model Score : 0.9914255091103966
Accuracy Score : 0.9925
Classification Report :
              precision    recall  f1-score   support

           0       0.89      1.00      0.94        39
           1       1.00      0.99      1.00       740
           2       0.95      0.90      0.93        21

    accuracy                           0.99       800
   macro avg       0.95      0.97      0.95       800
weighted avg       0.99      0.99      0.99       800
```

MODEL IMPLEMENTATION USING DECISION TREE



THANK YOU