

**FAKE NEWS RECOGNITION**  Submitted by

**BALAJI VENKATESH.K (231801019)**

**BALAJI.E(231801017)**

**AI23331 - FUNDAMENTALS OF MACHINE LEARNING**

**Department of Artificial Intelligence and Data Science**

**Rajalakshmi Engineering College, Thandalam**

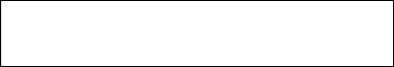
**Nov 2024**

****

**BONAFIDE CERTIFICATE**

**NAME………………………………………………………………………….**

**ACADEMIC YEAR……………SEMESTER………….BRANCH ………**



**UNIVERSITY REGISTER No.**

Certified that this is the bonafide record of work done by the above students in the Mini Project titled " **USING MACHINE LEARNING**" in the subject **AI23331 – FUNDAMENTALS OF MACHINE LEARNING** during the year **2024 - 2025.**

**Signature of Faculty – in – Charge**

**Submitted for the Practical Examination held on -----------------------**

**Internal Examiner External Examine**

**ABSTRACT**

With the increasing prevalence of misinformation and fake news in today's digital world, accurately identifying and classifying fake news has become a critical task. This project leverages machine learning techniques to tackle the issue of fake news dissemination. By analyzing the textual content of news articles, specifically their titles and descriptions, and using advanced natural language processing methods, the project aims to build a robust classification system.

Using a comprehensive dataset of labeled news articles, the project applies preprocessing steps such as text cleaning, tokenization, and TF-IDF vectorization to prepare the data for model training. A Random Forest Classifier is employed due to its effectiveness in handling high-dimensional data and its capability to produce accurate and reliable results.

The trained model is evaluated based on key performance metrics, including accuracy, precision, recall, and F1-score, to ensure its efficacy in real-world scenarios. The results highlight the potential of machine learning in distinguishing fake news from real news, contributing to the broader effort of maintaining information integrity and enhancing public trust.

By providing a reliable tool for fake news detection, this project aims to support the fight against misinformation, promoting a more informed and discerning public. Future work could involve the integration of more advanced models and larger datasets to further improve the accuracy and reliability of the classification system.

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**ABSTRACT** iii

**1 INTRODUCTION 1**

1.1 GENERAL

1.2 NEED FOR THE STUDY

1.4 OVERVIEW OF THE PROJECT

1.3 OBJECTIVES OF THE STUDY

**2 SYSTEM REQUIREMENTS 3**

2.1 HARDWARE REQUIREMENTS

2.2 SOFTWARE REQUIREMENTS

**3 SYSTEM OVERVIEW 4**

3.1 SYSTEM ARCHITECTURE DIAGRAM

3.2 MODULE DESCRIPTION

3.2.1 MODULE 1

3.2.2 MODULE 2

**4 RESULT AND DISCUSSION 6**

**5 CONCLUSION 7**

5.1 CONCLUSION

**APPENDIX 8**

A1.1 SAMPLE CODE

REFERENCES

**CHAPTER 1**

**INTRODUCTION**

**1.1 General  
The rise of social media platforms and news outlets has led to a sharp increase in the spread of fake news. This project aims to develop a machine learning-based approach to classify news articles as fake or real. We use the Random Forest Classifier, which is a robust algorithm, well-suited for this type of classification problem. The dataset contains news article titles, descriptions, and labels indicating whether the news is fake or real.**

**1.2 Need for the Study  
With fake news becoming a pressing issue, automated solutions to detect such news are critical. Traditional methods of detection are time-consuming and subjective. This study proposes a machine learning solution that automates the classification process, providing quick, consistent, and accurate results, which can be used by news agencies, social media platforms, and fact-checking organizations to filter out false information.**

**1.3 Objectives of the Study**

* **Develop a robust machine learning model: Train a classifier to predict whether news is real or fake.**
* **Utilize textual data: Extract features from titles and descriptions to train the model.**
* **Evaluate model performance: Assess the effectiveness using performance metrics such as accuracy, precision, and recall.**
* **Improve classification efficiency: Enhance model performance through fine-tuning and optimization.**

**1.4 Overview of the Project  
The project follows a clear process:**

* **Data Collection: Collect news articles with titles, descriptions, and labels (real or fake).**
* **Data Preprocessing: Handle missing values, tokenize text, and vectorize it using TF-IDF.**
* **Model Building: Train the Random Forest Classifier.**
* **Evaluation: Measure the model's performance using accuracy and other metrics.**
* **Future Work: Explore alternative models such as Support Vector Machines (SVM) or XGBoost for further optimization.**

**CHAPTER 2**

**SYSTEM REQUIREMENTS**

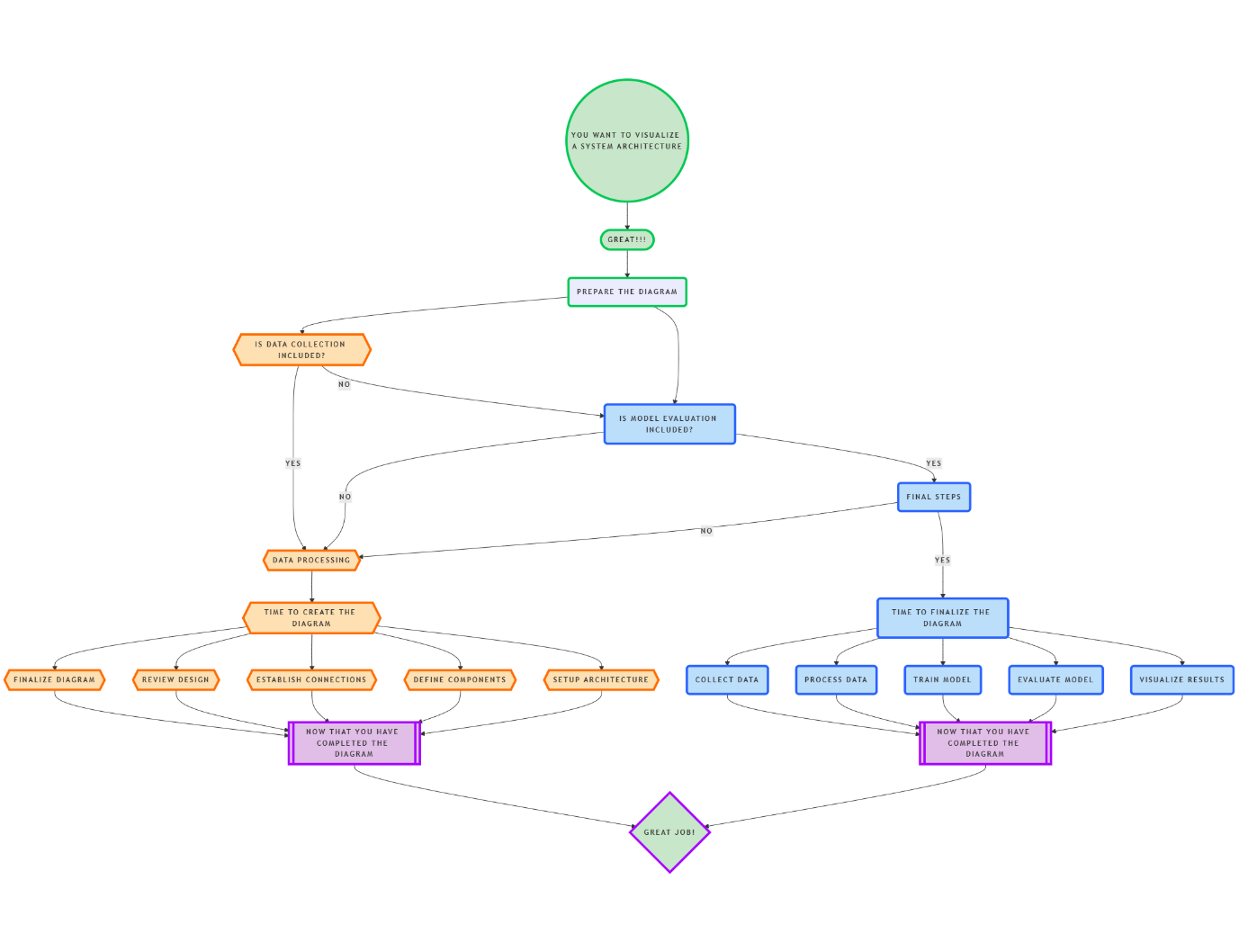
**2.1 Hardware Requirements**

* **Processor: Intel Core i5 or higher (for efficient processing of large datasets)**
* **RAM: 8GB or more**
* **Storage: 256GB SSD or more**
* **Display: HD Monitor (for easy visualization of data and results)**

**2.2 Software Requirements**

* **Operating System: Windows 10 or Linux**
* **Development Environment: Python 3.x, Jupyter Notebook**
* **Libraries:**
  + **Pandas for data manipulation**
  + **NumPy for numerical computations**
  + **Scikit-learn for machine learning models and metrics**
  + **Matplotlib and Seaborn for data visualization**
  + **TF-IDF Vectorizer from sklearn.feature\_extraction.text**

**DIAGRAM:**

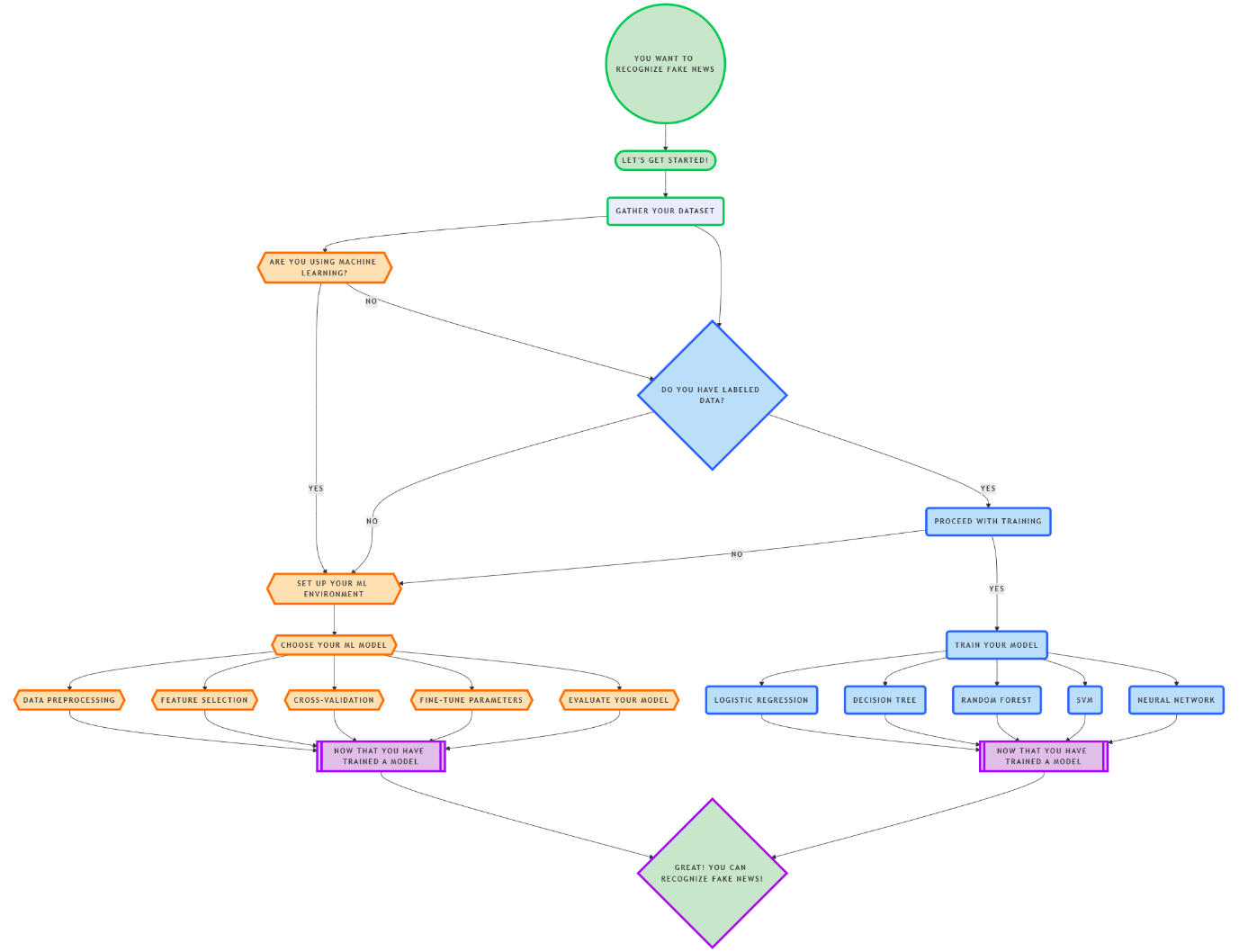
****

**FLOWCHART REPRESENTATION OF MODEL EVALUATION**

**CHAPTER 3**

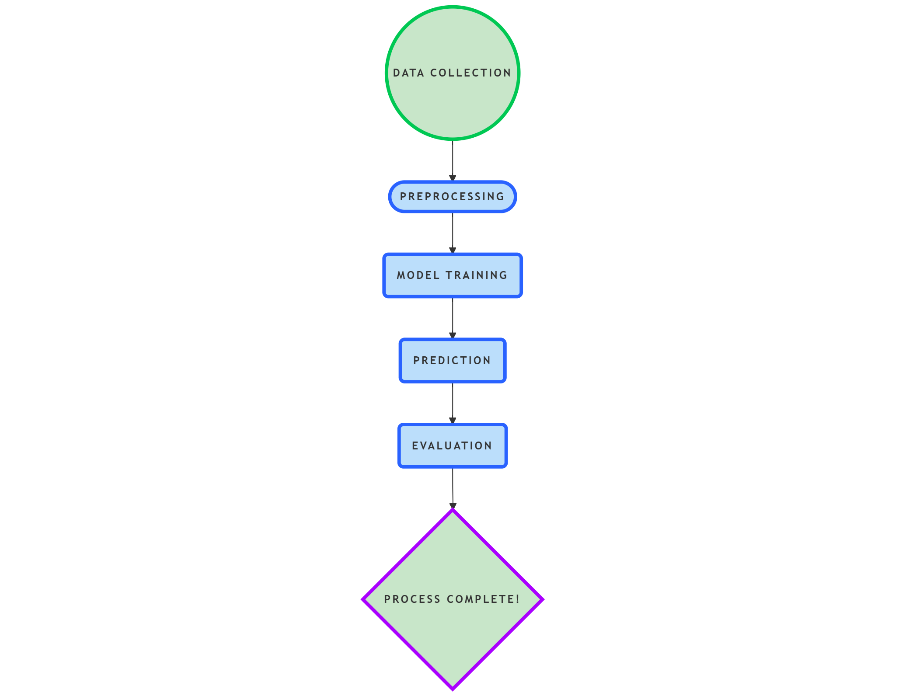
**SYSTEM OVERVIEW**

A diagram illustrating the flow from data collection to preprocessing, model training, evaluation, and final prediction.

****

**3.1 System Architecture Diagram  
The system is designed to process news articles and classify them into "fake" or "real" categories. The architecture consists of:**

* **Data Collection: Raw news data is collected.**
* **Preprocessing: Text data is cleaned, missing values are handled, and features are extracted.**
* **Model Training: The data is passed through a machine learning pipeline to train a model.**
* **Prediction: The trained model classifies new articles.**
* **Evaluation: The model’s performance is assessed using key metrics.**

****

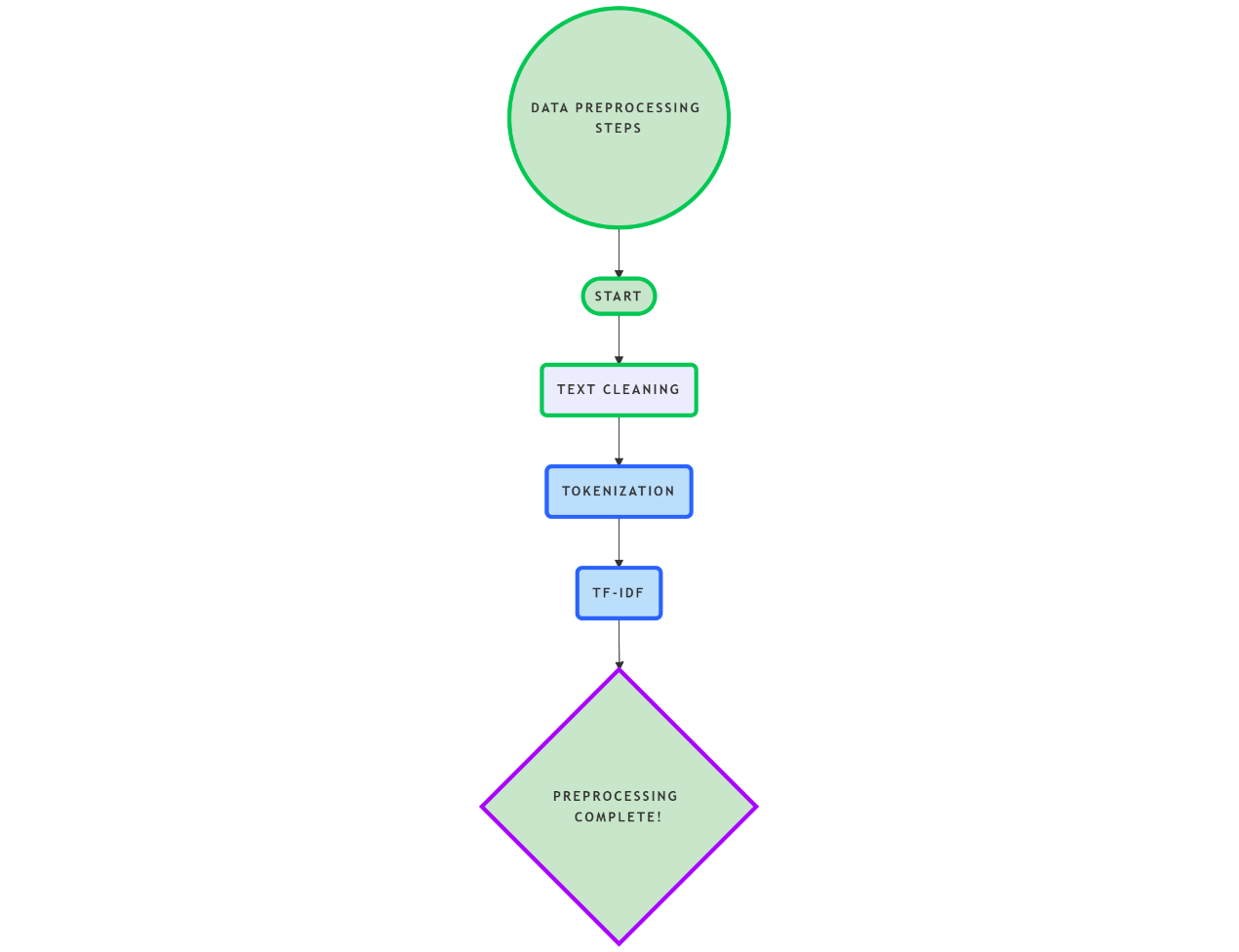
**BLOCK DIAGRAM REPRESENTATION**

**3.2 MODULE DESCRIPTION**

**3.2.1 MODULE 1: DATA PREPROCESSING**

**This module involves preparing the raw text data for machine learning. The steps include:**

* **Handling Missing Data: Rows with missing or incomplete news titles or descriptions are dropped, ensuring no loss of information during training.**
* **Text Tokenization and Normalization: Text is tokenized into words and normalized (e.g., converting all text to lowercase) to avoid redundant features.**
* **TF-IDF Vectorization: The TF-IDF method is used to convert the raw text data into a numerical form. TF-IDF helps to weigh the importance of words in the document relative to their frequency across the corpus.**

****

**FLOWCHART OF DATA PRE-PROCESSING**

**Module 2: Model Training and Evaluation  
Once the data is preprocessed, the machine learning model is trained and evaluated:**

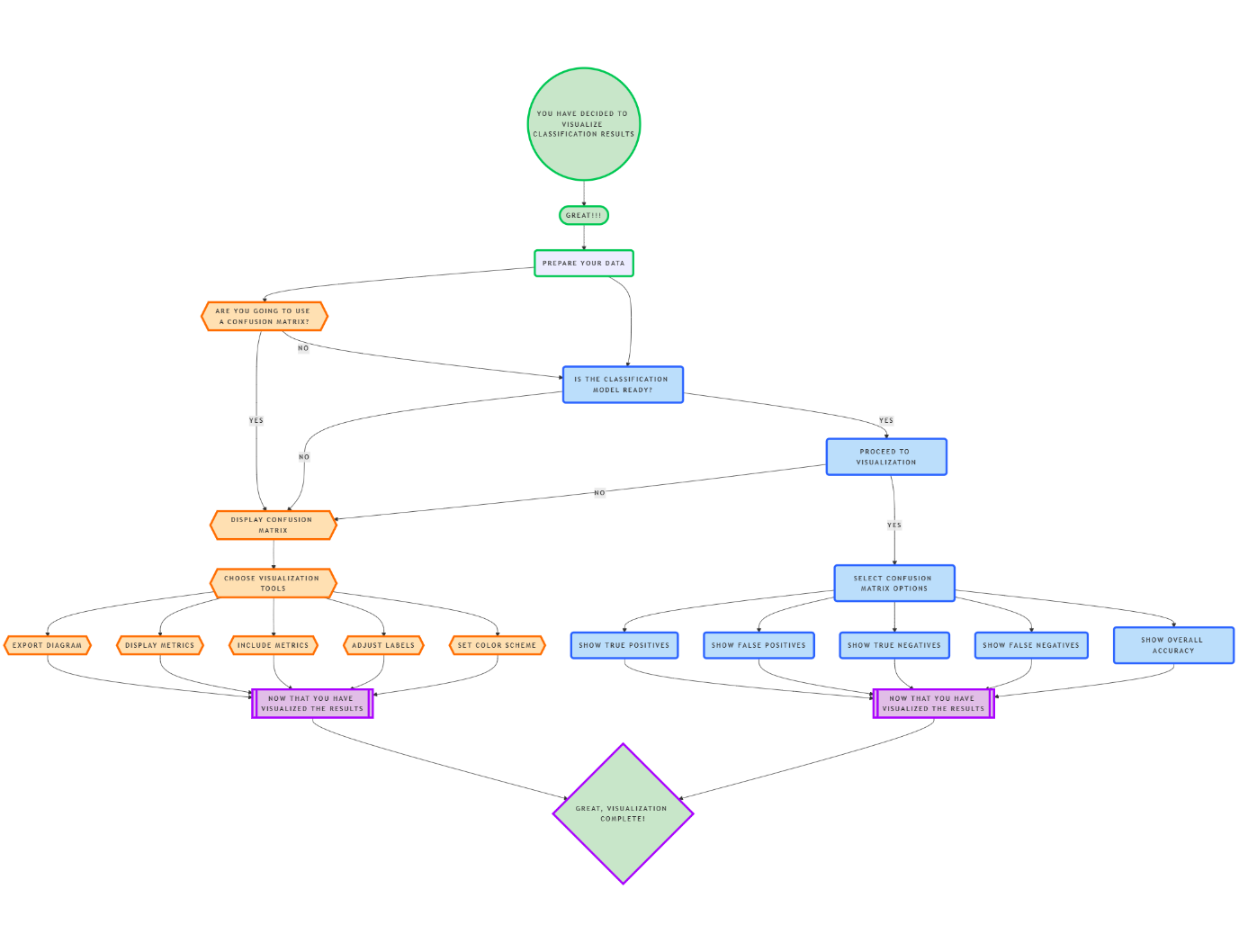
* **Model Training: We use a Random Forest Classifier, which builds a series of decision trees based on different features of the dataset. The ensemble of trees makes the classification process more robust.**
* **Hyperparameter Tuning: Parameters such as the number of estimators (n\_estimators ) and tree depth (max\_depth) are tuned to improve the model's performance.**
* **Model Evaluation: The model's effectiveness is evaluated using:**
  + **Accuracy: Measures the proportion of correctly classified articles.**
  + **Precision and Recall: Focuses on how well the model detects fake news and minimizes false positives or false negatives.**
  + **Confusion Matrix: Shows true positives, true negatives, false positives, and false negatives for a better understanding of the model’s performance.**

**CHAPTER 4**

**RESULT AND DISCUSSION**

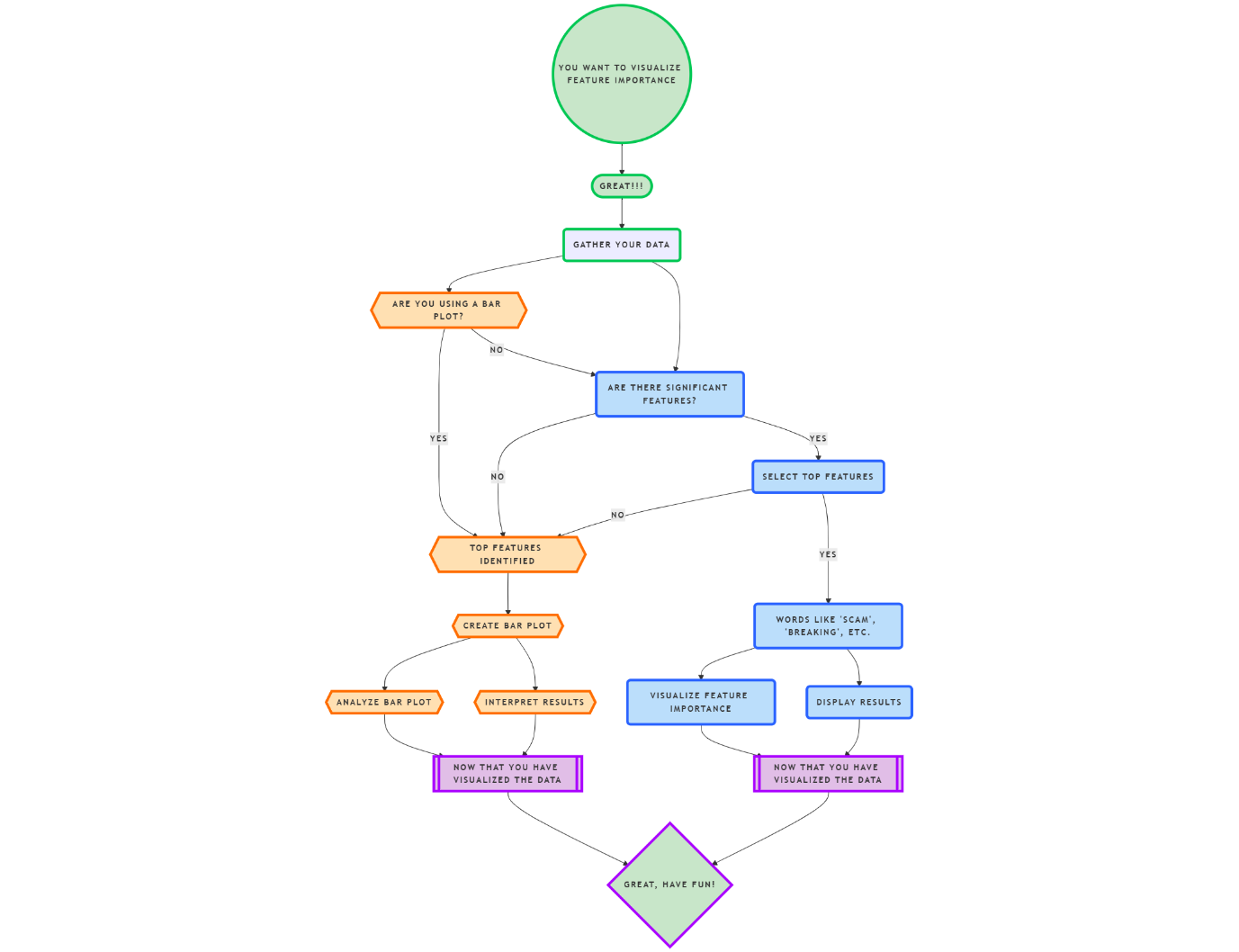
**The evaluation phase of the project provides insight into the effectiveness of the model:**

* **Accuracy: Indicates the overall performance of the classifier.**
* **Precision: The model's ability to correctly identify fake news as fake.**
* **Recall: Measures the model’s ability to detect all instances of fake news.**
* **F1-score: The harmonic mean of precision and recall, providing a balanced measure of model performance.**

****

**FLOWCHART OF CONFESION MATRIX**

**Feature Importance: Using Random Forest, we can analyze which terms in the article titles or descriptions most strongly influence the classification of fake news. This helps in understanding what makes an article fake or real.**

****

**FLOWCHART OF BAR PLOT VISUALIZE THE TOP FEATURES**

**Insights Across Data Segments: By segmenting the dataset into various categories (e.g., keywords, source), we can analyze if the model performs better for certain types of articles, helping stakeholders understand where to focus resources in filtering fake news.**

**CHAPTER 5**

**CONCLUSION**

**The model shows promising results in distinguishing between real and fake news. The Random Forest Classifier, combined with TF-IDF vectorization, provides a reliable and interpretable solution to fake news detection. The evaluation metrics indicate the model's robustness, with good precision and recall values. However, further optimization through techniques like SVM or XGBoost could improve accuracy and reduce misclassification rates. This study highlights the importance of machine learning in combating fake news and can be extended to other text-based classification tasks.**

**APPENDIX**

**A1.1 SAMPLE CODE**

**import pandas as pd**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.feature\_extraction.text import TfidfVectorizer**

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.metrics import accuracy\_score, classification\_report**

**file\_name = "derana\_news.csv"**

**data = pd.read\_csv(file\_name)**

**data['text'] = data['Title'] + " " + data['Description']**

**data.dropna(subset=['text', 'label'], inplace=True)**

**X = data['text']**

**y = data['label']**

**vectorizer = TfidfVectorizer(stop\_words='english', max\_features=5000)**

**X\_vec = vectorizer.fit\_transform(X)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_vec, y, test\_size=0.3, random\_state=42)**

**clf = RandomForestClassifier(n\_estimators=100, random\_state=42)**

**clf.fit(X\_train, y\_train)**

**y\_pred = clf.predict(X\_test)**

**print("Accuracy:", accuracy\_score(y\_test, y\_pred)**

**print("Classification Report:\n", classification\_report(y\_test, y\_pred))**

**References**

**"Text Mining with R: A Tidy Approach" by Julia Silge & David Robinson**

**"Introduction to Machine Learning with Python" by Andreas C. Müller & Sarah Guido**

**"Random Forests" by Leo Breiman**