CROP RECOMMENDATION CHATBOT

MINI PROJECT

Submitted by

YUVASREE R (412422243186)

VARSA S S (412422243169)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

SRI SAIRAM ENGINEERING COLLEGE

(An Autonomous Institution; Affiliated to Anna University, Chennai - 600 025)

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER 2024

SRI SAIRAM ENGINEERING COLLEGE

(An Autonomous Institution; Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report **"CROP RECOMMENDATION CHATBOT"** is the Bonafide work of **YUVASREE R (412522243186), and VARSA S S (412522243169)** who carried out the **MINI PROJECT** under my supervision.

SIGNATURE	**SIGNATURE**
Staff in Charge	HOD
Ms. Jeena A Thankachan	**Dr. Swagata Sarkar**
Assistant Professor	**Head of the Department**

Submitted for project Viva – Voce Examination held on ________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

"Let our advance worrying become advance thinking and planning." -Winston Churchill.

Such a successful personality is our beloved founder Chairman, **Thiru. MJF. Ln. LEO MUTHU**. At first, we express our sincere gratitude to our beloved chairman through prayers, who in the form of a guiding star has spread his wings of external support with immortal blessings.

We express our gratitude to our **CEO Mr. J. Sai Prakash Leomuthu** for having given us spontaneous and whole-hearted encouragement for completing this project.

We express our sincere thanks to our beloved **Principal, Dr. J. Raja** for having given us spontaneous and whole-hearted encouragement for completing this project.

We are indebted to our **Head of the Department Dr. Swagata Sarkar** for her support during the entire course of this project work.

We express our gratitude and sincere thanks to our **subject (20AIPC503, Natural Language Processing and Chatbot) handling staff, Assistant Professor, Ms. Jeena A Thankachan,** for her valuable suggestions and constant encouragement for successful completion of this project.

We thank all the teaching and non-teaching staff members of the **Department of Artificial Intelligence and Data Science** and all others who contributed directly or indirectly for the successful completion of the project.

TABLE OF CONTENTS

ABSTRACT

This project introduces a chatbot-based crop recommendation system that leverages transformer models to analyze environmental and climatic factors, providing accurate crop suggestions. The dataset, sourced from Kaggle, includes vital agricultural parameters: nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall. These features are integrated and tokenized for transformer-based models such as BERT, RoBERTa, SBERT, and DistilBERT to predict optimal crops. Among the models evaluated, RoBERTa emerged as the best-performing model, achieving an impressive accuracy of 97.5%, demonstrating its robustness for crop recommendation tasks.The training process utilized Hugging Face's Trainer API, enabling seamless fine-tuning and evaluation of the models. This facilitated efficient optimization, ensuring that the models could accurately predict suitable crops based on input conditions. The results highlight the potential of transformer models to deliver precise recommendations in agricultural contexts.To enhance accessibility, a user-friendly chatbot interface was developed. This interface allows users to input specific conditions, such as nitrogen levels or temperature, through prompts. The chatbot processes these inputs in real time and provides tailored crop recommendations, empowering farmers and agricultural planners with actionable insights for informed decision-making.This innovative approach showcases the transformative potential of artificial intelligence in agriculture, addressing challenges like efficient crop selection and resource optimization. By integrating advanced transformer models with an intuitive chatbot interface, this project bridges the gap between complex machine learning techniques and practical agricultural applications, paving the way for smarter, data-driven farming practices.

CHAPTER 1
INTRODUCTION

1.1. OBJECTIVE

Agriculture, a critical component of global food security and economic stability, is increasingly challenged by climate variability, soil degradation, and evolving environmental conditions. Farmers and agricultural stakeholders require precise, data-driven tools to optimize resource use, improve crop yields, and ensure sustainable farming practices. However, traditional crop recommendation systems often fail to effectively analyze diverse environmental parameters or adapt to dynamic data patterns, limiting their utility in addressing modern agricultural challenges. This project aims to develop a robust crop recommendation system utilizing advanced transformer models, capable of analyzing complex relationships between critical agricultural parameters—such as soil nutrients (nitrogen, phosphorus, potassium), temperature, humidity, pH, and rainfall—to suggest optimal crops. By integrating these models with an interactive chatbot interface, the system seeks to provide real-time, science-backed crop recommendations, empowering users with actionable insights for efficient and sustainable agricultural decision-making

1.2. OUTLINE OF THE REPORT

The project, titled Crop Recommendation Chatbot Using Transformers, addresses the challenge of inefficient crop selection faced by farmers due to the lack of access to accurate, data-driven recommendations. By leveraging advanced transformer models, this project aims to develop a chatbot that provides real-time crop suggestions based on environmental and climatic conditions. The methodology begins with data collection and preprocessing, utilizing a Kaggle dataset with key agricultural parameters such as nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall. The dataset is prepared for transformer models by combining features into a single column and encoding crop labels for compatibility. Pretrained tokenizers, such as BertTokenizer, are employed to tokenize the input data.

Multiple transformer models, including BERT, RoBERTa, SBERT, and DistilBERT, are evaluated to determine their performance in crop prediction. Using Hugging Face's Trainer API, these models are fine-tuned, with RoBERTa achieving the highest accuracy at 97.5%. The chatbot interface is designed to be interactive and user-friendly, allowing farmers to input specific environmental conditions and receive accurate crop recommendations.

1.3. SCOPE OF THE PROJECT

The scope of this project is to develop an NLP-based chatbot for recommending the most suitable crops based on environmental conditions such as temperature, humidity, pH, and rainfall. The chatbot will use a machine learning model trained on a preprocessed dataset to provide accurate and efficient recommendations, making it a valuable tool for farmers, agricultural advisors, and researchers. The system will operate in a user-friendly conversational interface, ensuring accessibility for individuals with varying levels of technical expertise. It is designed to assist in optimizing crop selection, thereby improving agricultural productivity and reducing dependency on expert consultations.

In the future, the system can be enhanced to support multiple languages, provide fertilizer and pesticide recommendations, integrate real-time environmental data through IoT devices, and factor in market trends for crop selection. The chatbot is primarily intended for farmers seeking to make informed decisions about crop cultivation and for agricultural stakeholders aiming to improve advisory services.

CHAPTER 2

SDG JUSTIFICATION

SDG 2: Zero Hunger

The chatbot plays a vital role in combating hunger by offering tailored, data-driven crop recommendations that enable farmers to grow the most suitable crops for their environmental and climatic conditions. This precision ensures optimized agricultural productivity, leading to increased food availability and stability in the supply chain. By reducing the risks of crop failure through informed decision-making, the tool directly supports the goal of eliminating hunger and malnutrition worldwide.

SDG 8: Decent Work and Economic Growth

The chatbot empowers farmers with scientific insights, enabling them to make informed choices that enhance crop yield and quality. By improving agricultural efficiency, it boosts income levels for farmers, especially in rural areas, fostering economic growth and reducing poverty. Moreover, by reducing uncertainty and the risks of crop failure, the tool contributes to a more stable agricultural economy, promoting dignified livelihoods and stimulating sustainable growth in the agricultural sector.

SDG 12: Responsible Consumption and Production

Through careful analysis of environmental factors such as soil nutrients (nitrogen, phosphorus, potassium), temperature, and rainfall, the system ensures that farmers grow crops in alignment with the available resources. This minimizes wastage of inputs like fertilizers and water, reducing the environmental footprint of farming. Additionally, by promoting sustainable practices, the tool helps maintain soil health and balance agricultural demands with ecological sustainability, encouraging a more responsible production process.

CHAPTER 3

PROPOSED SOLUTION

3.1. SOURCE CODE:

```
pip install transformers torch pandas scikit-learn
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from transformers import RobertaTokenizer

# Load dataset
data = pd.read_csv('corpus2 (1).csv')

# Encode labels
label_encoder = LabelEncoder()
data['label'] = label_encoder.fit_transform(data['label'])

# Split dataset into train and test
train_texts, test_texts, train_labels, test_labels =
train_test_split(data['text'], data['label'], test_size=0.2,
random_state=42)

# Load pre-trained RoBERTa tokenizer
tokenizer =
RobertaTokenizer.from_pretrained('roberta-base')

# Tokenize text
train_encodings = tokenizer(list(train_texts),
truncation=True, padding=True, max_length=128)
test_encodings = tokenizer(list(test_texts), truncation=True,
padding=True, max_length=128)

import torch
from torch.utils.data import Dataset, DataLoader
from transformers import
RobertaForSequenceClassification, Trainer,
TrainingArguments

class CropDataset(Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
```

```
        self.labels = labels

    def __getitem__(self, idx):
        item = {key: torch.tensor(val[idx]) for key, val in
self.encodings.items()}
        # Ensure labels are accessed correctly as a list
        item['labels'] = torch.tensor(self.labels.tolist()[idx])  #
Change this line
        return item

    def __len__(self):
        return len(self.labels)

# After train-test split, reset indices to avoid potential
misalignment
train_labels = train_labels.reset_index(drop=True)
test_labels = test_labels.reset_index(drop=True)

train_dataset = CropDataset(train_encodings, train_labels)
test_dataset = CropDataset(test_encodings, test_labels)

# ... (rest of your code)

# Load pre-trained RoBERTa model
model =
RobertaForSequenceClassification.from_pretrained('roberta
-base', num_labels=len(label_encoder.classes_))

# Define training arguments
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=4,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
    evaluation_strategy="steps"
)

# Trainer
trainer = Trainer(
```

```
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset
)

# Train model
trainer.train()
# After training, save the model to a directory
model_save_path = "/content/trained_roberta_model"  #
Directory where the model will be saved
model.save_pretrained(model_save_path)
tokenizer.save_pretrained(model_save_path)

import numpy as np
from sklearn.metrics import accuracy_score,
precision_recall_fscore_support

# Perform evaluation
predictions = trainer.predict(test_dataset)

# Get the predicted labels
preds = np.argmax(predictions.predictions, axis=1)

# Get the true labels
labels = predictions.label_ids

# Calculate accuracy
accuracy = accuracy_score(labels, preds)

# Calculate precision, recall, and F1-score
precision, recall, f1, _ =
precision_recall_fscore_support(labels, preds,
average='weighted')

# Print metrics
print(f'Accuracy: {accuracy:.4f}')
print(f'Precision: {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1-Score: {f1:.4f}')

import torch
```

```
# Function to predict crop type based on input features
def predict_crop(model, tokenizer, features):
    inputs = tokenizer(features, return_tensors="pt",
truncation=True, padding=True, max_length=128)
    inputs = {k: v.to(model.device) for k, v in inputs.items()}
    outputs = model(**inputs)
    prediction = torch.argmax(outputs.logits, dim=1)
    return
label_encoder.inverse_transform(prediction.cpu().numpy())
[0]

# Chatbot interface
def chatbot():
    print("Welcome to the Agriculture Chatbot!")
    print("I will suggest the optimal crop to grow based on
environmental conditions.")
    print("Type 'quit' at any time to exit the chatbot.")

    while True:
        # Collecting inputs with detailed prompts
        n = input("Please provide nitrogen (N) level: ")
        if n.lower() == 'quit':
            print("Exiting the chatbot. Goodbye!")
            break

        p = input("Please provide phosphorus (P) level: ")
        if p.lower() == 'quit':
            print("Exiting the chatbot. Goodbye!")
            break

        k = input("Please provide potassium (K) level: ")
        if k.lower() == 'quit':
            print("Exiting the chatbot. Goodbye!")
            break

    temperature = input("Please provide temperature (°C): ")
        if temperature.lower() == 'quit':
            print("Exiting the chatbot. Goodbye!")
            break

        humidity = input("Please provide humidity (%): ")
        if humidity.lower() == 'quit':
            print("Exiting the chatbot. Goodbye!")
```

```
            break

        # Combine all inputs into a single feature string for
prediction
        features = f"{n}, {p}, {k}, {temperature},
{humidity}"

        # Predict and display the recommended crop
        crop = predict_crop(model, tokenizer, features)
        print(f"The optimal crop to be grown is: {crop}")

# Run chatbot
chatbot()
```

CHAPTER 4

REQUIREMENT SPECIFICATION

Functional Requirements

1. **User Input**:
    - The chatbot shall accept numerical environmental parameters:N,P,K,Temperature,Humidity,Rainfall
    - The chatbot shall support inputs in textual or numerical form (e.g., “The temperature is 30°C”).
2. **Crop Recommendation**:
    - The system shall analyze the input parameters and recommend a crop from a predefined list.
    - The system shall display a single most-suited crop recommendation or multiple options ranked by suitability.
3. **Chatbot Features**:
    - The chatbot shall greet users with:

        “Welcome to the Agriculture Chatbot!"

        "I will suggest the optimal crop to grow based on environmental conditions."

        "Type 'quit' at any time to exit the chatbot."

Non-Functional Requirements

1. **Performance**:
    - The system shall respond to user queries in under 2 seconds.
    - The model shall process at least 200 user queries per minute.
2. **Accuracy**:
    - The crop recommendation accuracy shall be at least **97%** based on the dataset used for training.
3. **Usability**:
    - The chatbot interface shall be user-friendly and intuitive, requiring no prior technical expertise.
    - The chatbot shall use simple, conversational language suitable for farmer

CHAPTER 5

IMPLEMENTATION

5.1.IMPLEMENTATION TECHNOLOGY

1. Data Collection and Preprocessing

The dataset used for crop recommendation was sourced from Kaggle, specifically designed to include key agricultural parameters that influence crop growth. It consists of **2200 rows** and **8 columns** with no missing values, ensuring a clean and complete dataset.

- **Features (Inputs):**
  - **Nitrogen (N):** A crucial nutrient for plant growth.
  - **Phosphorus (P):** Vital for root development and flowering.
  - **Potassium (K):** Important for overall plant health.
  - **Temperature, Humidity, pH, and Rainfall:** Essential environmental factors that affect crop growth.
- **Label (Output):** Specifies the crop type corresponding to the conditions.

Preprocessing Steps:

- **Feature Combination:** The first 7 numerical features were combined into a single column labeled **"text"** for compatibility with transformer models, which process textual data.
- **Label Encoding:** The categorical crop labels were converted into numerical values using **LabelEncoder**, allowing the models to handle them during training.
- **Dataset Splitting:** The dataset was divided into an **80% training set** and a **20% test set** to evaluate the model's performance.

2. Text Tokenization

Transformers require text input in a tokenized format. A pretrained tokenizer, such as **BertTokenizer**, was used to convert the "text" data into tokens that the models could process.

Steps in Tokenization:

- **Subword Tokenization:** The tokenizer splits the input "text" into subword units, capturing meaningful patterns.
- **Adding Special Tokens:** Special tokens like `[CLS]` (start of sequence) and `[SEP]` (end of sequence) are added to signify the boundaries of input.

- **Padding and Truncation:** The sequences are adjusted to a fixed length of **128 tokens** by padding shorter sequences and truncating longer ones.
- **Output Format:** The tokenizer outputs a dictionary containing token IDs, attention masks, and segment IDs, which are then fed into the models.

Both training and testing datasets underwent tokenization to ensure consistency in data processing.

3. Model Setup

Multiple transformer models were explored for the task, each offering unique strengths:

1. **BERT (Bidirectional Encoder Representations from Transformers):**
    - **Overview:** BERT uses a bidirectional attention mechanism to capture context in both directions, making it highly effective for tasks requiring an understanding of relationships between data points.
2. **RoBERTa (Robustly Optimized BERT Pretraining Approach):**
    - **Overview:** RoBERTa is an improved variant of BERT, trained on more data with larger mini-batches and dynamic masking strategies. It delivers superior results in tasks requiring deep contextual understanding.
3. **SBERT (Sentence-BERT):**
    - **Overview:** SBERT modifies BERT to produce sentence embeddings, making it ideal for tasks requiring similarity measures.
    - **Special Feature:** Instead of strict classification, SBERT outputs the five most similar crop recommendations based on the user's query, enabling flexibility in suggestions.
4. **DistilBERT (Distilled BERT):**
    - **Overview:** A compact and efficient version of BERT, designed for faster training and inference while retaining 97% of BERT's language understanding capabilities.

4. Model Training

Training Process:

- The **Trainer** class from Hugging Face's Transformers library was employed to streamline the training process.
- The **training dataset** was fed into the model for fine-tuning, while the **test dataset** was used to evaluate performance.

- **Steps Involved:**
    - Forward propagation: Inputs are passed through the model to generate predictions.
    - Loss calculation: The difference between predicted and actual values is measured.
    - Backpropagation: The model adjusts its parameters to minimize the loss function.
- Metrics such as **accuracy, precision, recall**, and **F1-score** were used to measure model performance.

5. Prediction

Once trained, the models were used for predicting crop types based on the input conditions provided by users.

Prediction Process:

- User inputs (e.g., nitrogen, temperature) are converted into a text format and tokenized.
- The tokenized input is passed through the model, which outputs **logits** (raw scores for each crop label).
- The logits are processed to determine the most likely crop type.
- The numerical label is mapped back to the original crop name using the **inverse transform** method of the LabelEncoder.

6. Chatbot Interface

A chatbot interface was developed to make crop recommendations accessible to users.

Features of the Chatbot:

- **User Input:** Prompts the user to input environmental parameters (e.g., N, P, K, temperature, etc.) one at a time.
- **Real-Time Interaction:** The chatbot analyzes the inputs and predicts the optimal crop type.
- **User-Friendly Design:**
    - Welcomes the user with a greeting and instructions.
    - Allows the user to exit anytime by typing **'quit'**.
- **Output:** Provides precise crop recommendations in a conversational manner, ensuring accessibility even for users with minimal technical knowledge.

This design ensures the tool is intuitive, flexible, and effective for assisting farmers with data-driven crop planning.

CHAPTER 6

RESULT AND CONCLUSION

6.1 RESULTS

The study evaluated the performance of transformer models—BERT, RoBERTa, SBERT, and DistilBERT—on a crop recommendation system. These models were tested against a dataset containing essential agricultural parameters, such as soil nutrients, temperature, humidity, pH, and rainfall. Among the models, RoBERTa demonstrated the highest accuracy of 97.5%, showcasing its exceptional capability to predict suitable crops based on environmental conditions. This result highlights the effectiveness of transformer models in capturing complex relationships between input features and crop suitability.

6.2. CONCLUSION

This experiment serves as a case study demonstrating the efficiency of transformer models in agricultural applications. The models, particularly RoBERTa, achieved significant success in predicting optimal crops based on diverse environmental factors. The approach underscores the scalability and intelligence of transformer-based solutions in addressing contemporary farming challenges. By enabling precise and data-driven agricultural decision-making, this study highlights the transformative potential of AI-driven systems in revolutionizing sustainable farming practices.

CHAPTER 7

REFERENCES

1. **Neelam Chandolikar, Chirag Dale, Tejas Koli, Mayank Singh, Tarussh Narkhede** (2022). Agriculture Assistant Chatbot Using Artificial Neural Network.

2. **Mihir Momaya, Anjnya Khanna, Jessica Sadavarte, Manoj Sankhe** (2021). Krushi – The Farmer Chatbot.

3. **Praveen Kumar Maduri, Saurabh Anand, Preeti Dhiman, Siddharth Pratap Singh, Manas Raj Shukla** (2021). Farmers Agriculture Assistance Chatbot.

4. **Prashant Y. Niranjan, Vijay S. Rajpurohit, Rasika Malgi** (2019). A Survey on Chat-Bot System for Agriculture Domain.

5. **Susheel Gounder, Mahaleppa Patil, Vishal Rokade, Nikita More** (2021). Agrobot: An Agricultural Advancement to Enable Smart Farm Services Using NLP.

6. **Pitchayagan Temniranrat, Kantip Kiratiratanapruk, Apichon Kitvimonrat, Wasin Sinthupinyo, Sujin Patarapuwadol** (2021). A System for Automatic Rice Disease Detection from Rice Paddy Images Serviced via a Chatbot.

7. **Patience U. Usip, Edward N. Udo, Daniel E. Asuquo, Otobong R. James** (2022). A Machine Learning-Based Mobile Chatbot for Crop Farmers.

8. **Abbott Po Shun Chen** (2020). B2B Marketing Crafts Intelligence Commerce: How a Chatbot Is Designed for the Taiwan Agriculture Service.

9. **Jais Binoy, Annie Thomas, Akhilesh Muralidharan** (2023). A Dialogflow Chatbot for Farmer Producer Organizations: Enhancing Agricultural Decision-Making.

10. **M. Amit Kumar, Chollangi Lakshmi, Mote Shiva Kumar, Dudoori Pavan Kumar, Kothapally Jadidya** (2023). Agriculture Helper Chatbot AI.