FERTILIZER RECOMMENDATION SYSTEM USING MACHINE LEARNING

PROJECT REPORT ON

Program : AIML

Submitted to: Besant Technologies

Submitted by: MANOJH R

Student Id : 25BFT421332

Co-ordinator Name: VALLI

Date of Submission: 29-07-2025

Acknowledgement

I would like to express my heartfelt gratitude to **Besant Technologies** for providing me with the opportunity to enroll in their online course on **Artificial Intelligence and Machine Learning (AIML)**. Over the span of one month, I gained valuable knowledge and hands-on experience which enabled me to successfully complete this project based on the concepts I learned.

A special thanks to my **coordinator and mentor**, whose continuous guidance, encouragement, and support were instrumental throughout my learning journey and the development of this project.

Table of Contents

1. Cover Page

2. Acknowledgement

3. Table of Contents

4. Concepts Used in Project

5. Source Code

6. Description of Source Code

7. Output

8. Conclusion

9. Bibliography

Concepts Used in Project

- Supervised Learning (Classification)
- Label Encoding of Categorical Features
- Random Forest Classifier
- Data preprocessing using pandas and numpy

Source Code

Fertilizer.py

```
import joblib
import numpy as np

model = joblib.load("fertilizer_model.pkl")
encoders = joblib.load("label_encoders.pkl")
soil_encoder = encoders['Soil Type']
crop_encoder = encoders['Crop Type']
fertilizer_encoder = encoders['Fertilizer Name']

soil_options = list(soil_encoder.classes_)
crop_options = list(crop_encoder.classes_)

temp = int(input(" enter temperature? "))
humidity = int(input("enter humidity level "))
moisture = int(input("enter soil moisture level "))

print("\n Choose soil type:")
print(", ".join(soil_options))
soil_type = input("Soil Type: ").strip()
while soil_type not in soil_options:
    print("Case sensitive alert.")
    soil_type = input("Soil Type: ").strip()

print("\n Choose crop type:")
print(", ".join(crop_options))
crop_type = input("Crop Type: ").strip()
```

```
while crop_type not in crop_options:
    print("only enter the crop type in the options with exact cases")
    crop_type = input("Crop Type: ").strip()

nitrogen = int(input("enter nitrogen "))
potassium = int(input("enter potassium level: "))
phosphorous = int(input("enter phosphorous level: "))

soil_val = soil_encoder.transform([soil_type])[0]
crop_val = crop_encoder.transform([crop_type])[0]
data = np.array([[temp, humidity, moisture, soil_val, crop_val, nitrogen, potassium,
phosphorous]])

#prediction
predicted = model.predict(data)[0]
fertilizer = fertilizer_encoder.inverse_transform([predicted])[0]

print("\n recommanded fertilizer for your field", fertilizer)
```

Description of Source Code

The above python script loads the trained model and category encoders. It asks the user to enter field parameters like temperature, humidity, moisture, soil and crop type, and nutrient levels. It checks only valid soil data and crop types data are entered. All inputs are processed and encoded, then feed to the model to predict the fertilizer. Finally, the recommended fertilizer is shown to the user.

Output

Fertilizer.py output:

Conclusion

This project shows how machine learning can help users choose the right fertilizer based on real-time conditions. It uses a simple interface and a smart model to make decisions easier. This can be a foundation for building a web or mobile app for agriculture support.

Bibliography

- https://scikit-learn.org/
- https://pandas.pydata.org/
- https://numpy.org/
- https://www.kaggle.com/datasets/shankarpriya2913/crop-and-soil-dataset