SMART FARMING

In this project, I leveraged the power of **BERT** (Bidirectional Encoder Representations from Transformers) to enhance **crop prediction** using a dataset that contains critical agricultural parameters. The dataset consists of features such as nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, pH levels, and rainfall, to predict the type of crop suitable for specific environmental conditions.

Dataset Used

The dataset contains the following columns:

- **N**: Nitrogen content in the soil.
- **P**: Phosphorus content in the soil.
- **K**: Potassium content in the soil.
- **Temperature**: The temperature of the environment.
- **Humidity**: The humidity level.
- **pH**: The pH level of the soil.
- **Rainfall**: The amount of rainfall.
- **Label**: The type of crop (e.g., rice, wheat) suitable for the given conditions.

N	P	K	temperatu	humidity	ph	rainfall	label
90	42	43	20.87974	82.00274	6.502985	202.9355	rice
85	58	41	21.77046	80.31964	7.038096	226.6555	rice
60	55	44	23.00446	82.32076	7.840207	263.9642	rice
74	35	40	26.4911	80.15836	6.980401	242.864	rice
78	42	42	20.13017	81.60487	7.628473	262.7173	rice
69	37	42	23.05805	83.37012	7.073454	251.055	rice
69	55	38	22.70884	82.63941	5.700806	271.3249	rice
94	53	40	20.27774	82.89409	5.718627	241.9742	rice
89	54	38	24.51588	83.53522	6.685346	230.4462	rice
68	58	38	23.22397	83.03323	6.336254	221.2092	rice
91	53	40	26.52724	81.41754	5.386168	264.6149	rice
90	46	42	23.97898	81.45062	7.502834	250.0832	rice
78	58	44	26.8008	80.88685	5.108682	284.4365	rice
93	56	36	24.01498	82.05687	6.984354	185.2773	rice
94	50	37	25.66585	80.66385	6.94802	209.587	rice
60	48	39	24.28209	80.30026	7.042299	231.0863	rice
85	38	41	21.58712	82.78837	6.249051	276.6552	rice
91	35	39	23.79392	80.41818	6.97086	206.2612	rice

Fine-Tuning BERT

To **fine-tune BERT** on structured tabular data, the process involved adapting the model to handle non-textual data like the given environmental features. The first step was transforming the tabular data into a format that could be understood by the language model. Each feature, such as nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, pH, and rainfall, was treated as a **"token"** and **embedded as a vector**. This transformation allowed the model to interpret these features similarly to how it would process words in a text-based task. By feeding the processed data into the **BERT** architecture, we were able to fine-tune it to classify crops effectively based on the given environmental parameters. The model learned to detect patterns and relationships among the features to make accurate predictions.

Results

The BERT model demonstrated the following metrics after training:

Step	Training Loss	Validation Loss	Accuracy	F1
200	3.091200	2.476535	0.409091	0.336285
400	1.124000	0.302536	0.968182	0.967552
600	0.228500	0.092298	0.988636	0.988542

- **Accuracy Score** measures the ratio of correctly predicted instances to the total instances, providing a straightforward assessment of a model's performance. However, it can be misleading in imbalanced datasets, where one class may dominate.
-
- **F1 Score** combines precision and recall into a single metric, making it particularly useful for evaluating models on imbalanced datasets. It considers both the false positives and false negatives, providing a more nuanced view of a model's ability to classify positive instances correctly.

These results suggest that BERT can effectively learn from agricultural features to make accurate predictions, thereby supporting decision-making in farming.

Predictions

Crop Prediction

The model was able to predict the most suitable crops based on the provided parameters:

- For data row: N=90, P=42, K=43, Temperature=20.87°C, Humidity=82%, pH=6.50, and Rainfall=202mm, the predicted crop was **Rice**.

```
[31]: print(predict("The nitrogen content is 90, phosphorus is 42, and potassium is 43,ph is 6.50 with a temperature of 29.86896°C."))
rice
```

The BERT model consistently made predictions in line with known agronomic principles, proving its utility for smart farming solutions.

Crop Description

This code generates a paragraph about a predicted crop using a pre-trained language model. It uses the Microsoft's phi-3.5 model for providing unique characteristics and features of the predicted crop. It constructs an input prompt, tokenizes it into a format suitable for the model, and then generates a response using beam search to improve output quality. The generated tokens are decoded back into text and printed, providing a detailed description of the crop's characteristics, uses, and importance.

```
input_text = f"The predicted crop is {predicted_crop}. Write a paragraph about its unique characteristics, uses, and its importance in agriculture."

input_ids = tokenizer.encode(input_text, return_tensors="pt")
outputs = model.generate(input_ids=input_ids, max_length=200, num_beams=4, early_stopping=True)

generated_text = tokenizer.decode(outputs[0], skip_special_tokens=True)
print(generated_text)
```

```
The predicted crop is rice. Write a paragraph about its unique characteristics, uses, and its importance in agriculture.

Rice, scientifically known as Oryza sativa, is a staple food for more than half of the world's population, making it one of the most important crops in agriculture. It is a versatile cereal grain that thrives in a variety of climates, from tropical to subtropical regions. Rice plants have unique characteristics, such as their ability to grow in flooded fields, known as paddy fields, which helps control weeds and pests naturally. The grains are rich in carbohydrates, providing a significant source of energy, and contain essential nutrients like vitamins, minerals, and dietary fiber.

Rice is used in a wide range of culinary dishes across different cultures, from sushi in Japan
```

Refer to the following notebook for the implementation of the code:

https://www.kaggle.com/code/jainhemang/smart-farming