**Smart Planter**

**Week 1 Update**

**Dataset Generation:**

* To create a balanced dataset with an equal number of 'Healthy', 'Moderate', and 'Unhealthy' instances, I create a function **generate\_data** to first generate a third of the dataset with each label, and then shuffle the dataset to mix it.
* **Explanation**:
  + I divide the total number of data points by 3 to get an equal number of data points for each category ('Healthy', 'Moderate', and 'Unhealthy').
  + I then generate data for each category separately, using different ranges of values for the features to simulate the different conditions for each category.
  + After generating the data, I create a DataFrame and shuffle it using df.sample(frac=1).reset\_index(drop=True) to mix the data points from the different categories together.
* Here is the result, showing top 10 rows of dataset:

A screenshot of a computer screen

Description automatically generated

**Build Neural Networks to Predict Plant Health:**

**Step 1: Data Preprocessing**

Before feeding the data into a neural network, it's essential to preprocess it. This includes encoding categorical labels and normalizing the feature values.

**Step 2: Splitting Dataset**

Split the data into training and testing sets i.e 20 percent testing data and 80 percent training dataset.

**Step 3: Building neural networks:**

Initially I used 3 dense layers with one output layer having SoftMax activation function to classify the categories correctly.

Then compile which means build the model with Adam as optimizer, sparse\_categorical\_crossentropy as loss function and metrics is accuracy.

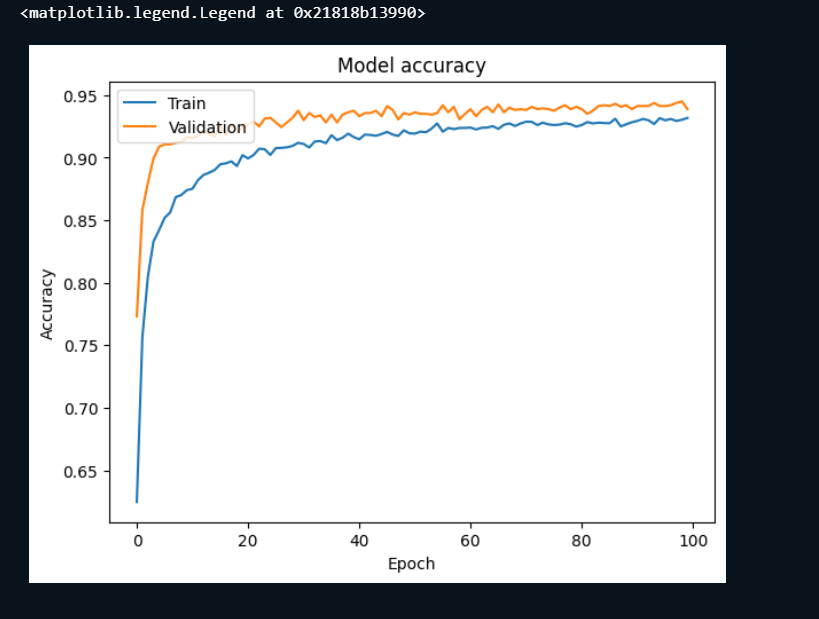
**Step 4: Training Model:**

Now its time to train the model, so I feed the training and testing data to the model with 100 epochs, 32 batch size and 20 percent validation split.

It takes a little time to train the model on the given dataset.

**Step 5: Visualize the model accuracy and loss:**

Here is the visualization.



The above graph shows the accuracy is not stable yet, which means it required further tunning to make the model further accurate. Later, I will do that.

**Step 6: Evaluate Model:**

Now its time to test the model on the testing data to make sure the model will predict the unseen data accurately or not.

After evaluating the model on test data, I got below result, which shows a good stats but still need to be improve



Step 7: Making Predictions:

I feed the model with some unseen data on which the model is not train and get the result which shows

A screen shot of a computer code

Description automatically generated

Based on given data points the model predicts the plant is healthy.

**Conclusion:**

Until now the model is working fine. It requires further tunning.

I will also use other prediction models and will compare the results.