# Task 01

**Submission Date: 05/04/2024**

**Group Members:**


- SATHEESHA FERNANDO
- NISAL SARANGA LIYANAGE
- JAYANI ADIKARI
- TARINYA PERERA

**Report content:**

- Approach

1.1 Programming Language

1.2 Libraries

1.3 Machine Learning Model

- Challenges
- Insights on Model Evaluation
- Suggestions for Better Performance

# Chapter 01 | Approach

### 1.1 Programming Language

**Python was used as the programming language throughout the entire process. The following are the reasons why we utilized python to achieve this:**

- **Python has a rich collection of various Machine Learning libraries and frameworks.**
- **Python is easy to learn and use.**
- **Python offers strong community support and clear documentation.**

### 1.2 **Libraries**:

**NumPy**: Utilized to convert decision function scores to prediction probabilities and to get the indexes with the top 3 highest probabilities.

```
# Convert decision function scores to probabilities using softmax function
def softmax(x):
    exp_x = np.exp(x - np.max(x))
    return exp_x / exp_x.sum(axis=1, keepdims=True)

predicted_probabilities = softmax(decision_scores)

# Get the top 3 predictions for each sample
top_three_indices = np.argsort(predicted_probabilities, axis=1)[:, -3:]
```

**Pandas**: Used to convert the csv file (dataset) to a dataframe and perform data manipulation on them.

```
df = pd.read_csv('Crop_Dataset.csv')

print(df.shape)  # added this to check the size of the dataset.

df = df.dropna(axis='index', how='any')  # any row which has a null value will be dropped out

label_encoder = LabelEncoder()
df['Label_Encoded'] = label_encoder.fit_transform(df['Label'])

scaler = StandardScaler()
features_to_scale = ["N", "P", "K", "temperature", "humidity", "ph", "rainfall"]
df[features_to_scale] = scaler.fit_transform(df[features_to_scale])

df.to_csv('new_dataset.csv')
X = df[features_to_scale]
y = df['Label_Encoded']
```

**scikit-learn (sklearn):** Employed for machine learning tasks such as preprocessing, model selection, and evaluation.

```
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn import metrics
```

**joblib**: Used for saving and loading the model and label encoder.

```
joblib.dump(model, 'model.joblib')
joblib.dump(label_encoder, 'label_encoder.joblib')  # Save the LabelEncoder
```

1.3 **Machine Learning Model**:

**Support Vector Machine (SVM) was used as the Machine Learning model for this classification part. The following are the reasons for choosing SVM.**

- SVMs excel in managing datasets with numerous features.
- SVMs employ various kernel functions to capture complex relationships between features and labels.
- Although most commonly SVMs are used as binary classifiers, SVMs can also be used to handle multi-class classification tasks, making them adaptable for categorizing crops into multiple classes based on environmental conditions.

# Chapter 02 | Challenges

- Deciding which particular features to be selected for the model training. We came to a conclusion that combined features like Total_Nutritients, Temperature_Humidity and Log_Rainfall should be ignored for a practical solution.
- Model Tunining with right combinations of hyperparameters using grid search or randomized search is time consuming and not feasible with the limited time we have.
- Limitations in time and skilled members.

# Chapter 03 | Insights on Model Evaluation

The accuracy score is calculated using the **metrics.accuracy_score()** function from scikit-learn to evaluate the model's performance.

```
y_pred = model.predict(X_test)
acc = metrics.accuracy_score(y_test, y_pred)
```

```
Model's accuracy is: 0.9954545454545455
```

And when the saved model is used for predicting the top 3 crops suitable for cultivation under different values of pH level, rainfall, temperature, humidity, N level, P level, K level, it gives 3 predicted crops' names as expected.

```
Enter the level of Nitrogen (N): 22
Enter the level of Phosphorus (P): 71
Enter the level of Potassium (K): 17
Enter the temperature: 18.15300153
Enter the humidity: 19.38602098
Enter the pH value: 5.509295379
Enter the rainfall: 107.6907964
Sample 1: Top Three Crops - ['lettuce', 'cauliflower', 'spinach']
```

# Chapter 04| Suggestions for Better Performance

- Tunining the model with right combinations of hyperparameters using grid search or randomized search can be utilized to get better accuracy in the trained model.
- Augment the dataset through techniques like data synthesis or leveraging external sources to enrich the training set and address class imbalances.
- Experiment with different algorithms and select the one that gives the highest accuracy.
- Employ cross-validation and appropriate evaluation metrics to assess the model's generalization performance accurately.