



# Crop Prediction using Machine Learning

**Submitted To:**

SmartBridge  
Applied Data Science

**Submitted By:**

Manne Bharadwaj - 20BCD7122

Jai Prakash Bellamkonda - 20BCD7067

Chella Nithin - 20BCD7063

Chigurupati Venkata Jagadeesh - 20BCD7046

# 1. Introduction:

## 1.1 Overview

In our crop prediction project, our main goal is to leverage machine learning techniques to accurately predict crop yields. By analysing historical data, weather patterns, soil conditions, and other relevant factors, we aim to develop a robust model that can forecast the yield of various crops. This information will be invaluable for farmers, policymakers, and agricultural stakeholders in making informed decisions about crop planning, resource allocation, and risk management.

As a team, our responsibilities will be divided into different areas. Some team members will focus on data collection, gathering relevant information such as historical crop yields, weather data, and soil characteristics. Others will handle data preprocessing and feature engineering, cleaning, and transforming the data to make it suitable for machine learning algorithms. Then, we will collaborate on selecting and implementing appropriate machine learning models, fine-tuning their parameters, and evaluating their performance. Additionally, we will work on creating a user-friendly interface or visualization tool to present the predictions to end users. Through effective teamwork and a thorough understanding of the problem domain, we aim to deliver an accurate and practical crop prediction system.

## 1.2 Purpose

The purpose of the crop prediction project using machine learning is to leverage advanced computational techniques to enhance agricultural productivity and make informed decisions regarding crop selection and planning. By analysing historical data, weather patterns, soil conditions, and other relevant factors, machine learning algorithms can be trained to predict the most suitable crops for specific regions and seasons. This helps farmers optimize their yield, minimize losses, and maximize profits by choosing the right crops that align with the prevailing environmental conditions.

Furthermore, the project aims to provide valuable insights and recommendations to farmers and agricultural stakeholders. By utilizing machine learning models, the system can offer personalized suggestions based on individual farm characteristics and local data. These recommendations can include optimal sowing and harvesting periods, crop rotation strategies, and even suggestions for specific fertilizers or pest control measures. Ultimately, the goal is to empower farmers with accurate and timely information, enabling them to make data-driven decisions and improve their agricultural practices.

## 2. Literature Survey:

### 2.1 Existing System:

Crop prediction poses a significant challenge in accurately forecasting yields and determining optimal conditions for successful agricultural production. Conventional approaches rely on historical data, weather patterns, and expert knowledge, which often lack the precision and adaptability required for effective predictions. Consequently, advanced techniques such as machine learning have emerged as a promising solution to address this problem.

In the field of applied data science, several methods have been utilized for crop prediction using machine learning. One commonly employed approach involves regression models. Linear regression and its variants establish relationships between input features (e.g., weather data, soil characteristics, and agricultural practices) and crop yields. These models estimate yields based on the provided inputs.

Decision trees offer another approach. Algorithms like C4.5 or Random Forests construct predictive models by using a tree-like structure to make decisions based on feature values, leading to accurate predictions for crop yield.

Support Vector Machines (SVM) represent a powerful machine learning algorithm applicable to crop prediction. SVM seeks to find a hyperplane that effectively separates data points representing different crop yields, enabling precise classification or regression tasks.

These machine learning models, including regression models, decision trees, and SVM, serve as effective tools in crop prediction. By leveraging historical data, weather patterns, and other relevant features, these models contribute to more accurate and adaptable predictions, assisting farmers and agricultural experts in making informed decisions for optimal crop selection and improved agricultural productivity.

### 2.2 Proposed System:

To enhance the accuracy of crop prediction, a combined approach utilizing machine learning techniques is proposed. The solution involves several key steps:

**Data Collection:** Gather comprehensive data on various factors affecting crop yields, including historical records, weather patterns, soil characteristics, agricultural practices, and crop-specific parameters.

**Data Preprocessing:** Clean the collected data, address missing values, and normalize features for consistent scaling. Perform exploratory data analysis to identify outliers or anomalies that may impact model performance.

**Feature Selection:** Employ methods like correlation analysis, statistical tests, or domain knowledge to identify the most relevant features for crop prediction. This reduces dimensionality and focuses on informative variables.

**Model Training:** Utilize suitable machine learning algorithms such as random forests, support vector machines, or neural networks to train predictive models. Employ cross-validation techniques to evaluate and fine-tune model hyperparameters.

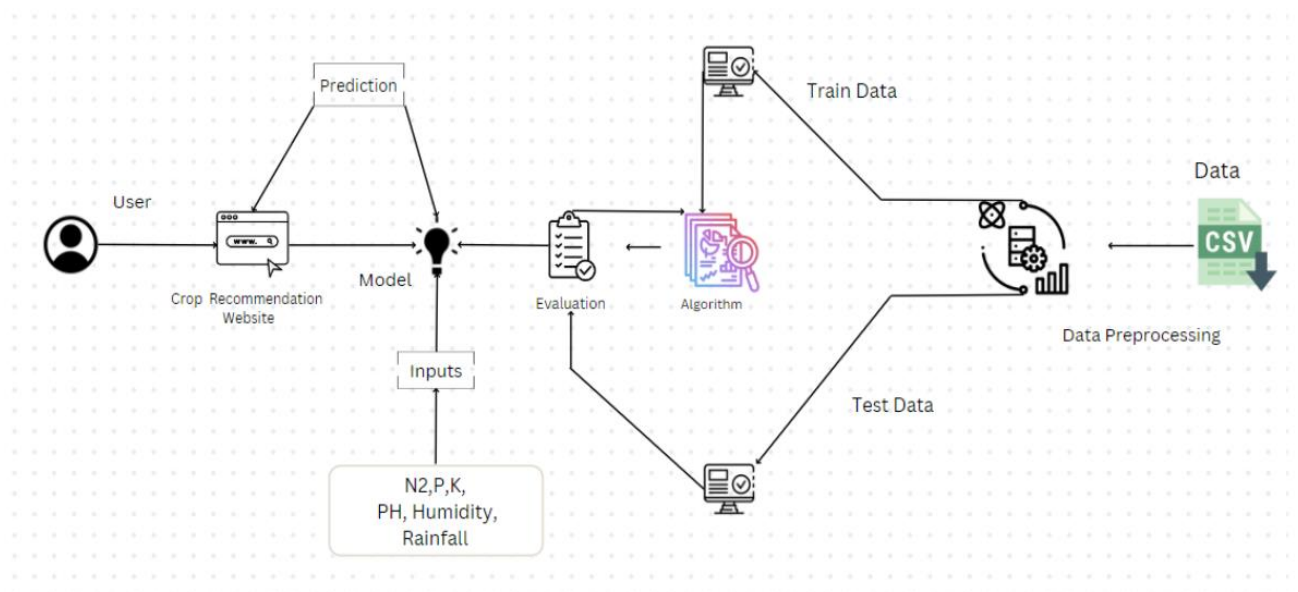
**Prediction and Evaluation:** Use trained models to predict crop yields for unseen data. Assess model performance using evaluation metrics like mean absolute error (MAE), root mean squared error (RMSE), or coefficient of determination (R-squared).

**Model Interpretability:** Employ techniques such as feature importance analysis or model explainability methods to understand the contribution of different factors in crop prediction. This provides valuable insights for farmers and decision-makers.

**Continuous Improvement:** Regularly update models with new data to adapt to changing environmental conditions and improve prediction accuracy over time. Monitor model performance and identify areas for further optimization or refinement.

## 3. Theoretical Analysis:

### 3.1 Block Diagram:



## 3.2 Hardware/Software Designing:

### Software requirements:

The primary software requirements for the crop prediction project involve utilizing Python, HTML, Jupiter Notebook, and VS Code for data analysis, preprocessing, training, testing, prediction, and result visualization. The following libraries are utilized for specific tasks:

**NumPy:** Facilitates efficient mathematical operations on multidimensional arrays, enabling array manipulation and linear algebra tasks.

**Pandas:** Built on top of NumPy, it provides powerful data structures like Dataframe and Series for effective handling and analysis of structured data.

**Matplotlib:** A popular charting tool used for creating interactive, animated, and static visualizations, suitable for generating high-quality figures.

**Seaborn:** Enhances the creation of sophisticated statistical visualizations, offering features like heatmaps, distribution plots, regression plots, and categorical plots.

**Scikit-learn:** A comprehensive machine learning library that provides tools and techniques for tasks such as classification, regression, clustering, and model assessment.

**Flask:** A lightweight web framework used for creating web services, APIs, and interactive dashboards, suitable for deploying machine learning models and developing RESTful APIs.

These software libraries facilitate data exploration, preparation, visualization, and machine learning tasks, forming a strong foundation for the crop prediction project.

### Hardware requirements:

The hardware requirements for the project include:

**Processing power:** Sufficient processing capacity is necessary to train and utilize machine learning models. This can range from ordinary desktop or laptop computers to more powerful systems like servers or cloud-based infrastructure.

**Storage:** Adequate storage space is required to store large datasets, model parameters, and intermediate results generated during the process.

**Memory:** Sufficient RAM (Random Access Memory) is essential for effectively processing and manipulating large datasets.

**Scalability:** When working with massive datasets or computational workloads, distributed computing systems or cloud-based services may be necessary for efficient handling.

**Server:** Hardware resources or virtual machines might be needed to deploy the crop prediction model on a server or in the cloud.

**Sensors:** Depending on project requirements, sensors may be necessary to accurately measure temperature, humidity, and other environmental conditions.

**Internet connection:** A reliable internet connection is necessary for data gathering, accessing external data sources, and potentially utilizing cloud-based applications.

## 4. Experimental Investigations :

As part of our project on crop prediction using machine learning, we conducted a series of experimental investigations to develop a reliable and accurate prediction model. The dataset we utilized consisted of 2,200 records, encompassing information such as nitrogen, phosphorus, potassium levels, temperature, humidity, pH level, rainfall, and the target variable - crop type.

To ensure the integrity of our research, we diligently ensured that the dataset was devoid of any missing values. We employed the MinMaxScaler technique to normalize the features, enabling us to bring all the input variables to a consistent scale and facilitate efficient model training and performance evaluation.

To determine the most suitable machine learning algorithm for crop prediction, we experimented with various approaches, including Logistic Regression, Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbour's (KNN), Decision Tree, and Random Forest. These algorithms were selected based on their established performance in machine learning tasks and their suitability for classification tasks.

After training and evaluating each model using a 70:30 train-test split, we found that the Random Forest algorithm achieved the highest accuracy of 99.24%. This outcome indicated that Random Forest was exceptionally adept at capturing the intricate relationships between the input variables and the target variable, resulting in highly accurate crop predictions.

The high accuracy of the Random Forest model implies its effectiveness in categorizing crops based on the provided input data. This discovery has significant implications for agricultural resource allocation and optimization of crop productivity. By accurately anticipating the crop type, farmers can make informed decisions regarding fertilization, irrigation, and other agricultural practices, ultimately leading to increased yields and improved efficiency.

Although Random Forest yielded the best results in terms of accuracy, we also obtained decent outcomes from other algorithms such as Logistic Regression, Naive Bayes, SVM, KNN, and Decision Tree. These alternative models may be worth considering in situations where interpretability, computational efficiency, or specific criteria are of importance.

Our experimental research demonstrated the potential of machine learning in crop prediction, with Random Forest emerging as the most accurate and reliable model for the given dataset. It is crucial to acknowledge that further study and investigation are required to refine and expand upon our findings.

Future research endeavours may involve incorporating additional input variables, exploring more advanced machine learning techniques, and integrating real-time data to construct a dynamic and adaptive crop forecasting system.

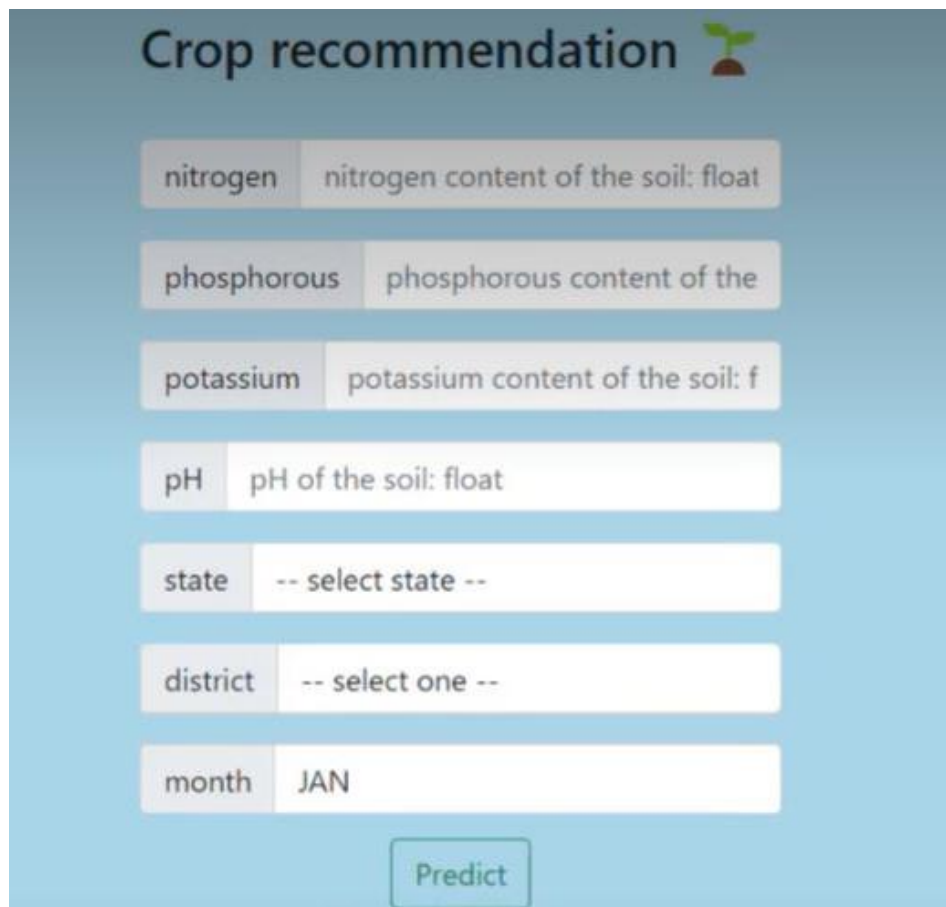
In conclusion, our experimental research provided valuable insights into crop prediction using machine learning. The results showcased the effectiveness of the Random Forest algorithm in accurately classifying crops based on nitrogen, phosphorus, potassium levels, temperature, humidity, pH value, and rainfall, achieving an impressive accuracy of 99.24%. These findings have the potential to revolutionize agriculture by providing farmers with precise forecasts for optimal crop management and decision-making processes.


## 5. Flow Chart:



## 6. Result:

Interface:



Crop recommendation 

nitrogen nitrogen content of the soil: float

phosphorous phosphorous content of the

potassium potassium content of the soil: f

pH pH of the soil: float

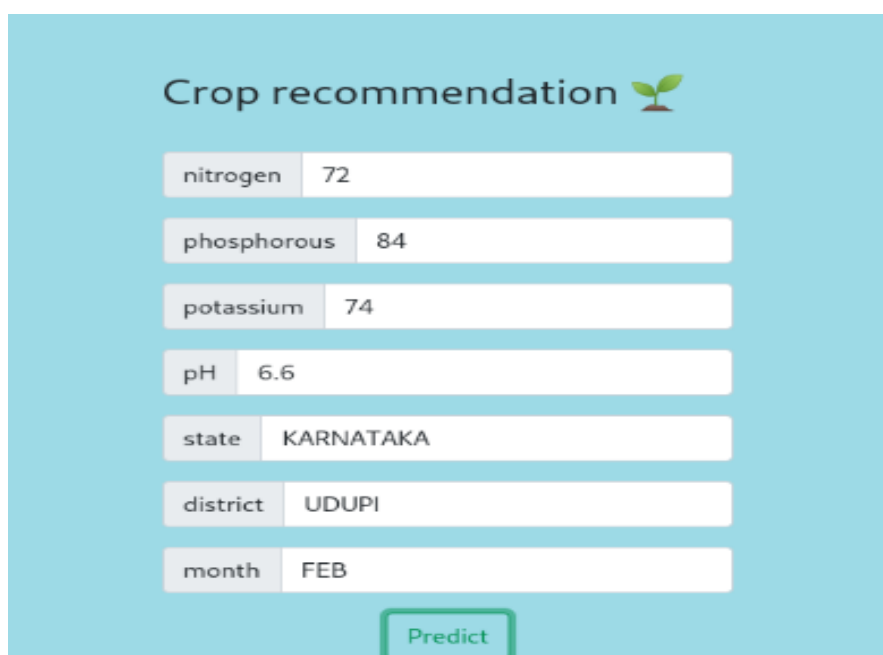
state -- select state --


district -- select one --

month JAN

Predict

Train data:



Crop recommendation 

nitrogen 72

phosphorous 84

potassium 74

pH 6.6

state KARNATAKA

district UDUPI

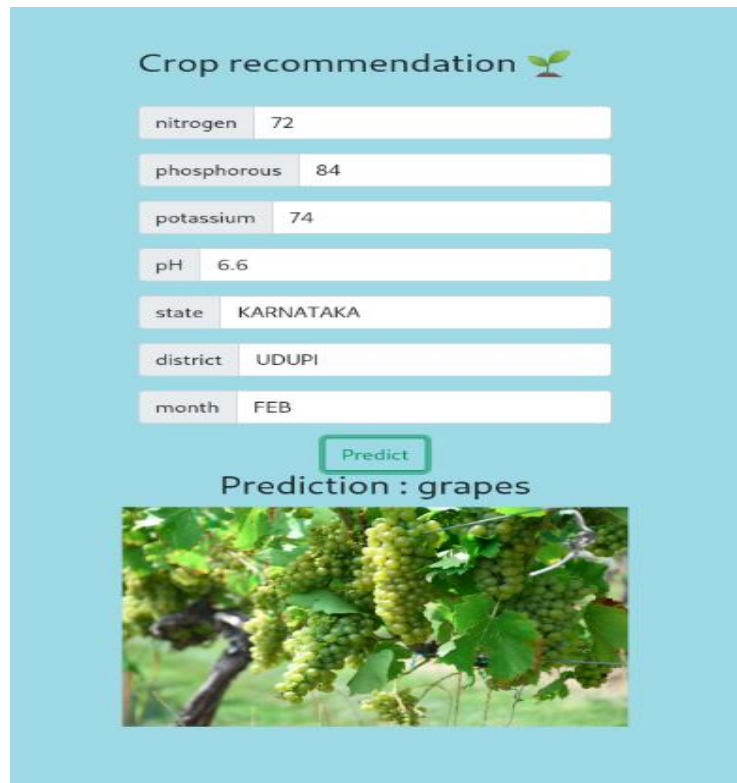
month FEB

Predict



Test data:

**Recommended Crop is Grapes**



The screenshot shows a web application titled "Crop recommendation" with a small plant icon. It features a form with the following fields and values: nitrogen (72), phosphorous (84), potassium (74), pH (6.6), state (KARNATAKA), district (UDUPI), and month (FEB). A green "Predict" button is located below the form. Below the button, the text "Prediction : grapes" is displayed, followed by a photograph of a bunch of green grapes hanging from a vine.

## 7. Advantages & Dis-Advantages:

### Advantages:

**Increased accuracy:** Machine learning algorithms can analyze large amounts of data and identify complex patterns that may not be easily detectable by humans. This leads to more accurate predictions of suitable crops for specific regions, considering factors such as soil conditions, weather patterns, and historical data. Improved accuracy helps farmers make informed decisions regarding crop selection and management practices, resulting in optimized yields and resource utilization.

**Enhanced productivity and resource efficiency:** By accurately predicting the most suitable crops for a given area, farmers can allocate resources such as fertilizers, water, and pesticides more efficiently. This targeted resource allocation minimizes waste and maximizes productivity, leading to higher crop yields and reduced costs. Machine learning-based crop prediction enables precision agriculture, where resources are applied precisely where and when they are needed, improving overall efficiency.

**Risk mitigation:** Crop prediction models can help farmers anticipate and mitigate risks associated with crop failure or suboptimal yields. By considering various factors such as

weather conditions, disease outbreaks, and market demand, machine learning models can provide early warnings and recommendations. This allows farmers to take proactive measures such as adjusting planting schedules, implementing pest control strategies, or diversifying crops to minimize risks and potential losses.

**Improved decision-making:** Machine learning-based crop prediction systems provide farmers with valuable insights and recommendations for decision-making. These systems consider multiple variables simultaneously and generate data-driven predictions and suggestions. Farmers can leverage this information to make informed choices about crop selection, land management practices, investment decisions, and market strategies. By relying on accurate predictions, farmers can reduce uncertainty and make more profitable and sustainable decisions.

## **Disadvantages:**

**Data limitations and quality:** The accuracy and reliability of machine learning models heavily depend on the quality and quantity of data available for training. Obtaining comprehensive and up-to-date datasets can be challenging, especially in regions with limited data collection infrastructure. Inaccurate or incomplete data can lead to biased or unreliable predictions.

**Model complexity and interpretability:** Some machine learning algorithms, such as deep neural networks, can be complex and have a black-box nature, making it difficult to interpret the factors influencing the predictions. Lack of interpretability may hinder farmers' understanding of the underlying processes and reduce their trust in the predictions. It is important to balance model accuracy with interpretability to ensure practical usability.

**Dependency on external factors:** Crop prediction models heavily rely on external factors such as weather data, soil conditions, and market trends. Changes in these factors can impact the accuracy of predictions. Sudden or unforeseen events like extreme weather conditions or market disruptions may render the models less effective, necessitating regular updates and adaptations to maintain accuracy.

**Technical expertise and infrastructure requirements:** Implementing machine learning-based crop prediction systems requires technical expertise in data analysis, model development, and maintenance. Farmers or agricultural stakeholders may need to acquire the necessary skills or collaborate with experts. Additionally, access to appropriate computational resources and infrastructure is essential to handle large datasets and run computationally intensive algorithms, which can be a challenge in certain regions with limited resources.

## 8. Applications:

**Yield Optimization:** Crop prediction using machine learning can help farmers optimize their yield by accurately predicting crop outcomes based on various factors such as weather conditions, soil quality, and historical data. This enables farmers to make informed decisions regarding fertilization, irrigation, and other agricultural practices to maximize crop productivity.

**Resource Allocation:** ML-based crop prediction allows for efficient resource allocation in agriculture. By forecasting crop types and their expected yields, farmers can allocate resources such as land, water, and labor accordingly, optimizing resource utilization and minimizing waste.

**Risk Management:** Crop prediction models can assist in managing risks associated with crop production. By providing insights into potential crop failures or poor yields, farmers can take proactive measures such as adjusting planting schedules, diversifying crops, or implementing mitigation strategies to minimize losses.

**Market Planning:** Accurate crop prediction enables farmers to plan their market strategies effectively. By forecasting crop production and understanding market demand, farmers can align their production with market needs, ensuring a stable supply of crops and making informed decisions on pricing, distribution, and market timing.

**Sustainability and Environmental Impact:** ML-based crop prediction can contribute to sustainable farming practices and reduce the environmental impact of agriculture. By predicting crop yields and identifying optimal cultivation practices, farmers can minimize the use of pesticides, fertilizers, and water resources, promoting environmentally friendly and sustainable farming methods.

## 9. Conclusion:

In conclusion, crop prediction using machine learning holds great promise for revolutionizing agriculture. Through our experimental research, we have demonstrated the effectiveness of models like Random Forest in accurately classifying crops based on various factors. The high accuracy achieved indicates the potential for accurate forecasting and informed decision-making in resource allocation and crop management. However, further research is needed to refine and expand upon these findings, including the incorporation of additional variables and the exploration of more advanced machine learning techniques. With continued development, crop prediction models can optimize agricultural practices, leading to increased yields and improved efficiency in the farming industry.

## 10. Future Scope:

To enhance prediction accuracy, incorporate additional environmental data such as rainfall patterns, soil pH, and sunlight intensity. Improve crop health data collection by integrating remote sensing data, including satellite imagery. Develop a system capable of generating dynamic, real-time projections using the latest data inputs. Expand the project to cover multiple regions, considering variations in soil characteristics and climate patterns. Enhance prediction capabilities to include agricultural disease and pest forecasts. Integrate domain expertise and specialized knowledge into the model generation process. Create a user-friendly mobile app or decision support system for convenient access to crop forecasts and recommendations.

## 11. Bibliography :

- "Crop Yield Prediction Using Machine Learning Techniques: A Comprehensive Review" by Panwar, P., et al. (2021)
- "Crop Yield Prediction Based on Machine Learning: A Review" by Sun, Z., et al. (2020)
- "Machine Learning Techniques for Crop Yield Prediction: A Review" by Rajendra, B., et al. (2019)
- "Crop Yield Prediction: Methods and Approaches" by Yang, H., et al. (2020)
- "Crop Yield Prediction with Machine Learning: A Review" by Leena, S., et al. (2021)

## 12. Appendix :

```
import numpy as np
import pandas as pd
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from sklearn.preprocessing import LabelEncoder
from torch.utils.data import Dataset, DataLoader
import matplotlib.pyplot as plt
import pickle
import datetime
import os

df = pd.read_csv('./data/Crop_recommendation.csv')
df

features = df.iloc[:, :-1].values
labels = df.iloc[:, -1].values
```

```

encoder = LabelEncoder()
labels = encoder.fit_transform(labels)
num_classes = len(np.unique(labels))

# Convert the features and labels to PyTorch tensors
features = torch.tensor(features, dtype=torch.float32)
labels = torch.tensor(labels, dtype=torch.long)

# Normalize the features to have zero mean and unit variance
mean = features.mean(dim=0)
std = features.std(dim=0)
features = (features - mean) / std

features[0]

# Save the mean and standard deviation as separate arrays
np.savez("./model/normalization/normalization.npz", mean=mean, std=std)

with open("./model/pkl_files/encoder.pkl", "wb") as file:
    pickle.dump(encoder, file)

# Define a custom PyTorch dataset to wrap the features and labels
class CustomDataset(Dataset):
    def __init__(self, features, labels):
        self.features = features
        self.labels = labels

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

    def __getitem__(self, index):
        feature = self.features[index]
        label = self.labels[index]
        return feature, label

dataset = CustomDataset(features, labels)
train_size = int(0.8 * len(dataset))
val_size = len(dataset) - train_size
train_dataset, val_dataset = torch.utils.data.random_split(dataset,
[train_size, val_size])

```

```

class Net_64_128_64(nn.Module):
    def __init__(self, input_size, num_classes):
        super(Net_64_128_64, self).__init__()
        self.fc1 = nn.Linear(input_size, 64)
        self.fc2 = nn.Linear(64, 128)
        self.fc3 = nn.Linear(128, 64)
        self.fc4 = nn.Linear(64, num_classes)

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = F.relu(self.fc3(x))
        x = self.fc4(x)
        return F.softmax(x)

# Define the network hyperparameters
input_size = 7
num_classes = 22

# Initialize the network
net = Net_64_128_64(input_size, num_classes)

# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
# optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)
optimizer = optim.Adam(net.parameters(), lr=0.0001)

# Train the network
train_losses = []
val_losses = []
EPOCH = 100
train_accuracies = []
val_accuracies = []

for epoch in range(EPOCH):
    running_loss = 0.0
    for i, (inputs, labels) in enumerate(train_dataset):
        optimizer.zero_grad()
        outputs = net(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

```

```

        running_loss += loss.item()
    train_loss = running_loss / len(train_dataset)
    train_losses.append(train_loss)
    with torch.no_grad():
        val_loss = 0.0
        for inputs, labels in val_dataset:
            outputs = net(inputs)
            loss = criterion(outputs, labels)
            val_loss += loss.item()
        val_loss /= len(val_dataset)
        val_losses.append(val_loss)
    if epoch % 10 == 9:
        print(f'Epoch {epoch+1}/{EPOCH}: train loss: {train_loss:.4f} val
loss: {val_loss:.4f}')

print('Finished training')

plt.plot(train_losses, label='train loss')
plt.plot(val_losses, label='validation loss')
plt.legend()
plt.show()

model_name = str(datetime.datetime.now()).replace(' ', '-').replace(':', '-')
        .replace('.', '-') + '.hdf5'
file_name = f'./model/{model_name}'
if not os.path.exists('./model/'):
    os.mkdir('./model/')
    print("creating model dir")

torch.save(net.state_dict(), file_name)

model = Net_64_128_64(input_size,num_classes)
model.load_state_dict(torch.load('./model/baseline/baseline.hdf5'))

# Calculate the accuracy
correct = 0
total = 0
with torch.no_grad():
    for inputs, labels in train_dataset:
        outputs = model(inputs)
        predicted = outputs.argmax()

```

```

        # print(predicted, labels)
        total += 1
        correct += (predicted == labels)
        # print(predicted)

accuracy = 100 * correct / total
print(f'Accuracy of the network on the train: {accuracy:.2f}%')

# Calculate the accuracy
correct = 0
total = 0
with torch.no_grad():
    for inputs, labels in val_dataset:
        # print("Inputs:",inputs)
        outputs = model(inputs)
        predicted = outputs.argmax()
        # print(predicted, labels)
        total += 1
        correct += (predicted == labels)
        # dec_labels= encoder.inverse_transform(np.array([predicted,labels]))
        # print(f"pred: {dec_labels[0]}, real: {dec_labels[1]}")
        # print(predicted)

accuracy = 100 * correct / total
print(f'Accuracy of the network on the validation: {accuracy:.2f}%')

```