Intelihack NextGen

Task 01

Solution By

Team: Devin 2.0

University Of Colombo School of Computing

May 4, 2024

Report

Approach

1. **Data Preparation:** The dataset was loaded into a Pandas DataFrame and inspected for data structure and statistical properties. Missing values were removed to ensure data consistency. Features and labels were separated, with the label column being `Label_Encoded`.
2. **Feature Engineering:** Numerical features were standardized using `StandardScaler` to improve model performance and convergence.
3. **Model Training:** The dataset was split into training and testing sets (80/20 split). A Random Forest Classifier with 100 estimators was chosen for its robustness and non-linear modeling capabilities. The model was trained on the training set to learn patterns in the data.
4. **Model Evaluation:** The model was evaluated using accuracy and classification metrics (precision, recall, F1-score). Results showed high accuracy and consistent classification performance across all crop labels.
5. **Model Serialization:** The trained model was saved using Joblib to ensure it could be reused without retraining. The model was reloaded and tested to verify consistent performance.

Challenges Faced

- **Data Preprocessing:** Handling missing values was crucial to ensure data consistency.
- **Feature Selection:** Identifying which features should be used and how to handle feature scaling required careful planning.
- **Model Tuning:** Choosing the right model and configuring its parameters was challenging. The initial approach focused on using a standard Random Forest setup.

Insights Gained

- **Model Robustness:** The Random Forest model performed exceptionally well, indicating that it is suitable for this type of data.
- **Feature Importance:** Understanding which features contribute most to the model's performance could help simplify the model without reducing accuracy.
- **Reproducibility:** Saving and loading the model using Joblib ensures consistent performance, crucial for deploying machine learning models in real-world scenarios.

Suggestions for Improvement

- **Hyperparameter Tuning:** Perform hyperparameter tuning (e.g., `GridSearchCV`) to find optimal model parameters, which may further improve performance.
- **Cross-Validation:** Use k-fold cross-validation to ensure the model generalizes well across different data splits.
- **Feature Engineering:** Investigate further feature engineering techniques to identify which features contribute most to model performance.
- **Advanced Models:** Consider experimenting with more advanced models like Gradient Boosting Machines or XGBoost, which may yield better results in certain scenarios.

Instructions for Reproducing the Results

1. **Environment Setup:** Ensure Python is installed (version 3.6 or higher). Install the required libraries by running:

   ```
   pip install pandas scikit-learn joblib
   ```

2. **Dataset Preparation:** Place the dataset `Crop_Dataset.csv` in the appropriate directory, such as:

   ```
   mkdir -p ./mnt/data/
   mv /path/to/Crop_Dataset.csv ./mnt/data/
   ```

3. **Save the Code:** Copy the Python code into a `.py` file, e.g., `crop_recommendation.py`.

4. **Run the Code:** Execute the Python script using the command line:

   ```
   python crop_recommendation.py
   ```

5. **Evaluate Results:** The output will show the model's accuracy and a detailed classification report.