Project-8 Decision Tree and Neural Networks on Crop Recommendation Dataset

Vinay Achanta and James Sanford-Luevano

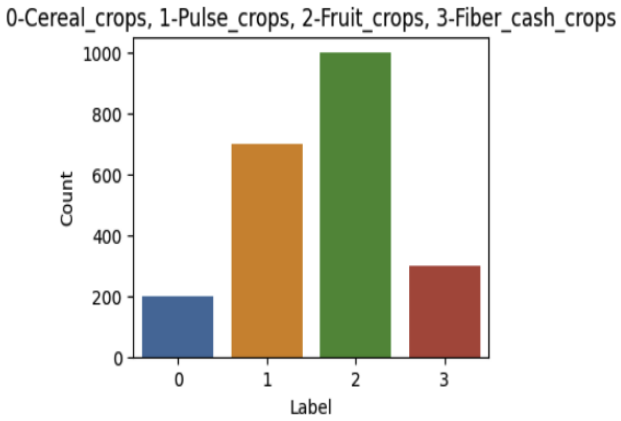
Introduction

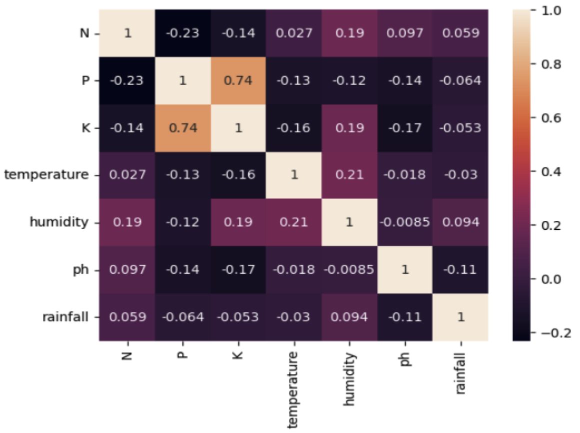
Crop recommendation is an important task in agriculture as it helps farmers choose the best crop to grow in their fields, taking into account various environmental factors. In this project, we used a dataset of crop and environmental factors for crop recommendation from Kaggle. We will apply a decision tree and neural network algorithms to build a model that can recommend the best crop to grow based on environmental factors and evaluate the model's performance with various accuracy metrics. We will also explore the effect of maximum depth on bias and variance of the decision tree model. Also, we will compare the performance of a decision tree classifier to a neural network model with different architectures.

GitHub Link
Presentation Slides Link

Dataset

We used Crop Recommendation Dataset from Kaggle in this project which contains 2200 rows of data. The dataset includes attributes like nitrogen(N), phosphorus(P), and potassium(K), temperature, humidity, pH, rainfall, and the label for the crop type that corresponds to a given set of attributes. This project helps farmers in selecting the best crop to grow. The dataset has been preprocessed and any missing values have been dropped and standardized attributes. Additionally, the correlation between the attributes has been analyzed and the "K" attribute has been dropped as it was found to be highly correlated with "P". The crop names have been categorized into four categories - cereal crops, pulse crops, fruit crops, and fiber cash crops which helps to provide better insights.





Analysis Technique

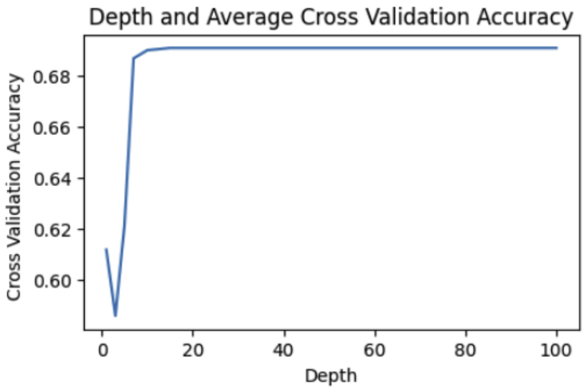
Decision Tree:

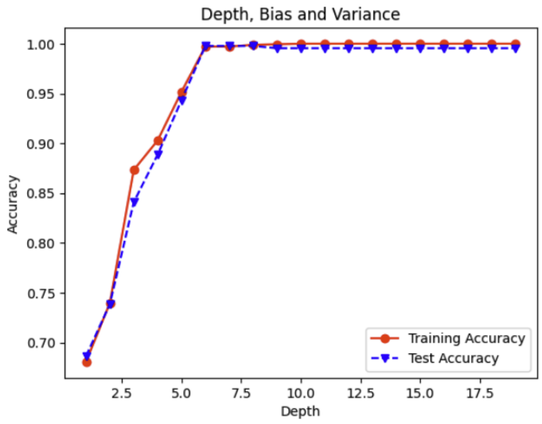
To evaluate the performance of decision tree classifiers, several analysis techniques have been used, including cross-validation, classification report, confusion matrix, accuracy, and F1 scores. These approaches have been taken to know the effect of depth on bias and variance and to know the algorithm's choices in selecting attributes to split on.

First, cross-validation was used to train a decision tree classifier for different depths, and the average cross-validation accuracy was computed for each depth. The results show that the accuracy initially improves with the increase in depth, reaching a maximum of 69.09% at depth 10. Beyond that, the accuracy flattens out, indicating overfitting of the training data.

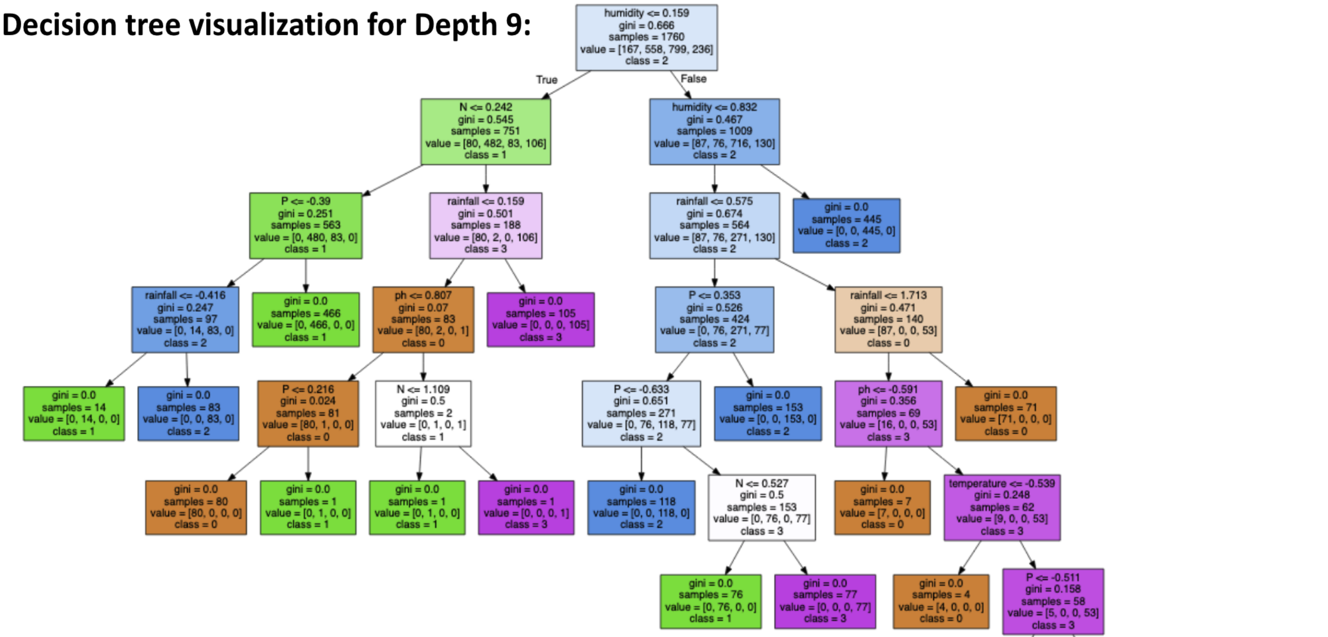
Second, two decision trees were trained with different depths on the training data, and their performance was evaluated. The decision tree classifier with a depth of 1 had poor performance, with an accuracy score of only 69%, whereas the decision tree classifier with a depth of 9 had excellent performance, with an accuracy score of 100%. The decision tree classifier with a depth of 1 performs badly at both identifying true positives and averting false positives, having low accuracy and low recall. With excellent precision and recall, the decision tree classifier with a depth of 9 does a good job at both detecting true positives and avoiding false positives. While the confusion matrix for the decision tree with depth 9 indicates that it properly predicted the majority of data, the confusion matrix for the decision tree with depth 1 indicates that a small proportion of samples were misclassified.

Third, the decision tree was plotted at different depths. The results showed that increasing the depth of the decision tree can reduce bias and increase accuracy up to a certain point, but beyond that point, it leads to overfitting and high variance. We can get a balance between bias and variance by selecting an appropriate depth for the decision tree model. Also, keeping the depth too low leads to high bias, while increasing the depth beyond a certain point leads to overfitting and high variance. By this, we can conclude that it is essential to select an optimal value of depth to ensure that the decision tree model achieves the best performance on the given dataset.

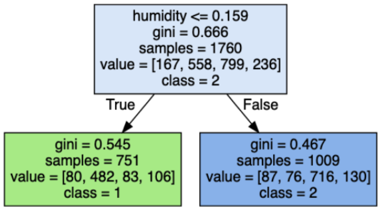


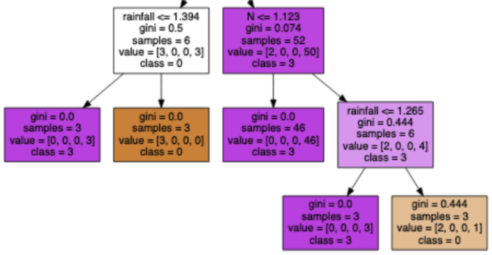


Decision tree visualization for Depth 9:


humidity <= 0.159 gini = 0.666 samples = 1760 value = [167, 558, 799, 236] class = 2
True
False
N <= 0.242 gini = 0.545 samples = 751 value = [80, 482, 83, 106] class = 1
humidity <= 0.832 gini = 0.467 samples = 1009 value = [87, 76, 716, 130] class = 2
P <= -0.39 gini = 0.251 samples = 563 value = [0, 480, 83, 0] class = 1
rainfall <= 0.159 gini = 0.501 samples = 188 value = [80, 2, 0, 106] class = 3
rainfall <= 0.575 gini = 0.674 samples = 564 value = [87, 76, 271, 130] class = 2
gini = 0.0 samples = 445 value = [0, 0, 445, 0] class = 2
rainfall <= -0.416 gini = 0.247 samples = 97 value = [0, 14, 83, 0] class = 2
gini = 0.0 samples = 466 value = [0, 466, 0, 0] class = 1
ph <= 0.807 gini = 0.07 samples = 83 value = [80, 2, 0, 1] class = 0
gini = 0.0 samples = 105 value = [0, 0, 0, 105] class = 3
P <= 0.353 gini = 0.526 samples = 424 value = [0, 76, 271, 77] class = 2
rainfall <= 1.713 gini = 0.471 samples = 140 value = [87, 0, 0, 53] class = 0
gini = 0.0 samples = 14 value = [0, 14, 0, 0] class = 1
gini = 0.0 samples = 83 value = [0, 0, 83, 0] class = 2
P <= 0.216 gini = 0.024 samples = 81 value = [80, 1, 0, 0] class = 0
N <= 1.109 gini = 0.5 samples = 2 value = [0, 1, 0, 1] class = 1
P <= -0.633 gini = 0.651 samples = 271 value = [0, 76, 118, 77] class = 2
gini = 0.0 samples = 153 value = [0, 0, 153, 0] class = 2
ph <= -0.591 gini = 0.356 samples = 69 value = [16, 0, 0, 53] class = 3
gini = 0.0 samples = 71 value = [71, 0, 0, 0] class = 0
gini = 0.0 samples = 80 value = [80, 0, 0, 0] class = 0
gini = 0.0 samples = 1 value = [0, 1, 0, 0] class = 1
gini = 0.0 samples = 1 value = [0, 1, 0, 0] class = 1
gini = 0.0 samples = 1 value = [0, 0, 0, 1] class = 3
gini = 0.0 samples = 118 value = [0, 0, 118, 0] class = 2
N <= 0.527 gini = 0.5 samples = 153 value = [0, 76, 0, 77] class = 3
gini = 0.0 samples = 7 value = [7, 0, 0, 0] class = 0
temperature <= -0.539 gini = 0.248 samples = 62 value = [9, 0, 0, 53] class = 3
gini = 0.0 samples = 76 value = [0, 76, 0, 0] class = 1
gini = 0.0 samples = 77 value = [0, 0, 0, 77] class = 3
gini = 0.0 samples = 4 value = [4, 0, 0, 0] class = 0
P <= -0.511 gini = 0.158 samples = 58 value = [5, 0, 0, 53] class = 3

Decision tree visualization for Depth 1:


humidity <= 0.159 gini = 0.666 samples = 1760 value = [167, 558, 799, 236] class = 2
True
False
gini = 0.545 samples = 751 value = [80, 482, 83, 106] class = 1
gini = 0.467 samples = 1009 value = [87, 76, 716, 130] class = 2


rainfall <= 1.394 gini = 0.5 samples = 6 value = [3, 0, 0, 3] class = 0
N <= 1.123 gini = 0.074 samples = 52 value = [2, 0, 0, 50] class = 3
gini = 0.0 samples = 3 value = [0, 0, 0, 3] class = 3
gini = 0.0 samples = 3 value = [3, 0, 0, 0] class = 0
gini = 0.0 samples = 46 value = [0, 0, 0, 46] class = 3
rainfall <= 1.265 gini = 0.444 samples = 6 value = [2, 0, 0, 4] class = 3
gini = 0.0 samples = 3 value = [0, 0, 0, 3] class = 3
gini = 0.444 samples = 3 value = [2, 0, 0, 1] class = 0

Neural Network:

Here in Neural Network, we tried three different architectures, each with different numbers of hidden layers and nodes in each layer. We used the MLPClassifier class from sci-kit-learn to create the neural network models. We also visualized each architecture using a graph.

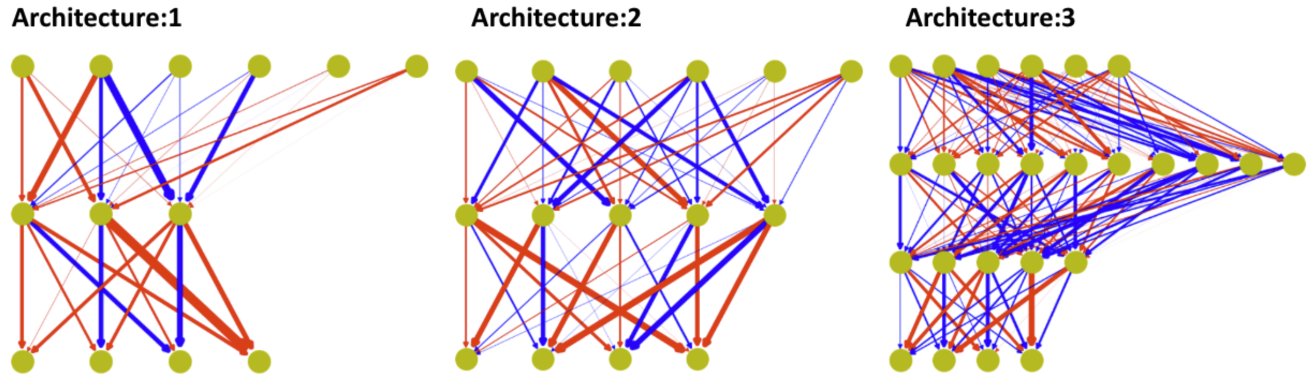
Architecture 1:
The first architecture used only one hidden layer with 3 nodes. The model was trained using the training data and tested using the testing data. The confusion matrix and classification report were calculated, and the accuracy was 0.85. The model performed well for classes 1, 2, and 3 but poorly for class 0. The precision for class 0 was 0.60, the recall was 0.45, and the F1 score was 0.52. The F1 score for classes 1, 2, and 3 were 0.89, 0.90, and 0.78, respectively.

Architecture 2:
The second architecture used only one hidden layer with 5 nodes. The confusion matrix and classification report were calculated, and the accuracy score was 0.93. The model performed well for all classes, and the F1 score for all classes were 0.64, 0.98, 0.97, and 0.83, respectively. The second architecture performed better than the first architecture in terms of precision, recall, and F1 score for all classes. The

confusion matrix also shows that the second architecture correctly predicted more instances of all classes than the first architecture.

Architecture 3:
The third architecture used two hidden layers with 10 and 5 nodes, respectively. The confusion matrix and classification report were calculated, and the accuracy score was 0.97. The model performed well for all classes, and the F1 score for all classes were 0.90, 1.0, 0.99, and 0.94, respectively. The third architecture performed the best among all the architectures in terms of precision, recall, and F1 score for all classes. The confusion matrix also showed that the third architecture correctly predicted more instances of all classes than the other architectures.

We found that the performance of the models improved with the increasing complexity of the architecture.



Results

In this project, we have evaluated the performance of decision trees, and neural networks, for a crop recommendation dataset.

For decision trees, we have used different analysis techniques, including cross-validation, classification report, confusion matrix, accuracy score, and visualization. We have trained decision tree classifiers with different depths, and their average cross-validation accuracy was computed for each depth. The results show that the accuracy initially improves with the increase in depth, reaching a maximum of 69.09% at depth 10. Beyond that, the accuracy flattens out, indicating overfitting of the training data. Two decision trees were trained with different depths. The decision tree classifier with a depth of 1 had poor performance, with an accuracy score of only 69%, whereas the decision tree classifier with a depth of 9 had excellent performance, with an accuracy score of 100%. Increasing the depth of the decision tree can reduce bias and increase accuracy up to a certain point, but beyond that point, it leads to overfitting and high variance. We can get a balance between bias and variance by selecting an appropriate depth for the decision tree model. We also analyzed the algorithm's choices in splitting attributes and found that the "humidity" attribute was deemed most important(as it has high gini inpurity value) in determining the target variable. The Scikit-learn module's decision tree classifier uses Gini impurity as the default criterion for splitting attributes and at a given tree level, the algorithm does not necessarily split on the same attribute for all nodes. Different attributes may be selected for splitting at each level depending on the attribute’s Gini impurity.

For neural networks, we have tried three different architectures, each with different numbers of hidden layers and nodes in each layer. We used the MLPClassifier class from sci-kit-learn to create the neural network models. We calculated the confusion matrix, classification report, and F1 score for each architecture. We also visualized each architecture using a graph. The first architecture used only one hidden layer with 3 nodes. The accuracy was 0.85. The model performed well for classes 1, 2, and 3 but poorly for class 0. The second architecture used only one hidden layer with 5 nodes. The accuracy score was 0.93. The model performed well for all classes, and the second architecture outperformed the first architecture in terms of precision, recall, and accuracy for all classes. The third architecture used two hidden layers with 10 and 5 nodes, respectively. The accuracy score was 0.97. The model performed the best among all the architectures in terms of precision, recall, and accuracy for all classes. We analyzed the correlation between edge weights and decision trees. However, no significant correlations were found. Alternatively, We might have used a variety of other strategies, including feature selection, and ensemble methods like bagging, and dropout, to enhance the performance of our model. Furthermore, normalization, scaling, or PCA preprocessing methods may be helpful in lowering the number of dimensions in the data and enhancing the model's accuracy.

In conclusion, both decision trees and neural networks were able to classify the crop recommendation dataset accurately. The decision tree with depth 9 and the architecture used two hidden layers with 10 and 5 nodes performed the best among all the models.