Crop Recommendation Capstone

C.G. Riffle

2023-03-20

Contents

Introduction

The expansion of technology coupled with predictive modeling and artificial intelligence (AI) will continue to drive the decision making process in every area of the economy from customer preference targeting through social media for products and services, to making informed decisions for crop production based on classification models. Recommendation systems are used across many sectors including eCommerce, academia, farming, and other industries to facilitate the recommendation of products and services based on indicators such as personal preferences or measured values and features to make classification recommendations. Recommendation systems leverage machine learning techniques applied to data sets. The process of building a recommendation system uses different strategies to filter data. For example content based filtering (based on item features) and collaborative filtering (based on user preferences or responses) are commonly used to build recommendation systems. The value for these system lies in the ability of the system to accurately recommend products and services ranging from movies, clothes, restaurants, books, and cars. An alternative approach is to build a classification recommendation system to support decision making processes in industries such as healthcare and precision farming. Recommendation system applications are valuable tools for saving time and money.

The goal for this Capstone project was to build and test a classification model to recommend crops for urban farmers using soil and climate data. Rather than focusing on user based responses and collaborative filtering for development of a model, modeling techniques suitable for classification models were used that offered a different set of challenges with a data set containing 22 possible outcomes. Precision agriculture and urban farming fall into my life science research and horticulture background. As urban areas continue to expand, precision agriculture and technology enhancements will become vital to maximize crop selection and growth in smaller spaces.

Data Set and Variables

A “Crop Recommendation Dataset” from India in kaggle was selected and downloaded as a csv file for this analysis (1, 2). The data set was used to build three models for precision farming that enables users to recommend crops based on seven variable data fields. The link to the license is included (3). The data set contains 22 variety of crops, some of which are typically grown in specific countries or climates/zones. some crops were unknown to me (e.g., moth beans, pigeon peas). Moth beans are a high-protein legume resistant to drought (4). This crop should be able to grow in an area with lower rainfall and humidity values.

The data contains seven variables that align with two categories each containing measured variables used as predictors for crop recommendation. The predictors in this data set are all important for plant growth.

Soil data: macronutrients (Phosphorous, Potassium, and Nitrogen) provide nutrients to the crops and are three main components of fertilizers.

Climate data: temperature, humidity, and rainfall.

The predictors are described below (1).

Nitrogen (N) Macronutrient represented as the ratio of Nitrogen content in the soil.

Phosphorous (P) Macronutrient represented as the ratio of Phosphorous content in the soil.

Potassium (K) Macronutrient represented as the ratio of Potassium content in the soil.

Temperature Represented as degrees Celsius. Involved in most plant processes.

Humidity Represented as percent Relative Humidity (%RH). Source of water to plants.

pH Measured value of the soil. It has a measured range from 0 to 14 with the low end representing acidic values, the center of the range representing neutral values and the high end representing basic values (5).

Rainfall Represented as mm. Source of water to plants.

Label Represents the crop to recommend.

Objectives and Approach

The goal of this work was to recommend a crop to grow based on soil characteristics (macromolecules and pH) and climate (temperature, humidity, rainfall). Models and approaches learned throughout and in the course book (6), as well as supplemental literature referenced throughout this report were used in understanding and development of the models. Literature suggested a few approaches to examine data that are appropriate for classification models used for a recommendation system including Naive Bayes, KNN, Random Forest, logistic regression, and Support Vector Machine (7, 8, 9). Principal Component Analysis (PCA) was included in the model as a method to reduce dimensions and maintain variance (10).

Approach: Packages were coded to automatically load followed by libraries used in the analysis. The csv file was loaded through github and data split into train and test data using the caret package. Initial exploration was done with the entire data set understand the structure and basis statistics. Data was processes and cleaned for analysis and modeling. Analysis from the csv downloaded to my computer indicated the first three columns were integers so they were converted to numeric values. The label column representing crops was converted to factors later in the analysis. A Random Forest model was run using the seven predictors which gave a high level of accuracy. The variable importance was checked and the analysis repeated with the top four variables.

Data were normalized and correlation of the predictor variables checked. A Principal Component Analysis (PCA) was run to see if the dimensions could be reduced while maintaining a majority of the variance. A scree plot and bi-plot were both included to understand how many components contribute to variance and a bi-plot to to help visualize similarities and their relative importance (11). The correlation was rechecked with PCA's to make sure multicolinearlity issues were addressed. Multinomial regression was performed using the first five components for prediction and the mis-classification error determined with the test set.

Methods and Analysis

Preparation of Data Sets

Load Libraries and Packages

Seven packages were included to automatically load. The following libraries were loaded and used for the analysis: readr, ggplot2, corrplot, dplyr, caret, tidyverse, ggcorrplot, FactoMineR, factoextra, class, randomForest, and nnet.

Load the Data Set The data set from kaggle was a compressed zip file (csv). The file was downloaded for analysis and copied to github to make available through the code below.

```
# Load the data
crop_data <- read_csv("https://raw.github.com/CGRiffle/crop-capstone-project/main/Crop_recommendation.cs
```

```
## Rows: 2200 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): label
## dbl (7): N, P, K, temperature, humidity, ph, rainfall
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Split Data The data set was split into a train and test set for model development and testing.

```
# Split the data set into a train and test set.
set.seed(1, sample.kind="Rounding")
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
test_index <- createDataPartition(y = crop_data$label, times = 1, p = 0.1, list = FALSE)

train_crop <- crop_data[-test_index,]
test_crop <- crop_data[test_index,]
```

Data Exploration and Visualization

The entire crop recommendation data set was used to explore and visualize the data prior to model development. Basic code functions were used to understand the structure, dimensions, and features of the data set. Data processing and cleaning was then performed to change 3 integer columns to numeric values (from csv file downloaded). The "label" column representing the crops was converted to a factor later in the analysis. The 7 predictor variables were categorized as either a macromolecule, pH, or a climate variable. The data distribution to understand the measurement range for each crop and and some representative histograms were used to understand additional information on the relationship observed between crops for each variable.

Data Structure

The basic data structure was determined using the str() function for the entire data set. The crop data set is a data.frame with 2200 observations of 8 variables including Nitrogen (N), Phosphorous (P), Potassium (K), temperature, humidity, pH (ph), rainfall, and the crop name (label). Each row represents a unique observation of 7 variables and a crop associated with the measured variables. Each variable has the data type (class) defined. The original data download to my hard drive had both numbers and integers and one character column representing the crops. Moving data access through github resulted in the classes below.

```
## spc_tbl_ [2,200 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ N          : num [1:2200] 90 85 60 74 78 69 69 94 89 68 ...
##  $ P          : num [1:2200] 42 58 55 35 42 37 55 53 54 58 ...
##  $ K          : num [1:2200] 43 41 44 40 42 42 38 40 38 38 ...
##  $ temperature: num [1:2200] 20.9 21.8 23 26.5 20.1 ...
##  $ humidity   : num [1:2200] 82 80.3 82.3 80.2 81.6 ...
##  $ ph         : num [1:2200] 6.5 7.04 7.84 6.98 7.63 ...
##  $ rainfall   : num [1:2200] 203 227 264 243 263 ...
##  $ label      : chr [1:2200] "rice" "rice" "rice" "rice" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   N = col_double(),
##   ..   P = col_double(),
##   ..   K = col_double(),
##   ..   temperature = col_double(),
##   ..   humidity = col_double(),
##   ..   ph = col_double(),
##   ..   rainfall = col_double(),
##   ..   label = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

The summary() function was used to understand some basic statistics for the data based on the class (data.frame) and type of data (e.g. numeric). The data is summarized for each column in the the data set.

The presence of no NA's listed indicates no data is missing (verified agian later in the analysis). All of the columns in the data set, except the label, has a mean and median value listed along with data to understand the range, minimum and maximum values. The "label" column describes the class (character), length, and mode. The macromolecules (N, P, and K) have similar ranges (K is slightly higher).

```
##        N                P                K            temperature
##  Min.   :  0.00   Min.   :  5.00   Min.   :  5.00   Min.   : 8.826
##  1st Qu.: 21.00   1st Qu.: 28.00   1st Qu.: 20.00   1st Qu.:22.769
##  Median : 37.00   Median : 51.00   Median : 32.00   Median :25.599
##  Mean   : 50.55   Mean   : 53.36   Mean   : 48.15   Mean   :25.616
##  3rd Qu.: 84.25   3rd Qu.: 68.00   3rd Qu.: 49.00   3rd Qu.:28.562
##  Max.   :140.00   Max.   :145.00   Max.   :205.00   Max.   :43.675
##     humidity           ph           rainfall         label
##  Min.   :14.26   Min.   :3.505   Min.   : 20.21   Length:2200
##  1st Qu.:60.26   1st Qu.:5.972   1st Qu.: 64.55   Class :character
##  Median :80.47   Median :6.425   Median : 94.87   Mode  :character
##  Mean   :71.48   Mean   :6.469   Mean   :103.46
##  3rd Qu.:89.95   3rd Qu.:6.924   3rd Qu.:124.27
##  Max.   :99.98   Max.   :9.935   Max.   :298.56
```

While there are seven variables that can be used to predict the crop, exploration of the crops was performed by first identifying the number of unique crops using the n_distinct() function and then summarizing the number of observations for each crop using the summarize function. Table 1 shows 22 distinct crops in the data set, each with 100 observations.

Table 1: **Distribution of Observations for Each Crop**

Crop Name	Number of Observations
apple	100
banana	100
blackgram	100
chickpea	100
coconut	100
coffee	100
cotton	100
grapes	100
jute	100
kidneybeans	100
lentil	100
maize	100
mango	100
mothbeans	100
mungbean	100
muskmelon	100
orange	100
papaya	100
pigeonpeas	100
pomegranate	100
rice	100
watermelon	100

The data set was checked for null values using the following code, colSums(is.na(crop_data)); and, the first six rows examined using the head() function.

```
## # A tibble: 6 x 8
##       N     P     K temperature humidity    ph rainfall label
##   <dbl> <dbl> <dbl>       <dbl>    <dbl> <dbl>    <dbl> <chr>
## 1    90    42    43        20.9     82.0  6.50     203. rice
## 2    85    58    41        21.8     80.3  7.04     227. rice
## 3    60    55    44        23.0     82.3  7.84     264. rice
## 4    74    35    40        26.5     80.2  6.98     243. rice
## 5    78    42    42        20.1     81.6  7.63     263. rice
## 6    69    37    42        23.1     83.4  7.07     251. rice
```

Data Preparation and Cleaning

In order to do Principal Component Analysis (PCA), KNN, and other models; the first three variables (N, P, K) were converted from integers to numeric variables so calculations can be applied appropriately (based on using the downloaded csv file). The "label"character variable was converted to a factor for other analysis later in this document. Changes for the test set were also applied. All changes to prepare and clean data were checked using the str() function.

```
# convert the first three columns of the training and test sets to numeric values.
train_crop[,1:3] <- lapply(train_crop[, 1:3] , as.numeric)
test_crop[,1:3] <- lapply(test_crop[, 1:3] , as.numeric)
```

```
## tibble [1,980 x 8] (S3: tbl_df/tbl/data.frame)
##  $ N          : num [1:1980] 90 85 60 74 78 69 69 94 89 68 ...
##  $ P          : num [1:1980] 42 58 55 35 42 37 55 53 54 58 ...
##  $ K          : num [1:1980] 43 41 44 40 42 42 38 40 38 38 ...
##  $ temperature: num [1:1980] 20.9 21.8 23 26.5 20.1 ...
##  $ humidity   : num [1:1980] 82 80.3 82.3 80.2 81.6 ...
##  $ ph         : num [1:1980] 6.5 7.04 7.84 6.98 7.63 ...
##  $ rainfall   : num [1:1980] 203 227 264 243 263 ...
##  $ label      : chr [1:1980] "rice" "rice" "rice" "rice" ...
```

```
## tibble [220 x 8] (S3: tbl_df/tbl/data.frame)
##  $ N          : num [1:220] 78 91 76 83 97 88 60 93 99 65 ...
##  $ P          : num [1:220] 58 35 40 41 59 55 36 56 41 37 ...
##  $ K          : num [1:220] 44 39 43 43 43 45 43 42 36 40 ...
##  $ temperature: num [1:220] 26.8 23.8 25.2 21.1 26.4 ...
##  $ humidity   : num [1:220] 80.9 80.4 83.1 82.7 84 ...
##  $ ph         : num [1:220] 5.11 6.97 5.07 6.25 6.29 ...
##  $ rainfall   : num [1:220] 284 206 231 233 271 ...
##  $ label      : chr [1:220] "rice" "rice" "rice" "rice" ...
```

Macromolecule Variables

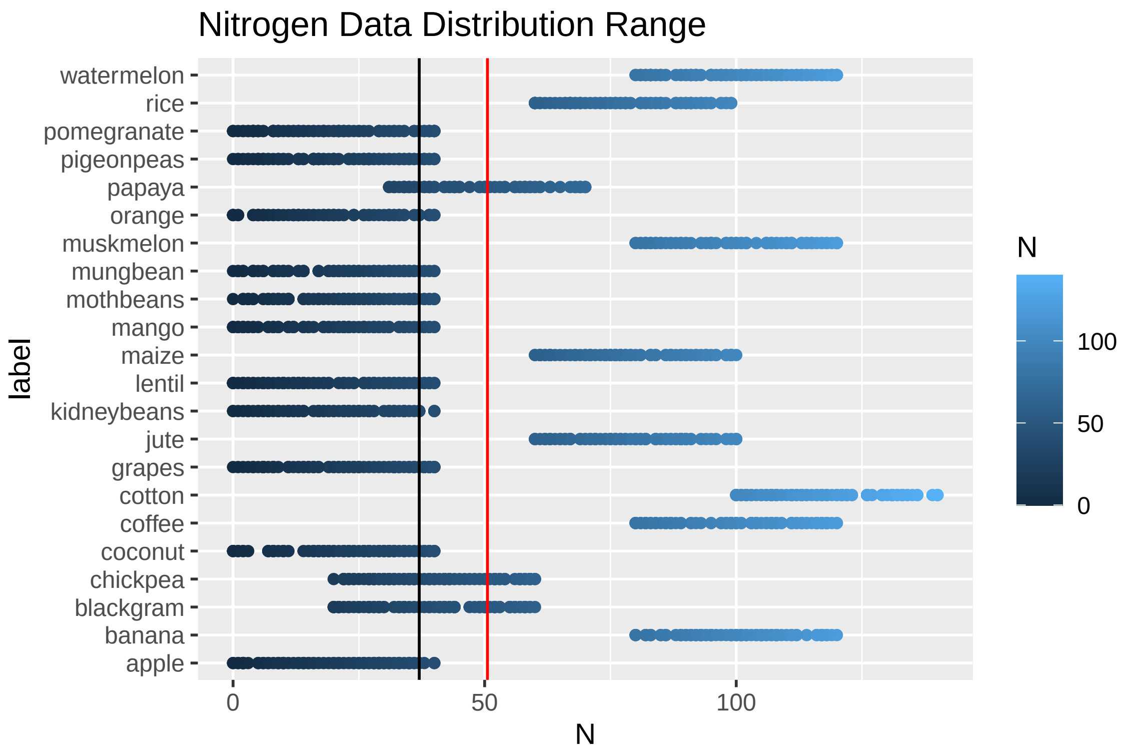
The macromolecules (K, P, N) were examined to understand the number of distinct values (K=73, P=116, N=136). Table 2 shows basic statistics (mean and standard deviation) for each crop.

Table 2: **Basic Macromolecule Statistics for Crops**

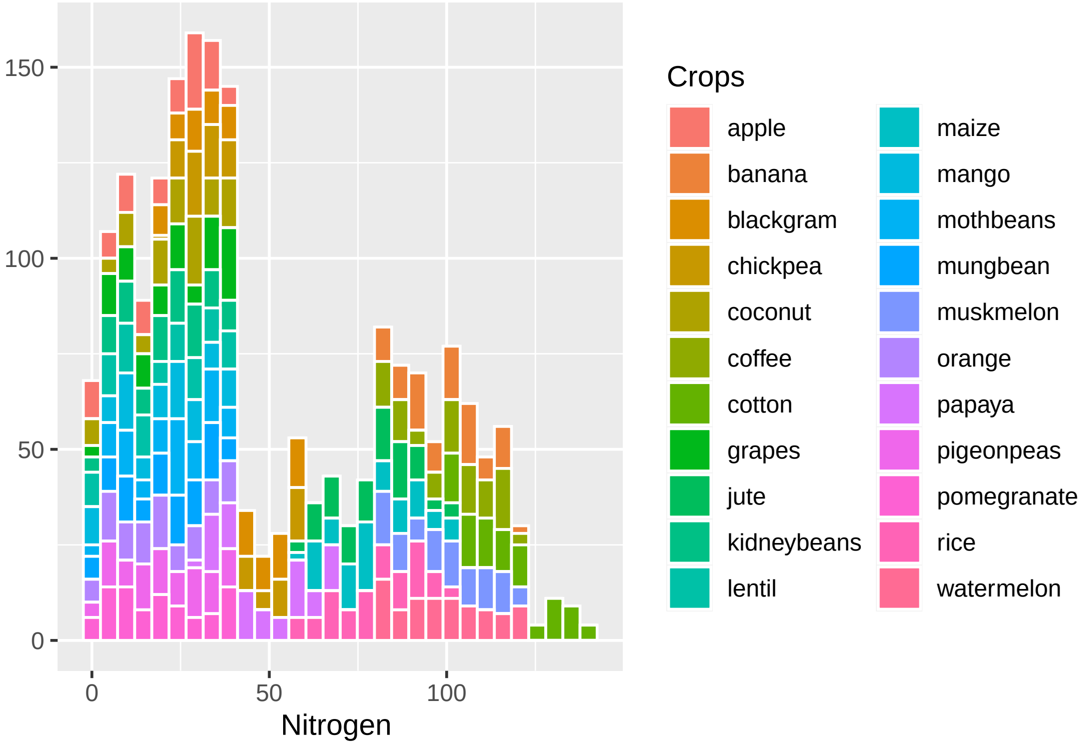
Crops	Mean N	SD N	Mean K	SD K	Mean P	SD P
apple	20.58889	11.80449	199.733333	3.287019	133.91111	8.221163

Crops	Mean N	SD N	Mean K	SD K	Mean P	SD P
banana	99.58889	11.07269	50.044444	3.287399	82.08889	7.635877
blackgram	40.57778	12.46025	19.266667	3.263003	67.43333	7.284044
chickpea	39.78889	11.86771	80.044444	3.287399	67.91111	7.304984
coconut	22.81111	11.52418	30.644444	3.017965	16.73333	8.482368
coffee	101.24444	12.33822	29.966667	3.230647	28.43333	7.357714
cotton	117.54444	11.85508	19.444444	3.201513	46.50000	7.242586
grapes	22.98889	12.46072	200.111111	3.285804	132.41111	7.554885
jute	78.81111	11.09600	39.977778	3.273928	46.87778	7.339026
kidneybeans	20.58889	10.97893	19.955556	3.172604	67.62222	7.873405
lentil	19.05556	12.30727	19.355556	2.946379	68.35556	7.256630
maize	79.00000	11.69462	19.700000	2.965911	48.67778	8.047371
mango	19.61111	12.03338	29.966667	3.048116	27.48889	7.648521
mothbeans	21.77778	11.21373	20.411111	3.097278	48.63333	7.365956
mungbean	20.72222	11.68599	19.755556	3.174021	47.63333	7.708816
muskmelon	99.93333	12.24727	50.133333	3.229777	17.53333	7.156674
orange	19.53333	12.02918	9.811111	3.034981	16.37778	7.765637
papaya	49.42222	12.09604	49.922222	3.127363	58.81111	7.185964
pigeonpeas	20.95556	11.61935	20.255556	2.798720	67.74444	7.375272
pomegranate	19.05556	12.76878	40.422222	2.986779	19.01111	7.275470
rice	79.54444	11.79713	39.655556	2.907414	47.77778	7.694705
watermelon	99.04444	12.55363	50.111111	3.120949	17.14444	7.567763

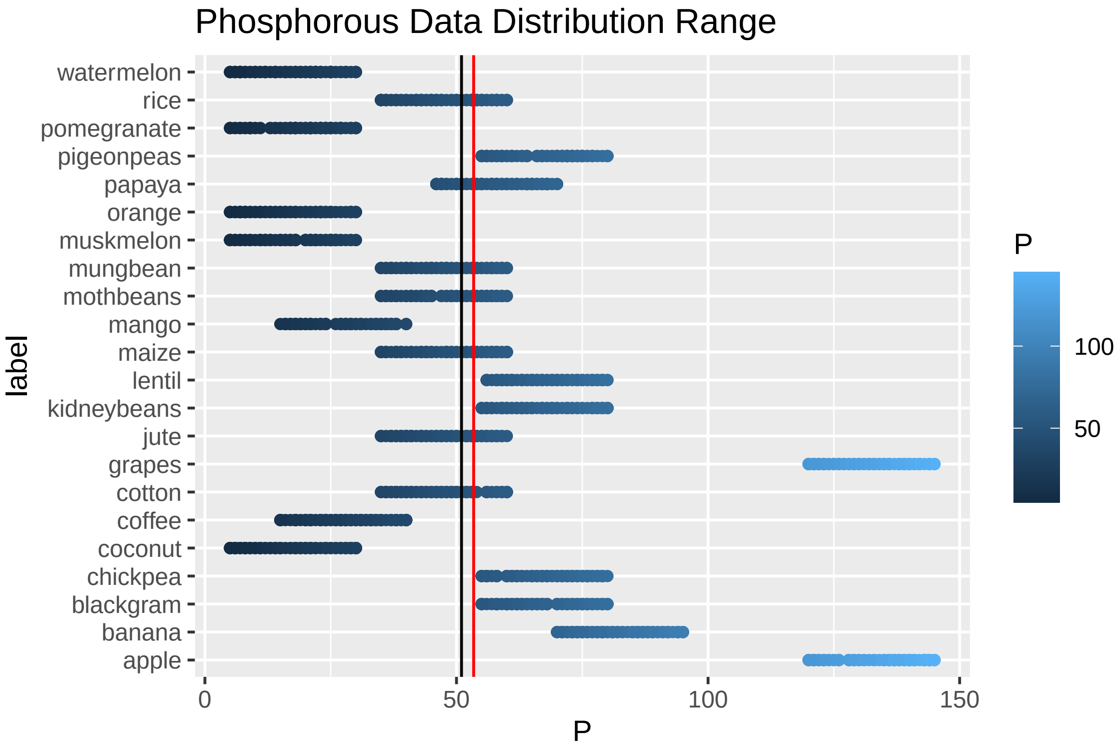
Nitrogen (N): The data distribution for the Nitrogen ratio has a range that extends from zero to 140. All distribution plots have a red vertical line representing the mean (50.55) and a black vertical line representing the median (37.00). Eleven crops have very similar Nitrogen ranges at the lower end of the scale (~0 to 40). A group of four crops have similar nitrogen ranges at the higher end of the range (~80-120). The histogram plot shows the distribution to be bimodal or multimodal for the crop data.



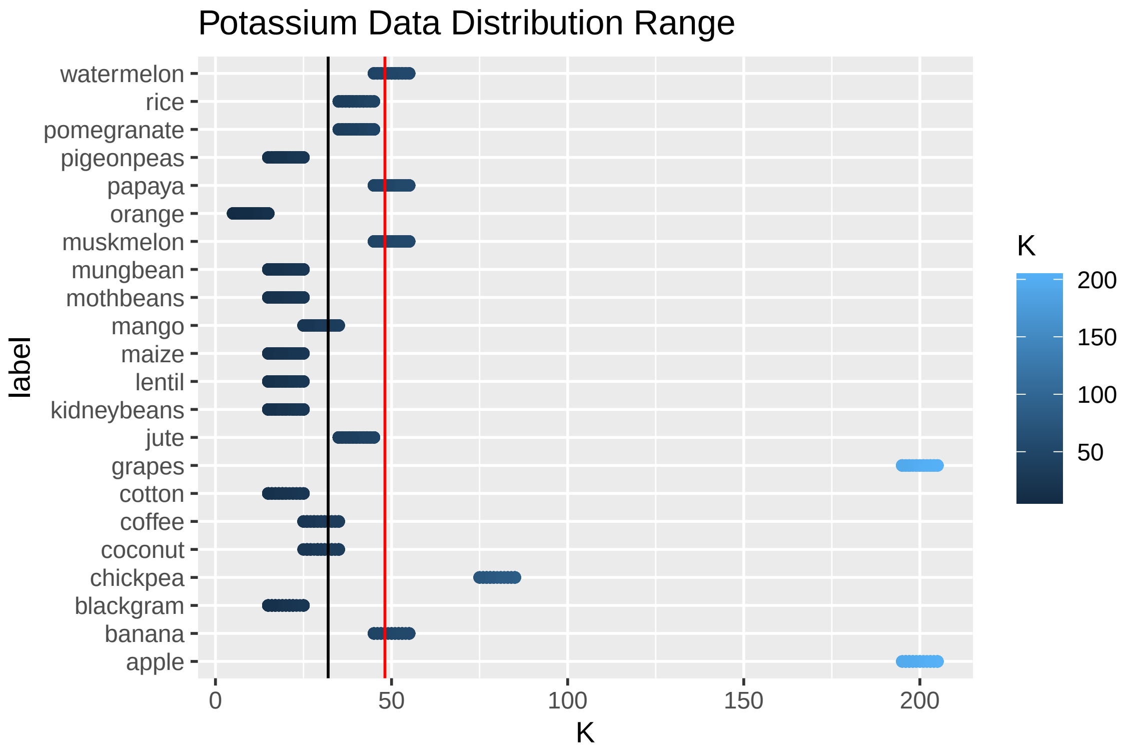
```
## Warning: ‘qplot()‘ was deprecated in ggplot2 3.4.0.
```



Phosphorous: The data distribution for the Phosphorous ratio has a range that extends from 5 to 145 with an overall mean of 53.36 and median of 51.00. A separate group of two crops, apples and grapes, have a high ratio of Phosphorous in the range of ~120-145.



Potassium: The data distribution for the Potassium ratio has a range that extends from 5 to 205 with a mean 48.15 and a median of 32.00. While most of the crops are at the lower end of the range, apples and grapes once again are at the high end of the range, separated from the other crops (~195-205). Chick peas also appear to be separate with a Potassium ration of ~75-85. For this distribution, the mean is greater than the median, skewed left with more observations at the lower end of the range.



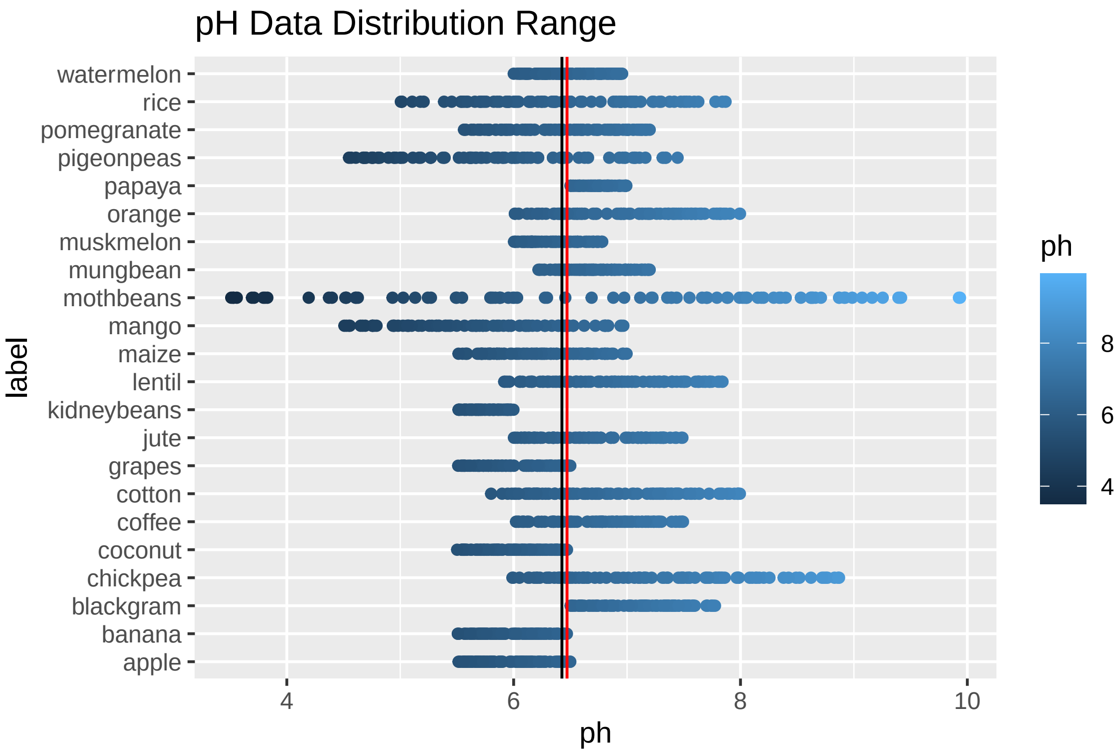
pH

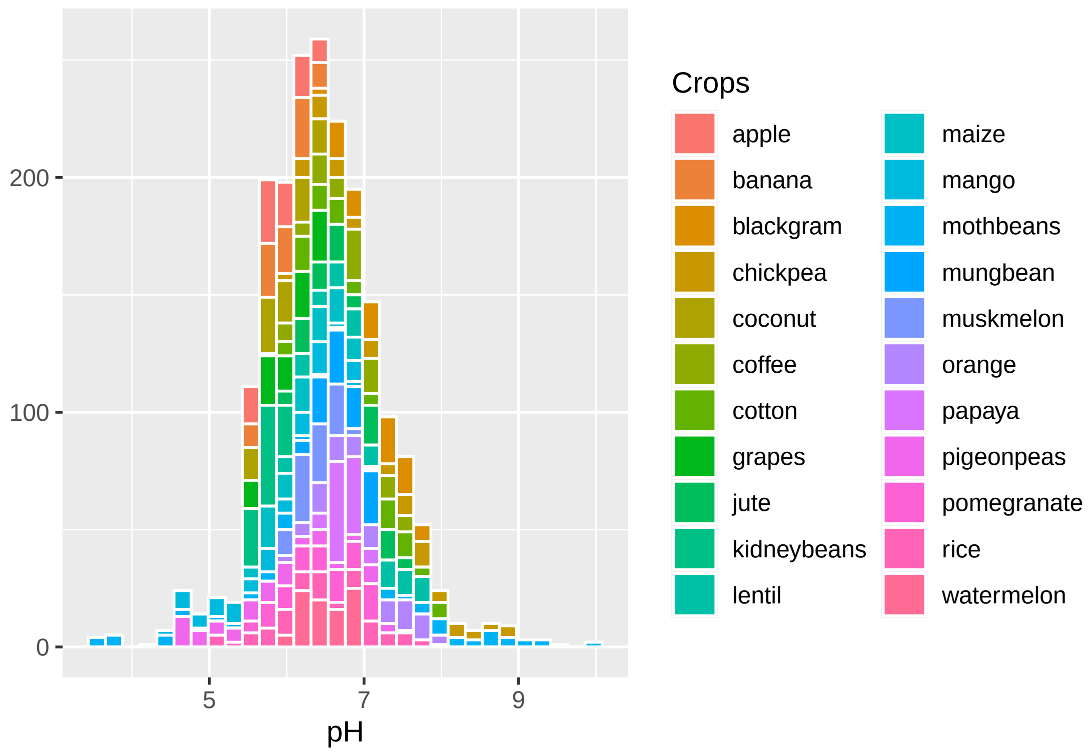
The soil pH has 1760 distinct values. Table 3 shows the average for individual crops ranges from 5.7 to 7.3. The pH data distribution ranges from 3.505 to 9.935. The broad range can be attributed to soil pH values associated with moth beans. If that crop were removed, the range would be tighter (~4.5 to 9). The mean is 6.469 and the median is 6.425, both slightly acidic. The qplot for pH appears to have a normal distribution with some tailing in acidic and basic zones attributed to moth beans.

Table 3: **Soil pH Mean and Standard Deviation for Crops**

Crops	Mean	SD
apple	5.931064	0.2756085
banana	5.989144	0.2663454
blackgram	7.147687	0.3695858
chickpea	7.321862	0.8182747
coconut	5.978307	0.2820854
coffee	6.784907	0.4194565
cotton	6.888537	0.6219646
grapes	6.024111	0.2997908
jute	6.731620	0.4530490
kidneybeans	5.752795	0.1398716
lentil	6.936933	0.5618958

Crops	Mean	SD
maize	6.242535	0.4064660
mango	5.758422	0.7033226
mothbeans	6.794881	1.8370119
mungbean	6.724814	0.2830360
muskmelon	6.362315	0.2329386
orange	7.045575	0.5597606
papaya	6.743937	0.1439225
pigeonpeas	5.811011	0.8303980
pomegranate	6.428547	0.4933839
rice	6.448728	0.7444498
watermelon	6.508859	0.2825410





Climate Variables

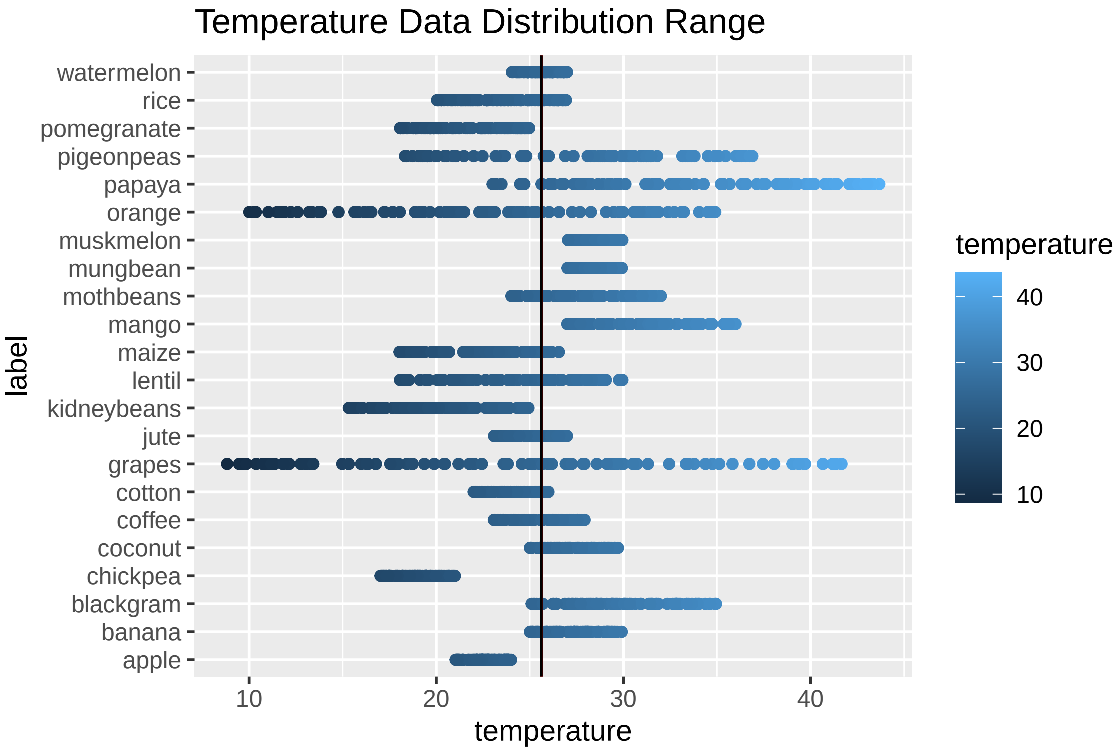
Temperature, humidity, and rainfall all have 1760 distinct values. Table 4 shows the mean and standard deviation for each crop. Considering crops vary in their climate zone requirements and can be seasonal (e.g., Ohio), a variable range across crops is expected. In Ohio, the outdoor planting season starts with spring crops (e.g., strawberries, peas, spinach), summer crops (e.g., tomatoes, peppers, kale), followed by fall crops (e.g., beets, radishes, swiss chard). See the local planting guide (12). Availability of controlled greenhouse conditions extends the season year round and enables controlled nutrient supplementation while leveraging new technologies (e.g., hydroponics, vertical systems).

Table 4: **Basic Statistics for Crops**

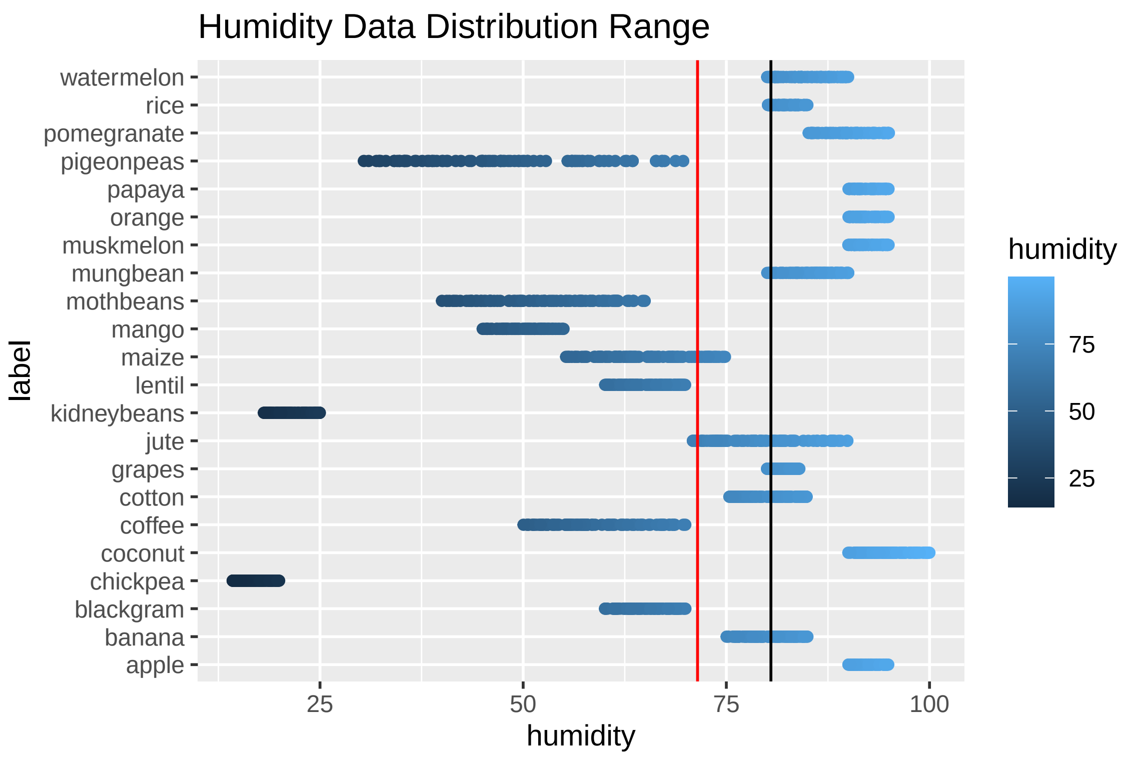
Crops	Mean Temperature (C)	SD Temperature (C)	Mean %RH	SD %RH	Mean Rainfall (mm)	SD Rainfall (mm)
apple	22.64731	0.8351995	92.27922	1.413818	112.87846	6.984500
banana	27.36034	1.4200858	80.41309	2.789918	104.59228	9.277667
blackgram	29.96216	2.7270690	65.16309	2.762307	67.87317	4.277319
chickpea	18.88608	1.1556128	16.82689	1.735169	79.97381	7.982897
coconut	27.33132	1.3486934	94.75629	2.735185	176.89100	29.117610
coffee	25.57031	1.4888879	59.02961	5.780833	158.89979	25.880072
cotton	23.97230	1.1245257	79.89602	3.034259	80.69465	10.938414
grapes	23.78574	9.8316840	81.87723	1.191933	69.63404	2.975099
jute	24.99347	1.1799102	79.44441	5.440926	175.62323	14.754287
kidneybeans	20.21899	2.5928111	21.63812	2.175702	105.95780	25.998290
lentil	24.51625	3.3761942	64.88012	2.990403	45.60133	5.787978
maize	22.29740	2.6740459	64.97150	5.519986	84.49458	15.559854
mango	31.19960	2.6541610	50.08793	2.736512	94.72410	3.388513

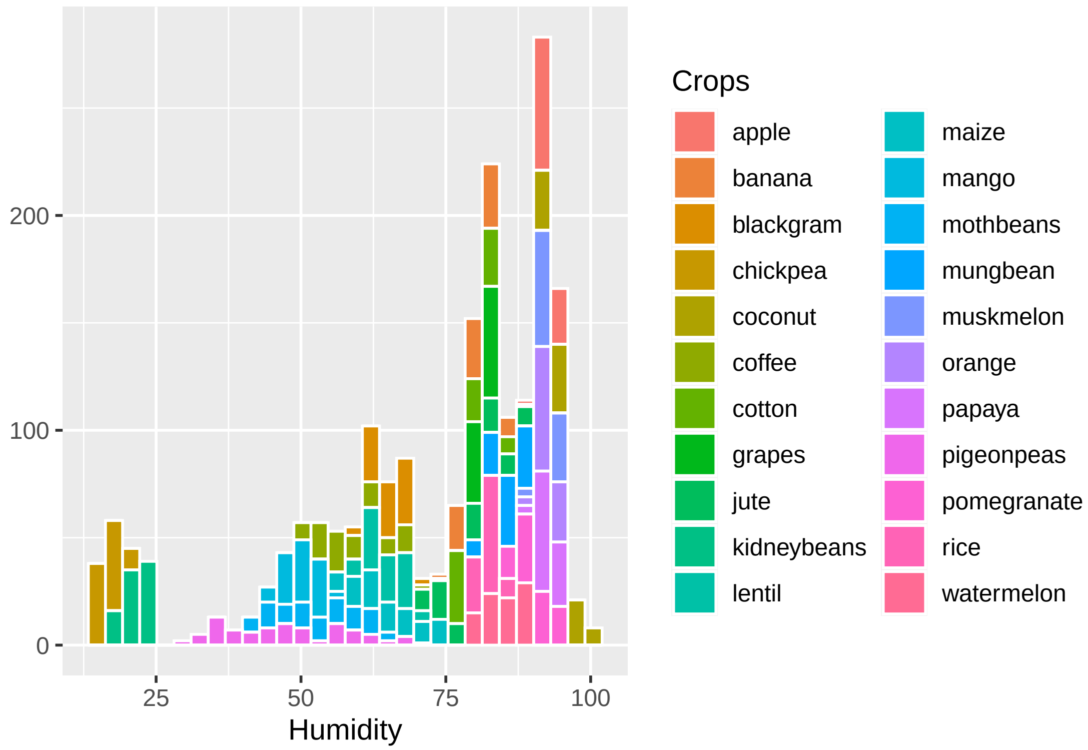
Crops	Mean Temperature (C)	SD Temperature (C)	Mean %RH	SD %RH	Mean Rainfall (mm)	SD Rainfall (mm)
mothbeans	28.20116	2.2476395	52.79526	6.998275	50.76276	13.465518
mungbean	28.54124	0.8448251	85.67034	2.757315	48.49570	7.077422
muskmelon	28.65618	0.8510113	92.42383	1.522767	24.79061	2.792453
orange	22.69461	7.1747368	92.19737	1.425608	110.18913	5.774343
papaya	34.29234	6.1890590	92.39429	1.448752	142.00142	65.151358
pigeonpeas	27.59501	5.7002189	47.74904	10.790896	148.13705	33.588570
pomegranate	21.70592	2.2031778	90.21162	2.825099	107.74981	2.874047
rice	23.62254	2.0675672	82.26768	1.436808	237.24298	34.250305
watermelon	25.60090	0.8199527	85.05605	3.016837	51.09248	5.913776

Temperature: The temperature data distribution ranges from 8.826C (48F) to 43.675C (111F). Pigeon peas, oranges, papaya, and grapes have a fairly wide temperature range for growth based on this data. The data set also shows 10 crops with a narrow temperature range (watermelon, muskmelon, mung bean, jute, cotton, coffee, coconut, chickpeas, bananas and apples). Let's look at one of these crops, apples. The temperature range for apples appears to be approximately 21C-24C (70F-75F). In an area with seasonal changes where apples grow well; such as Ohio, the temperature range is much broader than the range in the data supporting the need for understanding the impact of location and seasonal effects.

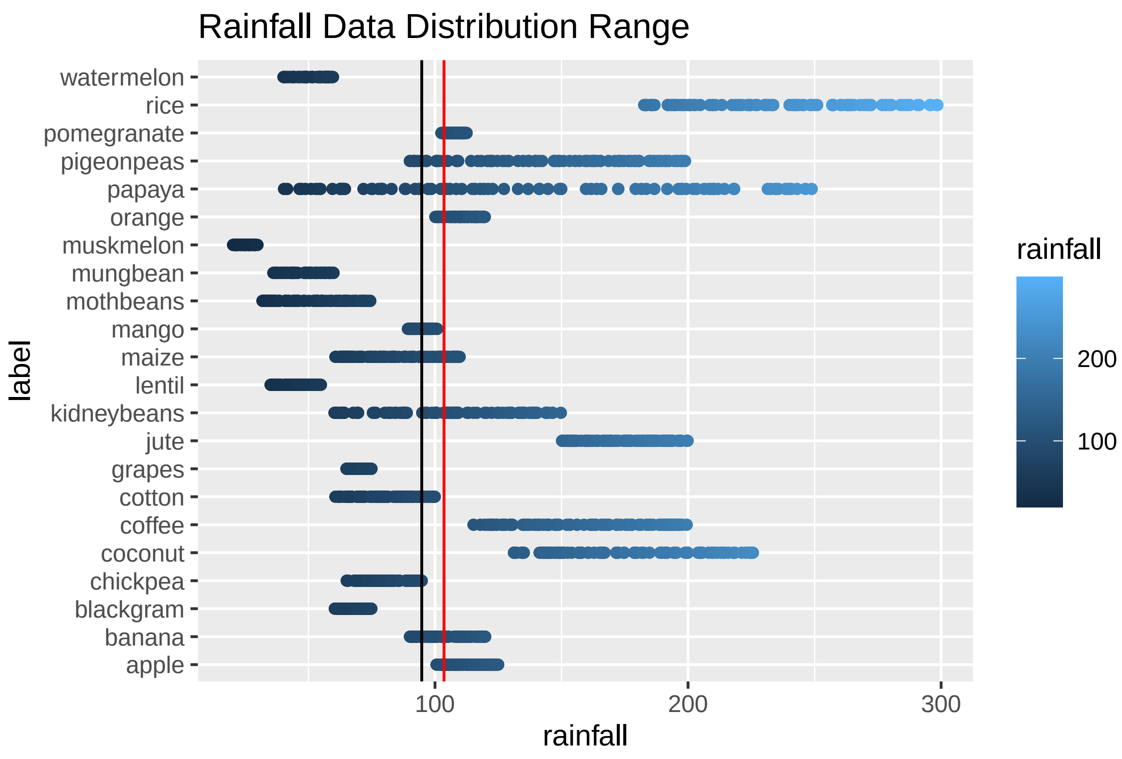


Humidity: The humidity data distribution ranges from 14.26% RH to 99.98% RH. The humidity data for Pigeon peas has the largest range. Some crops have humidity data reported with very tight ranges; similar to the temperature data, supporting the need to understand location and seasonality effects. The qplot for humidity indicates a multimodal distribution.





Rainfall: The rainfall data distribution ranges from 20.21 mm (0.8”) to 298.56 mm (11.8”). Rice has the highest rainfall reported for the crops.



Model Development

Model development for this recommendation system began with a Random Forest model (Model 1), which can be used for classification or regression, followed by an updated Random Forest model (Model 2) with some variables removed (6,13, 14). A correlation of the predictor variables was performed followed by Principal Component Analysis and Multinomial Regression.

Model 1 Random Forest

The first model selected to examine for model development was Random Forest using all seven predictor variables. The label column representing the crops was converted to a factor in both the training and test data sets. Ten-fold cross validation was used for the training control. The highest accuracy (99.5%) was achieved using the training set with an mtry value of 2. When the model was checked with the test data, an accuracy of 99.1% was achieved.

```
# Set the K-fold Cross Validation using the trainControl() function
control_rf <- trainControl(method = "cv", # resample data
                            number = 10,   # folds
                            search = "grid")
```

```
# Convert label to a factor in train and test data sets
train_crop2 <- train_crop
train_crop2$label <- factor(train_crop$label)

test_crop2 <- test_crop
test_crop2$label <- factor(test_crop$label)
```

```r
# Base Random Forest model using all 7 predictors
# Train model
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```r
rf_model <- train(label~.,
                  data = train_crop2,
                  method = "rf",
                  control_rf = control_rf,
                  ntree = 100,
                  metric = "Accuracy")


print(rf_model)
```

```
## Random Forest
##
## 1980 samples
##    7 predictor
##   22 classes: 'apple', 'banana', 'blackgram', 'chickpea', 'coconut', 'coffee', 'cotton', 'grapes', '
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1980, 1980, 1980, 1980, 1980, 1980, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.9946499  0.9943899
##   4     0.9928434  0.9924956
##   7     0.9896653  0.9891631
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```r
rf_model$bestTune
```

```
##   mtry
## 1    2
```

Check Accuracy - Test Data

```r
rf_preds <- predict(rf_model, test_crop2)
mean(rf_preds == test_crop2$label)
```

```
## [1] 0.9909091
```

Model 2 Random Forest - Reduced Variables

The importance of predictors for Model 1 was checked using the varImp() function. The top four variables based on importance (rainfall, humidity, K, and P) were used for Model 2 to understand the impact of removing three variables. The highest accuracy for Model 2 was (97.3%) when using the training set with an mtry value of 2. The model was checked for accuracy using the test data and accuracy improved to 100%.

```
varImp(rf_model)
```

```
## rf variable importance
##
##             Overall
## rainfall     100.00
## humidity      93.44
## K             76.76
## P             55.35
## N             30.80
## temperature   12.98
## ph             0.00
```

```
# The top 4 variables were selected for both the train and test data set.
train_crop3 <- train_crop2 |> select(rainfall, humidity, K, P, label)
test_crop3 <- train_crop2 |> select(rainfall, humidity, K, P, label)
```

```
# Train Model 2
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
rf_model2 <- train(label~.,
                  data = train_crop3,
                  method = "rf",
                  control_rf = control_rf,
                  ntree = 100,
                  metric = "Accuracy")


print(rf_model2)
```

```
## Random Forest
##
## 1980 samples
##    4 predictor
##   22 classes: 'apple', 'banana', 'blackgram', 'chickpea', 'coconut', 'coffee', 'cotton', 'grapes', '
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1980, 1980, 1980, 1980, 1980, 1980, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.9728033  0.9714819
##   3     0.9705792  0.9691502
##   4     0.9686585  0.9671359
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```
rf_model2$bestTune
```

```
##   mtry
## 1    2
```

Check Accuracy of Model 2 - Test Data

```
rf_preds2 <- predict(rf_model2, test_crop3)
mean(rf_preds2 == test_crop3$label)
```

```
## [1] 1
```

Model 3 Principal Component Analysis (PCA) Model Development

Looking at the Random Forest models was fairly routine. With this type of data, It was important to understand correlation of the predictor variables to see how the variables relate to each other and if dimension reduction was possible. References available for correlation plots were very informative and useful (11, 15).

PCA is used for dimension reduction and keeping a high level of variance. Ideally a data set for crop recommendation would have many more variables (additional macro and micronutrients, location, historical data (time based), seasonality, length of daylight, etc.). Many references were used to understand how to approach, visualize, and evaluate the principal components (10, 11, 16)

Check Correlation of Independent Variables

Checking the correlation of independent variables provided further insight into potential issues such as multicollinearity and understanding relationships between the predictor variables. The train_crop data set was normalized by scaling and then used to create a correlation matrix using only the predictor variables in the first 7 columns. ggcorrplot() was used to visualize the matrix.

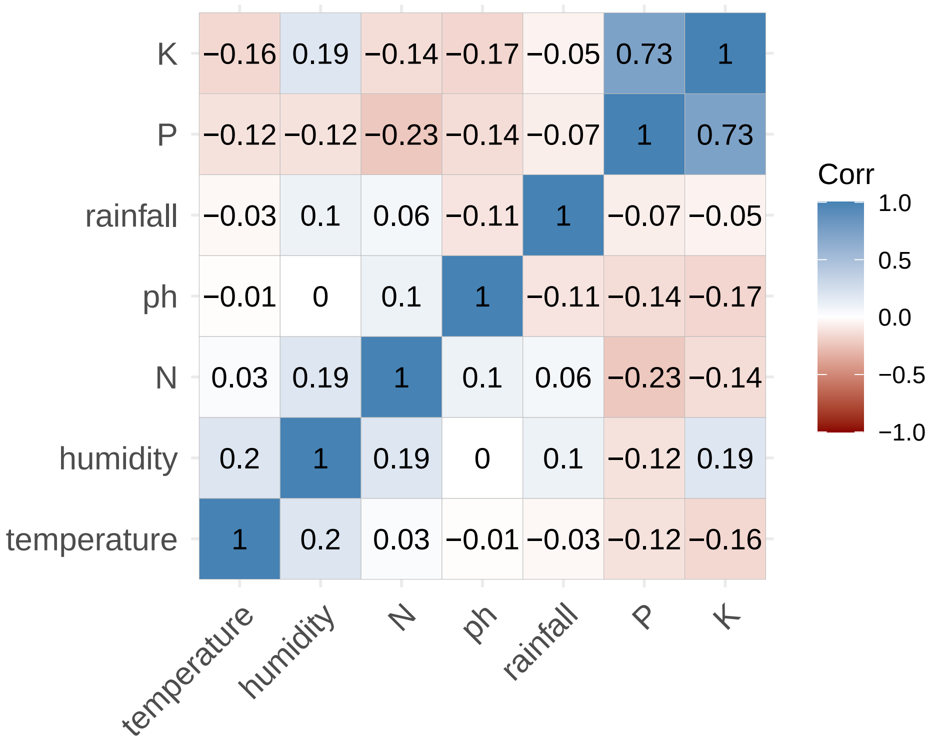
```
# Normalize numeric data in the data set using scaling (columns 1:7)
train_numeric <- train_crop[,1:7]

train_normal <- scale(train_numeric)
head(train_normal)
```

```
##              N           P           K temperature  humidity         ph
## [1,] 1.0711742 -0.34653415 -0.10114299  -0.9295301 0.4731776 0.04190168
## [2,] 0.9353968  0.13922699 -0.14063483  -0.7548854 0.3976666 0.73756702
## [3,] 0.2565096  0.04814678 -0.08139707  -0.5129333 0.4874453 1.78034243
## [4,] 0.6366865 -0.55905465 -0.16038076   0.1706978 0.3904308 0.66256067
## [5,] 0.7453084 -0.34653415 -0.12088891  -1.0764993 0.4553274 1.50507962
## [6,] 0.5009090 -0.49833451 -0.12088891  -0.5024259 0.5345237 0.78353269
##      rainfall
## [1,] 1.800297
## [2,] 2.230007
## [3,] 2.905889
## [4,] 2.523639
## [5,] 2.883300
## [6,] 2.672026
```

```
# Correlation matrix
train_corr_matrix <- cor(train_normal)
```

The correlation can be interpreted by higher positive values having a higher correlation and the negative values closest to -1.0 are the most negatively correlated. Analysis was first run using both princomp() and prcomp(). Based on the site Statistical Tools for High-throughput Data Analysis (STHDA), prcomp() was selected (16). The correlation matrix below shows that P and K are highly correlated. Consideration could be given to removing one of these feature variable, P or K.

```
ggcorrplot(train_corr_matrix, hc.order = TRUE, lab = TRUE, colors = c("darkred", "white", "steelblue"))
```



	temperature	humidity	N	ph	rainfall	P	K
K	−0.16	0.19	−0.14	−0.17	−0.05	0.73	1
P	−0.12	−0.12	−0.23	−0.14	−0.07	1	0.73
rainfall	−0.03	0.1	0.06	−0.11	1	−0.07	−0.05
ph	−0.01	0	0.1	1	−0.11	−0.14	−0.17
N	0.03	0.19	1	0.1	0.06	−0.23	−0.14
humidity	0.2	1	0.19	0	0.1	−0.12	0.19
temperature	1	0.2	0.03	−0.01	−0.03	−0.12	−0.16

PCA

The PCA analysis was performed by transforming the variables to components. The summary indicates the importance of components showing the first five components account for 87.6% of the variance. This is also observed with the Scree plot. The relationship of the variables to the components is shown using the print() function. For example, for PC1 there is a high correlation for P and K. The biplot for PC1 and PC2 also provides insight to the relationships of the components to the variables.

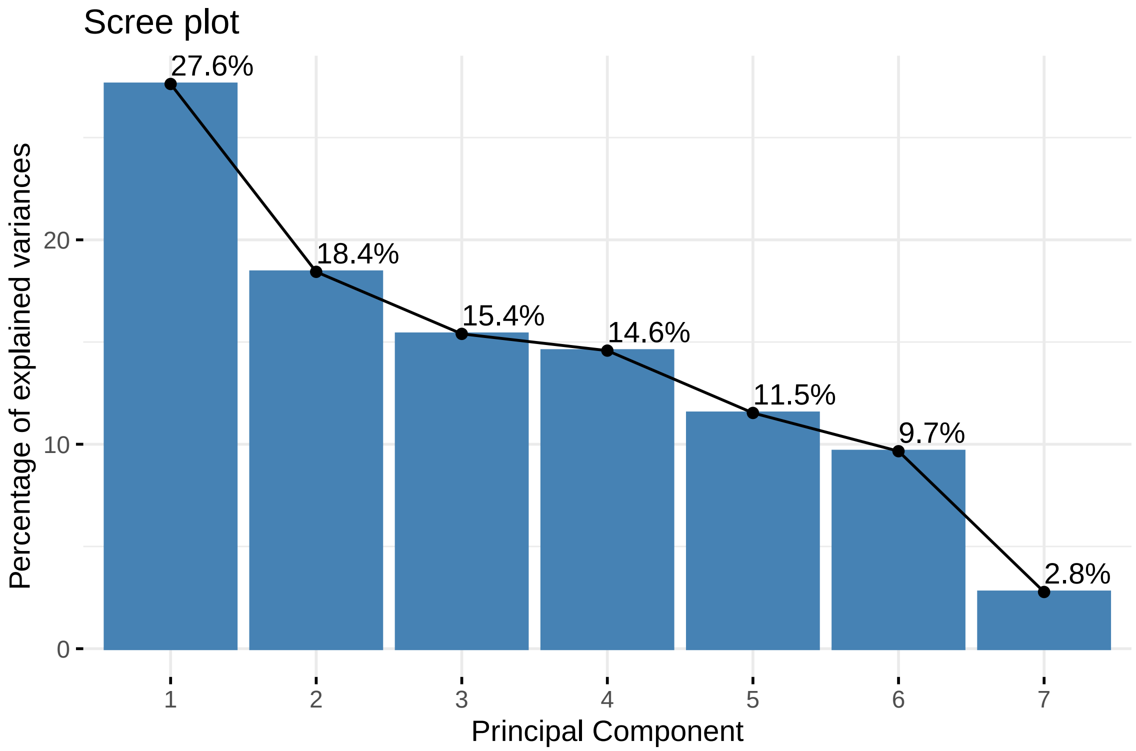
```
train_pca <- prcomp(train_crop[,1:7], center = TRUE, scale = TRUE)
summary(train_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     1.3905 1.1359 1.0382 1.0103 0.8984 0.82234 0.44069
## Proportion of Variance 0.2762 0.1843 0.1540 0.1458 0.1153 0.09661 0.02774
## Cumulative Proportion  0.2762 0.4606 0.6145 0.7603 0.8757 0.97226 1.00000
```

```
print(train_pca)
```

```
## Standard deviations (1, .., p=7):
## [1] 1.3905262 1.1359289 1.0382183 1.0102537 0.8984160 0.8223355 0.4406878
##
## Rotation (n x k) = (7 x 7):
##                     PC1        PC2         PC3         PC4         PC5
## N           -0.30740435  0.3389451  0.04021148  0.53515482 -0.51808628
## P            0.64224033  0.0404147  0.10669056  0.06176843  0.07916576
## K            0.62012366  0.2943189  0.13872247  0.17049133  0.03023384
## temperature -0.21055495  0.3504265  0.32410652 -0.66756912  0.15219647
## humidity    -0.07380584  0.7421889  0.19702576  0.08071211  0.12087907
## ph          -0.22948834 -0.2027827  0.51238423  0.46321309  0.64103321
## rainfall    -0.07765940  0.2865228 -0.74923591  0.11831040  0.52507734
##                     PC6          PC7
## N            0.48390732  0.005607043
## P            0.37974878  0.648243351
## K            0.02809808 -0.691959720
## temperature  0.49637602 -0.112344930
## humidity    -0.54633871  0.292022238
## ph           0.12820060 -0.041849004
## rainfall     0.24397649 -0.035932653
```

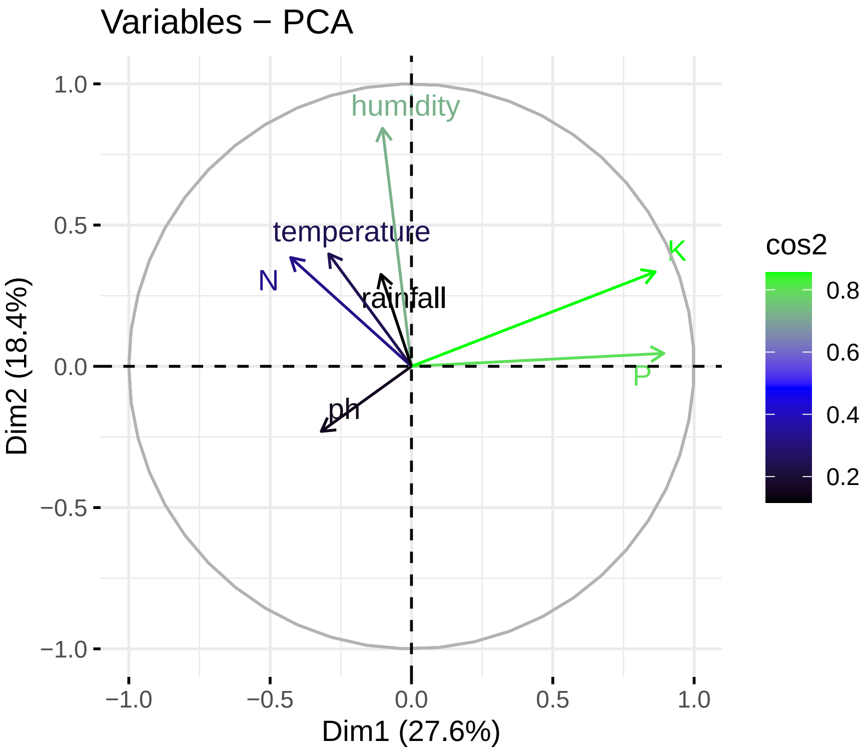
```
fviz_eig(train_pca, addlabels = TRUE) +
  xlab("Principal Component")
```



Bi-Plot

The bi-plot below visualizes attribute similarities and their relative importance (cos2) for the first 2 components. In a bi-plot, Variables grouped together are positively correlated to each other (e.g. rainfall, humidity, temperature, Nitrogen) and the higher distance from the center (origin) to the variable, the better the variable is represented. Some insight from the bi-plot are listed below. 1. Humidity is better represented than rainfall, temperature, and Nitrogen and has higher level of relative importance. 2. P and K are correlated to each other and have a high level of relative importance. 3. PC1 (Dim1) is positively correlated with K and P. 4. P and K are weakly correlated to the other variables. 5. pH is weakly correlated to other variables. 6. PC2 is positively correlated with all variables except pH

```
fviz_pca_var(train_pca, col.var = "cos2",
             gradient.cols = c("black", "blue", "green"),
             repel = TRUE)
```

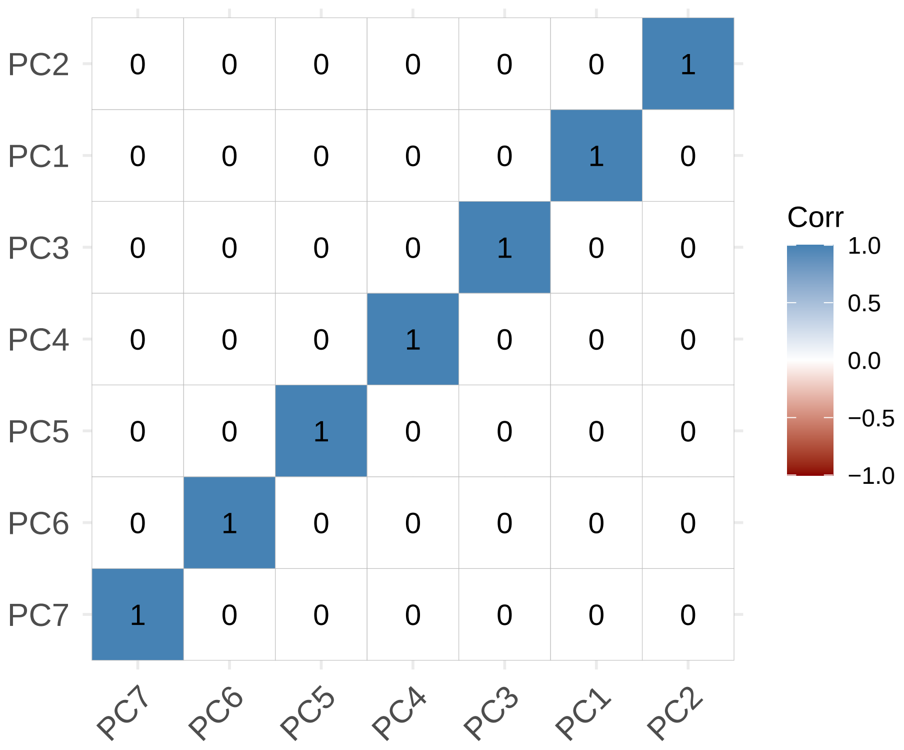


The next step was to check if multicolinearity was addressed. The correlation data of the compoents is plotted below. PCA successfully transforms the data into uncorrelated variables (18).

```
# Re-check correlation
check <-cor(train_pca$x)
check
```

```
##               PC1           PC2           PC3           PC4           PC5
## PC1  1.000000e+00  5.359602e-16  3.775393e-16  2.335361e-16 -5.129863e-16
## PC2  5.359602e-16  1.000000e+00 -1.441743e-16 -2.524904e-16 -9.281826e-17
## PC3  3.775393e-16 -1.441743e-16  1.000000e+00  8.970122e-16  9.199415e-16
## PC4  2.335361e-16 -2.524904e-16  8.970122e-16  1.000000e+00  1.534690e-15
## PC5 -5.129863e-16 -9.281826e-17  9.199415e-16  1.534690e-15  1.000000e+00
## PC6  1.591629e-15  1.959768e-16  6.153380e-16 -9.059401e-16 -3.255249e-15
## PC7  1.913279e-16 -1.054920e-16  4.614920e-16 -9.111623e-16 -6.245876e-17
##               PC6           PC7
## PC1  1.591629e-15  1.913279e-16
## PC2  1.959768e-16 -1.054920e-16
## PC3  6.153380e-16  4.614920e-16
## PC4 -9.059401e-16 -9.111623e-16
## PC5 -3.255249e-15 -6.245876e-17
## PC6  1.000000e+00  1.424143e-15
## PC7  1.424143e-15  1.000000e+00
```

```
ggcorrplot(check, hc.order = TRUE, lab = TRUE, colors = c("darkred", "white", "steelblue"))
```



Multinomial Regression using Principal Components for Prediction

Mulitnomial logistic regression was performed with the first 5 principal components (17). The crop data was added back into both the train and test data sets. The nnet() library was used to perform the analysis. Apple was selected for the reference since selected values separated it from a majority of the crops. The model was trained with 100 iterations. Coefficients are listed for each PCA used and crop. The test set was used with the model and misclassification error calculated (9.5%). Changing by data split for the the test set from 20% to 10% reduced the misclassification error from 50% (data not shown) to 9.5%.

```
# Add crop information back into the data sets
train <- predict(train_pca, train_crop)
train <- data.frame(train, train_crop[8])
test <- predict(train_pca, test_crop)
test <- data.frame(test, test_crop[8])
```

```
library(nnet)
train$label <- factor(train$label, ordered=FALSE)
train$label <- relevel(train$label, ref = "apple")
multinomial_model <- multinom(label~PC1+PC2+PC3+PC4+PC5, data = train)
```

```
## # weights:  154 (126 variable)
## initial  value 6120.264058
## iter  10 value 1990.754183
## iter  20 value 1587.756685
## iter  30 value 1163.612867
## iter  40 value 737.579147
## iter  50 value 455.502477
## iter  60 value 320.868821
## iter  70 value 269.687910
## iter  80 value 236.886865
## iter  90 value 219.558923
## iter 100 value 207.457448
## final  value 207.457448
## stopped after 100 iterations
```

```
summary(multinomial_model)
```

```
## Call:
## multinom(formula = label ~ PC1 + PC2 + PC3 + PC4 + PC5, data = train)
##
## Coefficients:
##              (Intercept)         PC1         PC2        PC3         PC4
## banana        -4.6466898  -24.905483   26.470248   9.238201  21.0718661
## blackgram     37.0561565  -23.420387  -81.852035  19.458111 -20.2116075
## chickpea     -57.9788304    0.104323 -122.402370   9.792966  16.6285438
## coconut        8.4158012  -73.204092  -50.174907 -30.788497 -12.4805104
## coffee       -33.0136441 -131.213896  -65.870427 -28.417434   3.3590716
## cotton       -19.2768901  -67.374049  -36.191860  19.027294  30.4903662
## grapes       -79.2747845   21.160017  -29.981026  86.770770   2.0631263
## jute          21.4752220  -49.230894  -12.966481  -7.078913  18.4718747
## kidneybeans -144.9908085  -36.606100 -173.532738 -30.745987 -56.3258190
## lentil        13.3637996  -12.597040  -99.653948  31.304062 -23.9512772
## maize         38.3192038  -56.759390  -65.657311   5.817009   0.4575307
## mango         15.3890354  -40.520338  -99.061225 -14.505691 -36.2523393
## mothbeans     -0.3428356  -39.964981 -115.231192  20.105845 -32.8573075
## mungbean      36.3006610  -23.503100  -75.586872  21.541800 -21.3575213
## muskmelon   -147.2877597  -49.507121  -63.370945 124.262810   0.9096107
## orange        16.2020863  -84.666581  -73.331252 -26.717785 -12.2542212
## papaya        23.7006064  -11.956536    4.754266  11.053588   2.3844511
## pigeonpeas    21.7523105  -19.602333  -92.501568 -18.785019 -34.6678912
## pomegranate   44.9524309  -54.225908  -59.582075 -11.091105  -7.6037158
## rice           4.6318384  -48.209178   -6.983412 -14.008433  21.2080832
## watermelon   -40.7128795  -70.692971  -30.455312  46.169880  13.4506745
##                     PC5
## banana       -41.734939
## blackgram    -37.533368
## chickpea     -39.823869
## coconut      -10.490258
## coffee       -44.439784
## cotton       -74.200137
## grapes       -83.119849
## jute         -20.736169
## kidneybeans  -26.942756
## lentil       -47.426703
## maize        -55.398977
## mango        -10.825142
## mothbeans    -45.836066
## mungbean     -38.960504
## muskmelon   -139.729563
## orange       -16.997433
## papaya         1.330481
## pigeonpeas    -1.553600
## pomegranate  -22.417812
## rice         -17.617224
## watermelon   -85.030010
##
## Std. Errors:
##             (Intercept)       PC1      PC2       PC3       PC4       PC5
## banana         88.06312  25.41659 90.05692  36.63007  62.78015  65.84227
## blackgram      16.82340  22.30711 54.54376  32.23891  57.49844  36.66341
## chickpea      138.35625 161.41254 50.53714  82.81914 249.60202 149.85891
## coconut        18.03737  23.84520 53.08226  32.61176  57.04383  35.98685
## coffee         24.86878  30.03891 53.35201  32.56081  57.02576  36.99001
## cotton         38.87497  27.93758 57.39047  33.99829  59.49167  42.06558
## grapes         15.96868  28.90634 30.05505  23.43170  57.45343  20.19331
## jute           17.04991  22.72316 51.90718  31.98355  56.88993  35.96259
## kidneybeans    71.82875  51.47532 63.40633  33.76840  59.88074  39.90347
## lentil         17.30464  22.63974 54.93102  32.42813  57.53764  36.82095
## maize          16.45061  23.05508 53.82477  32.12151  57.08121  37.58778
## mango          16.98699  23.18746 54.82982  32.31054  57.55404  36.11481
## mothbeans      17.72160  22.90991 55.17066  32.43057  57.54183  36.82762
## mungbean       16.84136  22.26963 54.52335  32.22230  57.49867  36.65351
## muskmelon     181.59830  54.22496 88.24038 138.02469  67.71607 101.70380
## orange         17.24033  23.78638 53.56240  32.44864  57.01089  36.01638
## papaya         17.19889  20.87082 51.46430  31.44468  56.63637  35.47737
## pigeonpeas     16.64619  24.03430 54.80889  32.22936  57.52431  36.13509
## pomegranate    16.31742  22.87126 53.41758  32.17005  56.98035  36.05662
## rice           17.34122  22.72468 51.93377  31.99598  56.90322  35.94024
## watermelon     40.53967  28.47664 57.71728  35.47611  59.26866  42.64592
##
## Residual Deviance: 414.9149
## AIC: 666.9149
```

```
multinomial_pred <- predict(multinomial_model, test)
multinomial_table <- table(multinomial_pred, test$label)
```

```
#error associated with the test set
multinomial_model_error <- 1 - sum(diag(multinomial_table)/sum(multinomial_table))
multinomial_model_error
```

```
## [1] 0.09545455
```

Results

Three models were evaluated for the crop recommendation system. Model 1 was a Random Forest model using all seven predictive variables for the analysis. The best mtr was 2 and resulted in an accuracy of 99.1% which seemed high for my first attempt but the data was not that complex with only 7 predictors. varImp was used to identify the top variables. The top four variables were selected for Model 2, an updated version of the first Random Forest model with fewer predictors. Both models had a high level of accuracy and have potential for use in a crop recommendation system.

PCA was selected a a method to evaluate for down selection of variables. Although the data set used only had seven predictors, there were 22 possible outcomes. The correlation matrix provided a good understand of how independent variables correlated. In that matrix, only P and K were highly correlated indicating one could be removed. The PCA analysis showed the first five components were required to achieve a variance over 87%. Multinomial logistic regression was used for prediction and the misclassification error was 9.5%. Only two variables were removed using PCA. A larger data set with more predictor variables may work better using this type of analysis. Other models could be evaluated as well (e.g. Naive Bayes, Support Vector Machine).

Conclusions

The data set used was basic and had limitations. The data was from India so predictor variable ranges in the data set may not be applicable to other locations. Additional data on seasonality, more nutrient components (e.g., micronutrients), daylight, crop variety, soil type,and incorporation of region collected (location) would also be useful for model development. In order to continue model development, I would look deeper into data collection for this data set. Identification and analysis of data with with U.S. zones, relevant crops, all macro- and micro- nutrients, climate, and seasonality, and additional crop variety would be ideal for my use. Nutrient recommendations would be interesting to tackle as well (e.g., N-P-K ratios) along with troubling shooting growth issues.

References

1. https://www.kaggle.com/datasets/siddharthss/crop-recommendation-dataset
2. https://www.kaggle.com/datasets/siddharthss/crop-recommendation-dataset?select=Crop_recommendation.csv)
3. https://creativecommons.org/licenses/by/3.0/igo/
4. https://pipingpotcurry.com/moth-dal-matki-curry/
5. https://www.usgs.gov/media/images/ph-scale-0#:~:text=The%20range%20goes%20from%200,than%207%20indicates%20a%20base.
6. http://rafalab.dfci.harvard.edu/dsbook/
7. https://www.sciencedirect.com/science/article/pii/S1877050922007293
8. https://www.geeksforgeeks.org/random-forest-approach-for-classification-in-r-programming/
9. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9412477/
10. https://www.r-bloggers.com/2021/05/principal-component-analysis-pca-in-r/
11. https://www.datacamp.com/tutorial/pca-analysis-r
12. https://www.almanac.com/gardening/planting-calendar/OH/Columbus
13. https://www.r-bloggers.com/2021/04/random-forest-in-r/
14. https://www.geeksforgeeks.org/random-forest-approach-for-classification-in-r-programming/
15. https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html
16. http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/118-principal-component-analysis-in-r-prcomp-vs-princomp/#:~:text=The%20function%20princomp()%20uses,preferred%20compared%20to%20princomp().
17. https://www.r-bloggers.com/2020/05/multinomial-logistic-regression-with-r/