Multiclass Crop Type Classification for Smart Farming

Zainab Hanjra

2025-05-03

Abstract

This project explores the application of machine learning for crop type classification in the context of smart farming, using the Smart Farming 2024 (SF24) dataset. As agriculture faces increasing demands for efficiency and sustainability, precision farming powered by data analytics offers a transformative solution. The dataset comprises 2,200 observations across 23 features including soil nutrients, weather conditions, irrigation practices, and more, with the goal of accurately classifying 21 distinct crop types grown in California. Following exploratory data analysis and preprocessing in R, we trained and evaluated seven supervised learning models: Logistic Regression, LDA, QDA, KNN, SVM, Neural Network, and Random Forest. Data was split into training and test sets, scaled appropriately, and categorical variables were converted for modeling compatibility. Among all models, Random Forest achieved the highest classification accuracy (99.55%), significantly outperforming others like KNN (52.27%) and SVM (88.78%). Feature importance analysis revealed that rainfall, humidity, and potassium were the most influential predictors. Retraining the Random Forest model with selected features confirmed its robustness, with only a marginal accuracy increase to 99.69%. These findings demonstrate that ensemble methods like Random Forest are highly effective for complex, multiclass agricultural datasets, offering both predictive power and interpretability. The results underscore the potential of data-driven approaches in optimizing crop management and advancing the goals of smart agriculture.

Introduction

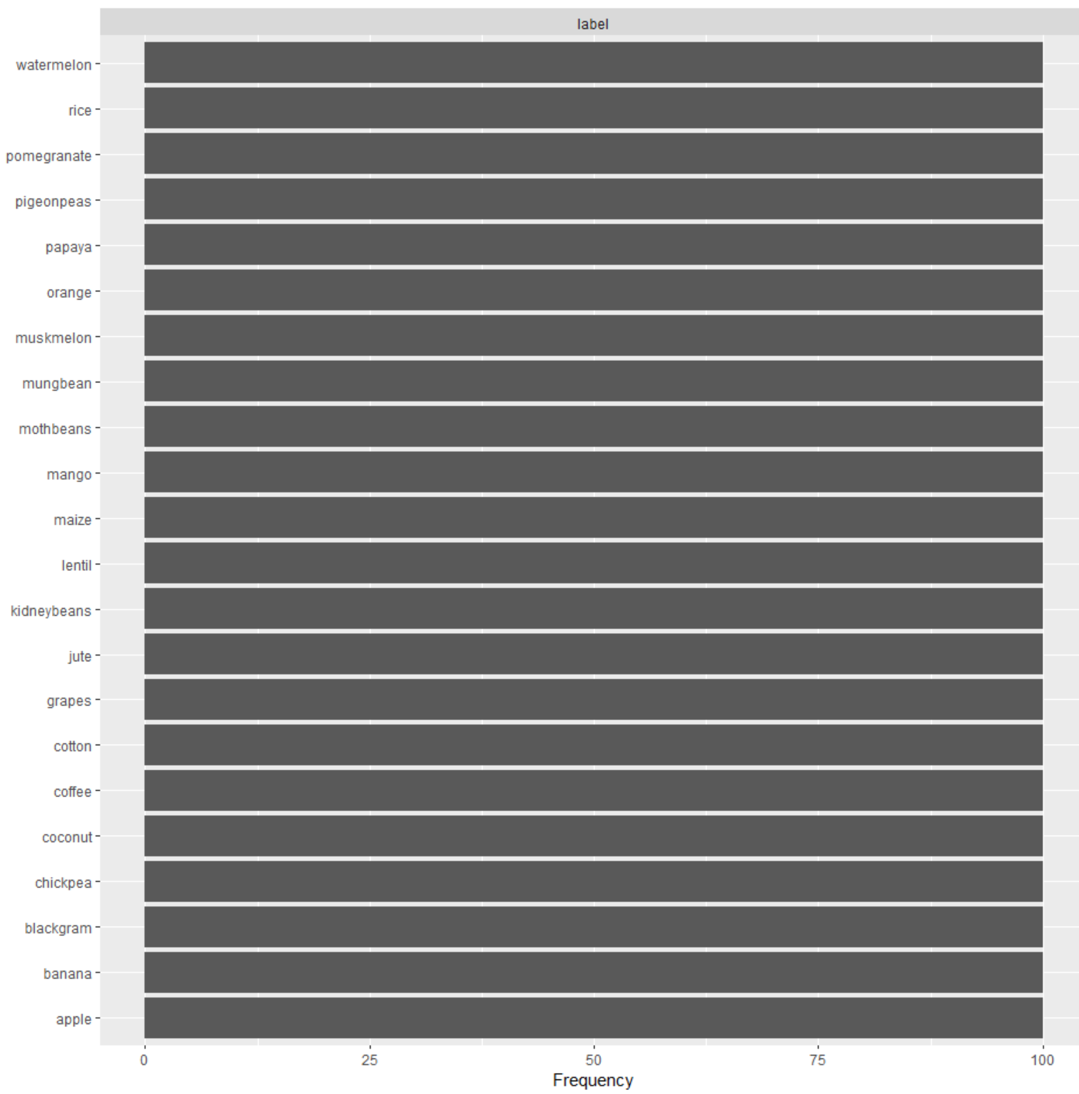
Agriculture is undergoing a digital transformation driven by the need to improve efficiency, sustainability, and resilience in the face of growing global challenges. Smart farming—which leverages data, sensors, and machine and statistical learning offers powerful tools to support better crop management, environmental monitoring, and decision-making. In this project, we use a real-world dataset, Smart Farming 2024 (SF24), comprising 2200 observations and 23 original features to address a critical task in precision agriculture: *multiclass classification* of crop types. Each record includes measurements of key environmental and soil factors such as nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, rainfall, pH, soil type, irrigation practices, and more. The target variable, *label* is categorical, representing 21 distinct crop types grown under various conditions across California. Our objective is to build predictive models capable of classifying crop types accurately based on the observed features. This task has practical applications in optimizing crop selection, managing agricultural inputs, and responding proactively to environmental stress.

Methods

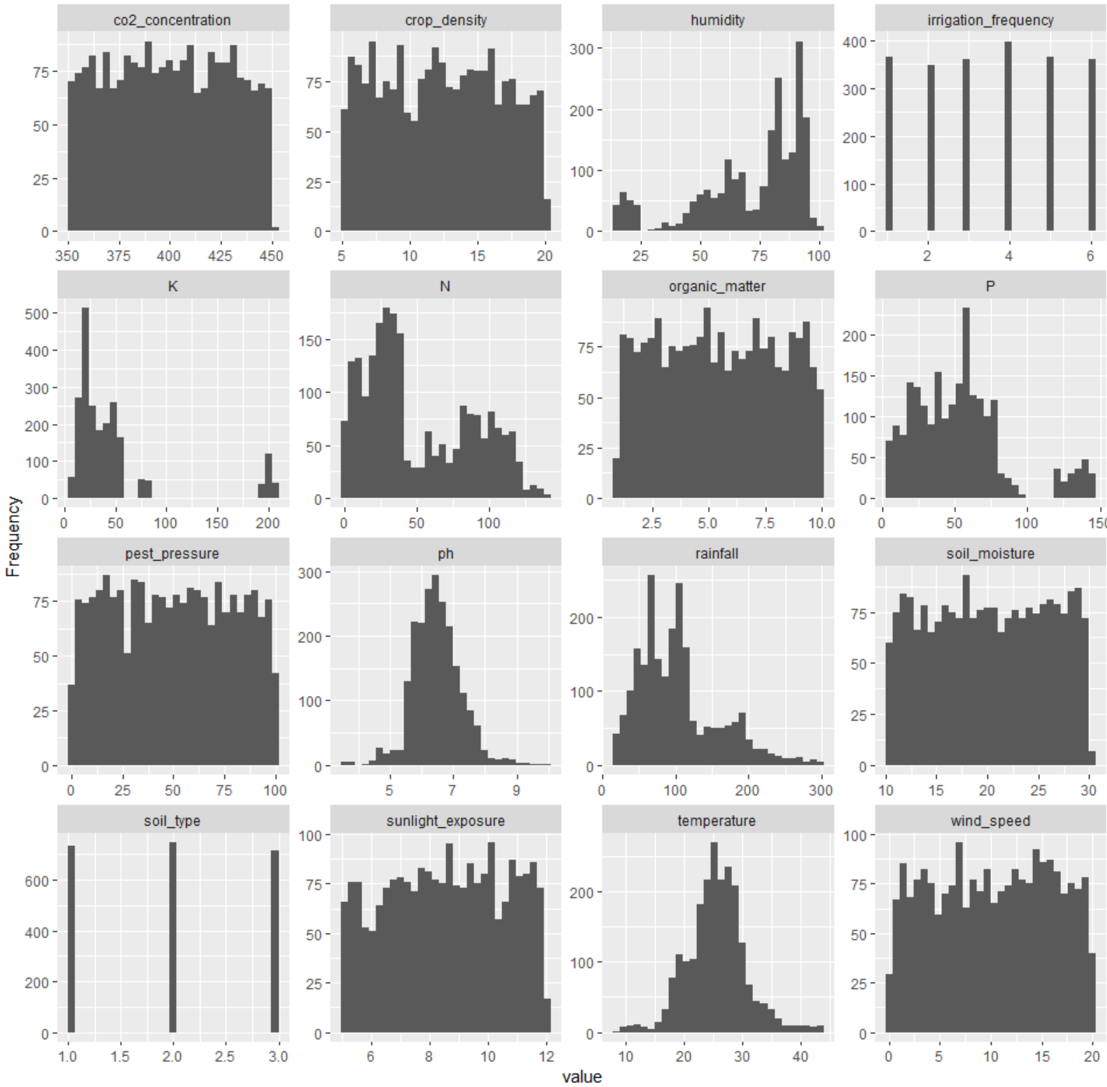
Exploratory Data Analysis

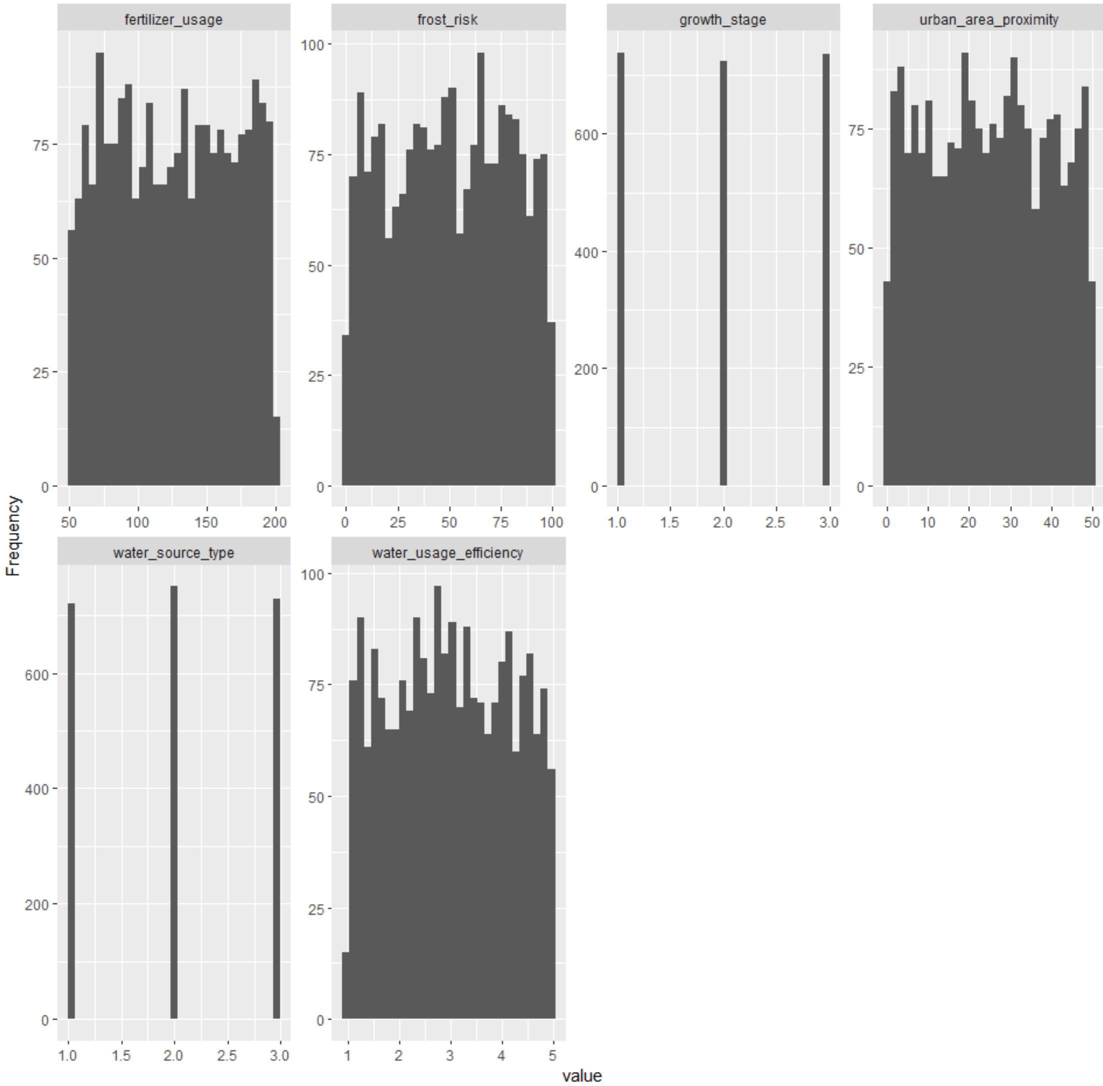
The analysis begins with exploratory data analysis (EDA) to overview the data, its structure and feature distributions. Correlation analysis helped assess relationships between variables.

Class Distribution of target variable, “label”

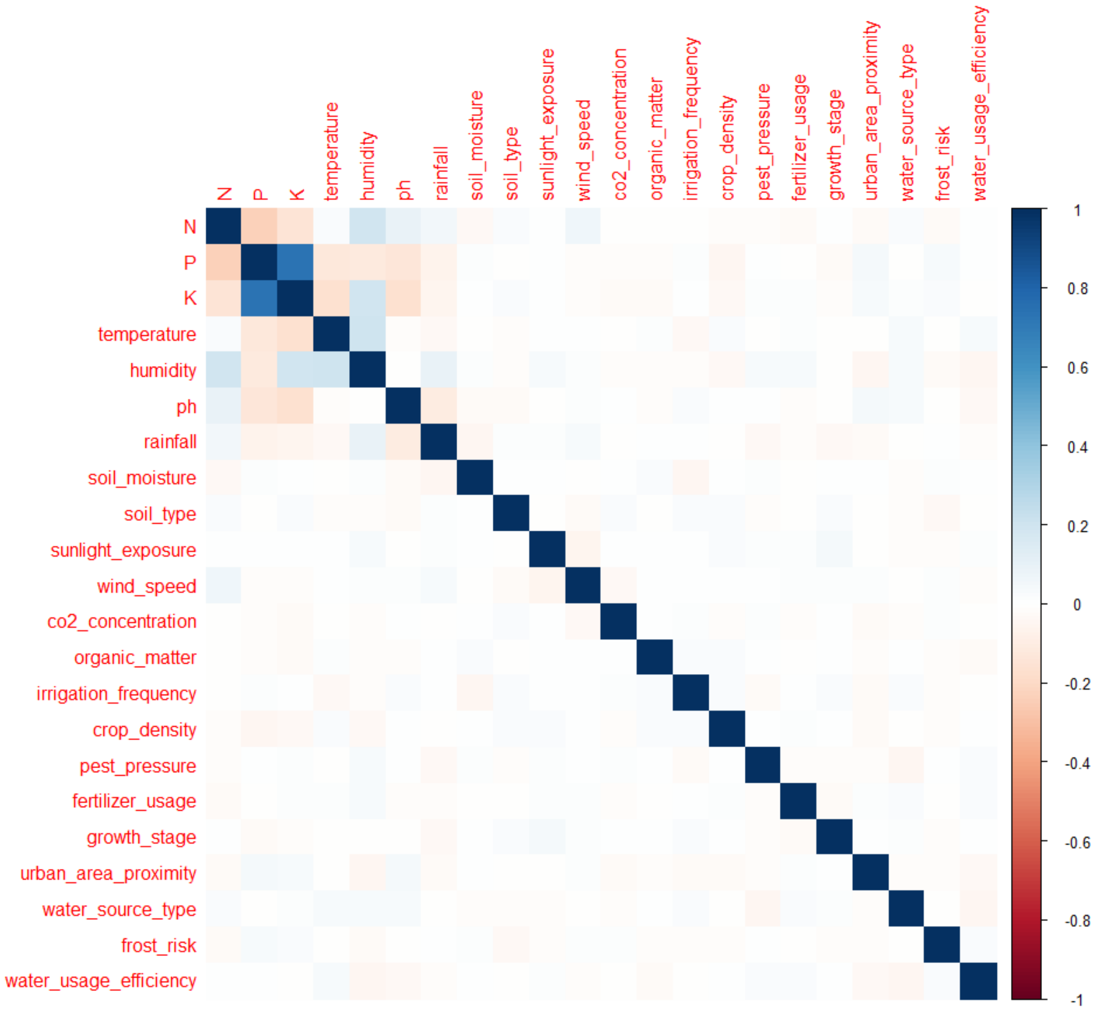


Distribution of numeric variables




fertilizer_usage
frost_risk
growth_stage
urban_area_proximity
water_source_type
water_usage_efficiency
Frequency
value

Correlation among predictors



Preprocessing Data

All analyses were conducted using R, and multiple libraries were utilized to handle data manipulation, visualization, modeling, and evaluation. Packages used included tidyverse, caret, glmnet, randomForest, nnet, e1071, pROC, DMwR2, smotefamily, and others. The preprocessing pipeline ensured data integrity, standardization, and proper preparation for model training and evaluation. All character variables were converted to factors to ensure compatibility with modeling functions.

Data splitting into training and test data

To maintain the integrity of model evaluation and prevent data leakage, scaling was performed after splitting the dataset into training (70%) and testing (30%) sets. Numeric features were standardized using scale(), with the test data scaled using the mean and standard deviation derived from the training data only.

```
trainIndex <- createDataPartition(scaled_data[[target]], p = 0.7, list =
FALSE)
trainData <- scaled_data[trainIndex, ]
testData  <- scaled_data[-trainIndex, ]
```

Model Matrix Conversion

To prepare for algorithms that require numerical input, model.matrix() was used to convert factor variables into dummy variables for training (x_train) and test (x_test) sets. The response variable (label) was isolated into y_train and y_test. Then training and test data were scaled separately to avoid information leakage. The test set was scaled using the mean and standard deviation derived from the training set only.

```
x_train <- model.matrix(as.formula(paste(target, "~ .")), data =
trainData)[,-1]
y_train <- trainData[[target]]
x_test  <- model.matrix(as.formula(paste(target, "~ .")), data = testData)[,-
1]
y_test  <- testData[[target]]
```

Model fitting

A diverse set of 7 models—multinomial logistic regression, LDA, QDA, KNN, random forest, SVM, and neural network were trained to capture both linear and non-linear patterns in the data. Logistic regression, LDA, and QDA were selected for their interpretability and statistical foundation. KNN was included for its simplicity and ability to model complex boundaries. Random forest and SVM were chosen for their robustness and strong performance in high-dimensional, non-linear settings. A neural network was used to explore deep, non-linear feature interactions that traditional models might miss.

```
suppressMessages(suppressWarnings(
  nnet.fit <- multinom(label ~ ., data = train_scaled)
))

# Logistic regression

nnet.fit= multinom(label ~ ., data = train_scaled)

# LDA

lda.fit=lda(label ~ ., data = train_scaled)

# QDA

qda.fit= qda(label ~ ., data = train_scaled)
```

```r
# KNN

knn.fit= knn(train = x_train, test = x_test, cl = y_train, k = 5)

# Random Forest

rf.fit <- randomForest(label ~ ., data = train_scaled,
                       ntree = 500,
                       mtry = sqrt(ncol(train_scaled) - 1),
                       importance = TRUE)

# SVM

svm.fit <- svm(label ~ ., data = train_scaled,
               kernel = "radial",
               cost = 1,
               gamma = 1/ncol(train_scaled))

# Neural Network

nn.fit <- nnet(label ~ ., data = train_scaled,
               size = 5,
               maxit = 500,
               decay = 0.01,
               trace = FALSE)
```

Model Comparison and Evaluation

Model performance was evaluated using accuracy and confusion matrices to compare classification success across models and identify specific misclassification patterns for each crop class.

```r
# True labels

true_labels <- test_scaled$label

# Logistic Regression

nnet.pred <- predict(nnet.fit, newdata = test_scaled)
nnet.acc <- mean(nnet.pred == true_labels)

# LDA

lda.pred <- predict(lda.fit, test_scaled)$class
lda.acc <- mean(lda.pred == true_labels)

# QDA
```

```r
qda.pred <- predict(qda.fit, test_scaled)$class
qda.acc <- mean(qda.pred == true_labels)

# KNN

knn.pred <- knn(train = train_scaled[, -which(names(train_scaled) ==
"label")],
                test = test_scaled[, -which(names(test_scaled) == "label")],
                cl = train_scaled$label, k = 5)
knn.acc <- mean(knn.pred == true_labels)

# Random Forest

rf.pred <- predict(rf.fit, newdata = test_scaled)
rf.acc <- mean(rf.pred == true_labels)

# SVM

svm.pred <- predict(svm.fit, newdata = test_scaled)
svm.acc <- mean(svm.pred == true_labels)

# Neural Net

nn.pred <- predict(nn.fit, newdata = test_scaled, type = "class")
nn.acc <- mean(nn.pred == true_labels)

### Comparing accuracies

accuracy_df <- data.frame(
  Model = c("Logistic Regression", "LDA", "QDA", "KNN", "Random Forest",
"SVM", "Neural Network"),
  Accuracy = c(nnet.acc, lda.acc, qda.acc, knn.acc, rf.acc, svm.acc, nn.acc)
)


plot_accuracy<-ggplot(accuracy_df, aes(x = reorder(Model, Accuracy), y =
Accuracy)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = round(Accuracy, 3)), vjust = -0.5, size = 3.5) +
  coord_flip() +
  labs(title = "Model Accuracy Comparison",
       x = "Model",
       y = "Accuracy") +
  theme_minimal()

### Plotting Confusion Matrices for all models
```

```r
plot_conf_matrix <- function(pred, actual, title) {
  cm <- table(Predicted = pred, Actual = actual)
  cm_df <- as.data.frame(cm)
  ggplot(cm_df, aes(x = Actual, y = Predicted, fill = Freq)) +
    geom_tile(color = "white") +
    geom_text(aes(label = Freq), size = 3.5) +
    scale_fill_gradient(low = "lightblue", high = "steelblue") +
    labs(title = title, x = "Actual", y = "Predicted") +
    theme_minimal()
}

p1 <- plot_conf_matrix(nnet.pred, test_scaled$label, "Logistic Regression")
p2 <- plot_conf_matrix(lda.pred, test_scaled$label, "LDA")
p3 <- plot_conf_matrix(qda.pred, test_scaled$label, "QDA")
p4 <- plot_conf_matrix(knn.pred, test_scaled$label, "KNN")
p5 <- plot_conf_matrix(rf.pred,  test_scaled$label, "Random Forest")
p6 <- plot_conf_matrix(svm.pred, test_scaled$label, "SVM")
p7 <- plot_conf_matrix(nn.pred,  test_scaled$label, "Neural Network")
```

Feature Selection

Random Forest

To enhance model interpretability and potentially improve generalization, feature selection was performed on the best-performing model, Random Forest. This model achieved the highest classification accuracy in our multiclass setting with 21 classes. Feature importances were extracted using the importance() function and ranked after which model was refitted using only selected features.

Results

Model Summaries

Logistic Regression

The residual deviance (0.00019) indicates an extremely good fit. The AIC (966.00) reflects model quality relative to complexity—useful for comparing with other models.

```
## Call:
## multinom(formula = label ~ ., data = train_scaled)
##
## Coefficients:
##             (Intercept)          N            P          K temperature
## banana        -308.3378  572.55413   510.021096 -844.02604   106.41125
## blackgram      367.1179  107.63241    11.426090 -412.49258    98.08159
## chickpea       487.2923  324.21416  -128.284035  180.27802   -49.93348
## coconut       -308.8934 -106.62804  -673.638149 -106.77579   107.82914
## coffee        -463.4879  725.67071 -1059.170477 -167.05030  -169.31040
```

```
## cotton       -1115.1265  778.23335    -7.908221 -2202.70107  -148.60323
## grapes         269.2270   38.74935  -391.918805   345.14179   -23.72347
## jute           601.9005  406.75529  -380.354162    71.64716  -162.87802
## kidneybeans   -445.6136  316.92846  -442.537224  -526.59029   -90.50343
## lentil       -1340.2308 -198.42477   451.344795 -1047.85044  -426.34839
## maize          153.6152  475.40524  -445.506094 -1003.65459  -237.06119
## mango          178.7806  -19.24116 -1159.676575   566.77290   278.14437
## mothbeans     -341.9710 -226.12736  -845.713860   238.62669   137.97609
## mungbean      -655.8652 -221.50972  -219.577585 -1207.12732  -144.56171
## muskmelon    -1047.8335  106.50126  -469.089937   282.22929   189.03036
## orange        -802.6366   98.18827  -611.979568 -2213.60136   -67.63268
## papaya        -541.0912  -51.07379   380.669299  -586.28892   182.05165
## pigeonpeas     233.4435 -104.35698   272.039349  -396.72022    62.88995
## pomegranate    245.6052 -184.08740  -838.463826   208.62518  -116.09175
## rice         -1071.0542  492.11559  -633.088570   587.18243  -436.48024
## watermelon     649.6520  213.13800  -649.184891   384.05577    83.82209
##                humidity          ph    rainfall soil_moisture soil_type
## banana       -176.30208 -174.917766  -239.64938    -38.593765  17.75017
## blackgram    -440.93277   65.616750  -414.83619     13.959736 -65.07105
## chickpea     -865.50819  100.402958  -203.20961    -64.033526 -18.60725
## coconut       191.74114  -80.972868    41.60053    -40.129580  91.13754
## coffee      -1167.72052  104.078181    62.77232     49.120560 -46.32951
## cotton        606.27912  198.485861  -490.14476    -99.237552 -98.45697
## grapes       -692.06926   35.226413  -805.53688    -68.022260 -42.17084
## jute         -565.14728  127.476347  -115.68532    -51.654393 -13.85168
## kidneybeans -1301.94717   47.991340  -156.11864    -48.779794 -16.69797
## lentil       -288.18324  131.642986 -1782.49601    100.621190 -10.71665
## maize        -576.03747    3.919389  -223.97100    -67.393151 -30.62695
## mango        -887.90957   34.152699  -120.17424    -67.799115 -42.33801
## mothbeans    -939.46680   60.229074 -1275.50486    -48.226304 -35.22863
## mungbean      302.00024   55.188425  -976.69667    -24.412176   8.34252
## muskmelon     136.02472   37.393547 -1537.89974    -50.531785 -32.58599
## orange        -80.69437   13.056706  -192.42914    -49.054802  12.04244
## papaya       1071.36028  123.560679   -70.47463      4.151843 137.78610
## pigeonpeas   -577.52912  -26.495945    65.43523    -29.945213  18.02068
## pomegranate  -433.67943   28.220271  -339.49065    -37.846356  20.45102
## rice          -52.81828   88.302167   700.39342    -68.954357 -43.51933
## watermelon   -648.94457   39.288234  -323.44154    -49.603087 -21.12986
##             sunlight_exposure  wind_speed co2_concentration organic_matter
## banana              79.517474 -132.403603        -7.8960674      -8.587169
## blackgram            3.758894  -13.184852        -7.3090212      23.354829
## chickpea            10.265619    6.200597       -73.1873953      23.054773
## coconut            -20.563640  -49.567090        -6.3963774       1.417455
## coffee               8.517057  -48.418494        60.2559228      93.422621
## cotton             -19.334932  -10.291946        29.4765334      -6.459006
## grapes              -4.659246  -29.607727        -1.7162470      -1.367853
## jute                 4.868476  -37.661968        28.3470199      23.285461
## kidneybeans        -70.407491   23.653027      -100.2942242     -22.508937
## lentil              32.188253  -32.933233        46.1054501      72.416952
## maize               13.463281  -40.043853        82.2338480      -1.068405
```

```
## mango             -1.392600  -77.511203       -51.9392285      39.657466
## mothbeans         71.152877  -41.829602       -20.5238358      12.119998
## mungbean          34.945755   10.892335         6.1051397      39.273786
## muskmelon         22.006996  -64.150089        23.3861374      42.016514
## orange           -28.312885   -4.621651       -11.6658765      52.506456
## papaya            27.070011    8.328384       -25.8575656     -50.207881
## pigeonpeas       -27.508704    1.014325       -50.9016665      15.602358
## pomegranate       32.631239  -41.704142         0.3656926      22.912431
## rice             -53.257521 -123.807080        38.2739225     -54.081105
## watermelon        39.314200  -27.836784        49.2904942      11.429787
##             irrigation_frequency crop_density pest_pressure
fertilizer_usage
## banana               -37.716858   -75.275567    -29.217042        -
52.538173
## blackgram             34.028094    26.094785      1.701067        -
17.321784
## chickpea             -38.528640    16.480767     22.543270        -
19.845737
## coconut               -4.676940    34.389909     11.411146        -
61.838913
## coffee               -50.221858   -41.913925     -4.534914        -
12.177840
## cotton               -29.506709   103.366737     54.096962        -
33.330272
## grapes               -58.777323    13.624828     75.496421        -
76.903254
## jute                 -80.469633   -30.350831     39.904619        -
25.462860
## kidneybeans         -108.680814    22.542418     18.709365        -
22.473808
## lentil                 7.403214   -39.322643     -8.512629        -
41.069528
## maize                -32.884557    -6.560902     67.418154         -
7.244177
## mango                -80.312998    35.403220     20.345773        -
69.625472
## mothbeans            -14.715549    37.435821     71.548925        -
52.013869
## mungbean              -4.020095    -9.156018      6.908473        -
68.401949
## muskmelon            -33.542606    58.450939     13.278696        -
85.968944
## orange                 5.214271    -6.878943      2.901442        -
66.555779
## papaya                -2.231867   -36.842819     19.194486       -
108.632617
## pigeonpeas            -8.298908    15.057469     15.832922        -
52.865291
## pomegranate          -61.181315    95.147507     30.450159        -
60.522802
```

```
## rice                 -84.847504   -70.746913     82.598947       -
13.022196
## watermelon           -28.995370     8.264524     30.404617       -
80.766235
##              growth_stage urban_area_proximity water_source_type frost_risk
## banana         10.057931           48.5294523          4.492642 -27.614701
## blackgram      13.962985           -0.1943375          1.173649 -12.077955
## chickpea      -30.126947          -46.6353563         38.490875 -47.668490
## coconut      -100.590113           31.3650032         32.218365 -11.645760
## coffee         85.680003          -72.2152320         53.125733 -51.943017
## cotton          4.952715          -27.1276057         43.822969 -47.385866
## grapes         11.507350          -30.2333288          6.292668 -25.432739
## jute           26.842834          -48.1357616          8.911026 -18.155709
## kidneybeans   -28.037231            3.7488670         55.046778 -22.371409
## lentil        -37.155358           34.7329912         71.282341  66.584925
## maize          41.892212          -31.1444538         19.777725 -50.970919
## mango          37.773191           14.0433528        -48.454714 -26.083883
## mothbeans      31.423348           29.7366874          4.550115  -2.288031
## mungbean       -7.213380           26.1616934         15.886658   1.662185
## muskmelon     -39.661322          -32.2318170          6.009323 -34.012849
## orange         34.203938           -5.5635536         31.451894   5.157985
## papaya        -59.854474           21.8820166         33.285844   2.490309
## pigeonpeas    -51.627018            7.1363097         30.206379 -46.817139
## pomegranate   -88.455322           77.1134698        -11.418399 -49.610065
## rice          -27.918636           28.0100966        -82.960649 -38.083923
## watermelon     36.118408          -31.9537024         11.091647 -26.481441
##              water_usage_efficiency
## banana                   -61.101162
## blackgram                 10.517405
## chickpea                  21.921239
## coconut                    5.456552
## coffee                     9.893065
## cotton                   -10.667035
## grapes                     5.710697
## jute                      30.097678
## kidneybeans               -9.090692
## lentil                    10.515950
## maize                    -12.553008
## mango                    -45.022435
## mothbeans                -59.524648
## mungbean                  -5.155548
## muskmelon                 84.617280
## orange                   -69.765869
## papaya                   -35.517501
## pigeonpeas                -9.770811
## pomegranate               -2.047857
## rice                      94.746765
## watermelon                35.225735
##
## Std. Errors:
```

```
##              (Intercept)            N            P            K
temperature
## banana      2.955614e+02 3.565144e+02 1.836037e+02 1.093261e+01
1.939976e+01
## blackgram   3.576910e+03 6.323136e+02 6.968517e+02 2.201556e+03
4.697121e+02
## chickpea    9.850790e+02 1.589469e+03 6.993019e+02 1.495102e+03
2.230075e+03
## coconut     9.603548e+02 2.475554e+02 2.227393e+02 9.234707e+01
2.019260e+02
## coffee      2.694359e-07 4.269839e-07 2.021913e-07 1.073559e-07 1.215317e-
07
## cotton      1.495833e+03 2.899243e+03 1.015039e+03 1.270069e+02
2.021472e+02
## grapes      1.473564e+03 1.350221e+03 3.464860e+03 4.408525e+03
4.784091e+03
## jute        1.610237e+04 3.165010e+04 2.140721e+04 1.269175e+04
4.264952e+04
## kidneybeans 3.453189e+02 2.476699e+02 1.408940e+02 1.784022e+02
3.668765e+02
## lentil      2.625623e-05 1.437383e-05 7.050699e-06 1.560915e-05 7.977241e-
06
## maize       1.718330e+04 6.029310e+03 4.586476e+03 1.198448e+04
1.496358e+04
## mango       2.422590e+04 1.033698e+04 2.981579e+04 6.237988e+03
2.281881e+03
## mothbeans   2.466703e+03 2.617775e+03 1.059837e+03 1.457051e+03
4.152414e+03
## mungbean    4.733717e+02 2.369199e+02 4.123732e+02 3.662326e+02
1.404811e+03
## muskmelon   7.360938e-16 3.494794e-16 3.451716e-16 5.633588e-17 1.295108e-
17
## orange      4.423156e+02 2.258205e+02 3.799879e+02 3.446397e+02
1.403063e+03
## papaya      2.478745e+04 2.740329e+02 4.498156e+03 2.176657e+03
3.046805e+04
## pigeonpeas           NaN          NaN          NaN          NaN
NaN
## pomegranate 2.225864e+04 8.125299e+03 2.888112e+04 5.338080e+03
5.719996e+03
## rice        6.370779e+03 3.354437e+04 2.139627e+04 1.320673e+04
3.970049e+04
## watermelon  1.454738e+04 7.152294e+03 7.440646e+03 3.350023e+03
1.942877e+04
##                 humidity           ph     rainfall soil_moisture
soil_type
## banana      1.195835e+02 1.756582e+02 3.727927e+01  4.059216e+02
3.652611e+02
## blackgram   5.775628e+02 4.740880e+03 1.922036e+03  1.410379e+03
2.868573e+00
```

```
## chickpea    2.590027e+03 1.588029e+03 5.294050e+02  2.562650e+03
2.927169e+03
## coconut     9.288994e+02 3.608548e+02 1.401943e+03  1.538504e+03
1.469578e+00
## coffee      1.327175e-07 8.773498e-08 4.467220e-07  8.998764e-08
3.325528e-07
## cotton      1.038745e+03 3.887137e+02 1.660231e+03  1.448569e+03
1.848583e+03
## grapes      1.766320e+03 8.823331e+02 8.005354e+02  1.755138e+03
2.041059e+03
## jute        1.686477e+04 2.686475e+04 6.884723e+03  1.653769e+04
4.621875e+04
## kidneybeans 7.544330e+02 3.745372e+02 9.185298e+00  2.764000e+02
4.267525e+02
## lentil      5.688298e-06 2.839026e-05 2.490644e-05  1.507600e-05
3.373998e-05
## maize       5.659525e+03 1.297801e+04 4.912054e+03  2.620586e+04
6.623129e+04
## mango       1.223207e+04 4.471036e+03 1.418796e+03  3.972411e+03
2.508161e+04
## mothbeans   4.018212e+03 3.062724e+03 6.836994e+02  3.250956e+03
2.859468e+03
## mungbean    4.743605e+02 8.037175e+02 1.733511e+01  2.892191e+02
5.850030e+02
## muskmelon   7.041447e-16 3.253681e-16 6.714714e-16  4.608856e-16
5.903183e-19
## orange      4.449583e+02 7.468478e+02 3.746329e+01  3.576403e+02
5.466233e+02
## papaya      2.100515e+04 8.258007e+03 1.335809e+04  1.166695e+04
2.125373e+04
## pigeonpeas           NaN          NaN          NaN           NaN
NaN
## pomegranate 1.410010e+04 6.901915e+03 1.520775e+03  1.634371e+03
2.750771e+04
## rice        1.718563e+04 3.226288e+04 1.019146e+04  2.632517e+04
3.213028e+04
## watermelon  1.448849e+04 9.169426e+03 1.391418e+04  8.486120e+03
2.700937e+04
##             sunlight_exposure   wind_speed co2_concentration
organic_matter
## banana           3.378579e+02 4.379622e+02      2.599904e+02
4.301295e+02
## blackgram        5.510156e+03 5.645466e+03      1.130897e+03
2.037521e+03
## chickpea         2.319557e+03 1.842186e+03      2.381023e+03
2.799191e+03
## coconut          1.337146e+03 1.058051e+03      1.375610e+03
7.515232e+02
## coffee           2.713954e-07 5.434332e-08      2.107823e-07   3.458627e-
07
```

```
## cotton          5.880526e+02 2.041886e+03      2.304320e+03
2.541756e+03
## grapes          1.649217e+03 2.179965e+03      2.454700e+03
2.190190e+03
## jute            2.096040e+04 3.024426e+04      1.775025e+04
1.929534e+04
## kidneybeans     4.371586e+02 4.951841e+02      6.010899e+02
2.804740e+02
## lentil          1.361316e-05 1.200615e-05      3.708748e-05   3.201607e-
05
## maize           7.380884e+03 2.569037e+04      1.237803e+04
2.635721e+04
## mango           1.617384e+03 5.333427e+03      2.346340e+04
2.122158e+04
## mothbeans       3.033671e+03 2.938952e+03      2.552061e+03
3.459495e+03
## mungbean        5.441929e+02 4.558908e+02      7.552428e+02
1.420515e+02
## muskmelon       3.793705e-17 5.628814e-16      3.662658e-16   1.503383e-
16
## orange          5.399150e+02 4.744002e+02      8.316795e+02
2.133191e+02
## papaya          3.191709e+04 3.384028e+04      7.684403e+03
3.109749e+04
## pigeonpeas               NaN          NaN               NaN
NaN
## pomegranate     4.047367e+02 7.034930e+03      2.489898e+04
2.273197e+04
## rice            2.550829e+04 4.485807e+04      2.446522e+04
2.240954e+04
## watermelon      3.814298e+04 3.860877e+04      5.610758e+03
4.323066e+04
##             irrigation_frequency crop_density pest_pressure
fertilizer_usage
## banana             8.915737e+01 7.941814e+01  3.806582e+00
1.467761e+02
## blackgram          5.231054e+03 1.745239e+03  5.455827e+03
5.077322e+03
## chickpea           1.059690e+03 3.174595e+03  1.502726e+03
2.829427e+03
## coconut            8.535349e+02 1.556733e+03  2.791985e+02
7.112225e+02
## coffee             3.961160e-07 4.443370e-07  4.524653e-07     3.274682e-
07
## cotton             1.560024e+03 2.237187e+03  2.580879e+03
2.381645e+03
## grapes             2.234531e+03 1.832072e+03  1.123832e+03
1.098982e+03
## jute               2.251595e+04 2.597232e+04  4.143146e+04
2.520405e+04
```

```
## kidneybeans         3.072265e+02 1.192888e+02  2.798459e+01
1.005361e+02
## lentil              2.427821e-05 4.221845e-05  3.435946e-05     3.592673e-
05
## maize               2.330914e+04 2.111548e+04  4.472550e+04
2.782359e+04
## mango               1.655956e+04 2.533104e+04  6.123118e+03
2.053153e+04
## mothbeans           2.358969e+03 3.797887e+03  2.958920e+03
4.771093e+03
## mungbean            6.995129e+02 6.362987e+02  4.472475e+01
8.843560e+00
## muskmelon           6.548948e-16 9.779591e-16  2.471033e-16     5.952244e-
16
## orange              7.763298e+02 7.334817e+02  4.144230e+01
8.686263e+01
## papaya              6.270191e+03 2.393114e+04  1.524710e+04
3.074477e+04
## pigeonpeas                   NaN          NaN           NaN
NaN
## pomegranate         1.946331e+04 2.876054e+04  9.068890e+03
1.771146e+04
## rice                2.838023e+04 2.401020e+04  3.538022e+04
4.553768e+04
## watermelon          8.093430e+03 2.164756e+04  1.375807e+04
3.189912e+04
##             growth_stage urban_area_proximity water_source_type
frost_risk
## banana      3.605394e+02         4.233490e+02      5.167003e+00
1.144094e+02
## blackgram   4.363280e+03         1.913038e+03      4.439739e+03
4.850174e+03
## chickpea    5.457441e+03         1.365898e+03      3.464684e+03
1.236317e+03
## coconut     3.792347e+00         9.093120e+02      1.634911e+01
2.520579e+02
## coffee      1.068739e-09         3.944956e-07      3.250169e-07 3.179149e-
07
## cotton      2.048654e+03         2.228139e+03      1.576534e+02
2.712644e+03
## grapes      7.446127e+02         9.624745e+02      7.526266e+02
1.930194e+03
## jute        3.325188e+04         3.917261e+04      3.246058e+04
2.211426e+04
## kidneybeans 1.363223e+00         3.478569e+02      6.036864e+00
3.598732e+02
## lentil      3.114177e-05         2.446128e-05      3.100338e-05 1.370721e-
05
## maize       4.199092e+04         1.814399e+04      3.003985e+02
2.693447e+04
```

```
## mango       2.306476e+03         1.599310e+04      2.839399e+03
8.274415e+03
## mothbeans   6.543866e+03         2.968828e+03      3.585917e+03
3.310501e+03
## mungbean    5.774407e+02         2.012928e+02      5.710084e+02
8.122071e+02
## muskmelon   2.905894e-18         1.197522e-15      1.286839e-17 8.963608e-
16
## orange      5.395571e+02         2.164303e+02      5.335468e+02
8.047169e+02
## papaya      3.116866e+04         6.962239e+03      4.333388e+02
2.032347e+04
## pigeonpeas           NaN                  NaN               NaN
NaN
## pomegranate 8.787094e+01         1.368086e+04      3.891256e+02
1.146524e+04
## rice        2.513073e+04         5.175119e+04      3.759698e+04
3.767662e+04
## watermelon  2.369590e+04         2.098148e+04      1.106598e+04
2.331127e+04
##             water_usage_efficiency
## banana                1.289798e+02
## blackgram             1.333644e+03
## chickpea              3.590598e+03
## coconut               4.161027e+02
## coffee                1.897626e-07
## cotton                7.774492e+02
## grapes                1.103404e+03
## jute                  6.068598e+03
## kidneybeans           4.607446e+02
## lentil                3.139609e-05
## maize                 9.346570e+03
## mango                 3.404747e+04
## mothbeans             4.186695e+03
## mungbean              5.022126e+02
## muskmelon             1.209647e-15
## orange                4.884959e+02
## papaya                1.874898e+04
## pigeonpeas                     NaN
## pomegranate           3.333109e+04
## rice                  5.263398e+03
## watermelon            2.076236e+04
##
## Residual Deviance: 0.000191721
## AIC: 966.0002
```

LDA

```
##         Length Class  Mode
## prior    22    -none- numeric
## counts   22    -none- numeric
## means   484    -none- numeric
## scaling 462    -none- numeric
## lev      22    -none- character
## svd      21    -none- numeric
## N         1    -none- numeric
## call      3    -none- call
## terms     3    terms  call
## xlevels   0    -none- list
```

QDA

```
##         Length Class  Mode
## prior      22  -none- numeric
## counts     22  -none- numeric
## means     484  -none- numeric
## scaling 10648  -none- numeric
## ldet       22  -none- numeric
## lev        22  -none- character
## N           1  -none- numeric
## call        3  -none- call
## terms       3  terms  call
## xlevels     0  -none- list
```

Random Forest

```
##                 Length Class  Mode
## call                6  -none- call
## type                1  -none- character
## predicted        1540  factor numeric
## err.rate        11500  -none- numeric
## confusion         506  -none- numeric
## votes           33880  matrix numeric
## oob.times        1540  -none- numeric
## classes            22  -none- character
## importance        528  -none- numeric
## importanceSD      506  -none- numeric
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y                1540  factor numeric
## test                0  -none- NULL
```

```
## inbag               0  -none- NULL
## terms               3  terms  call
```

LDA assumes equal covariance across classes and uses linear decision boundaries, which is reflected in its simpler, less complex model and appropriate for linearly separable data. QDA, on the other hand, estimates class-specific covariance, allowing for non-linear decision boundaries, which explains its improved performance for data with more complex relationships, as seen in the greater flexibility in its model. Random Forest, which handles complex feature interactions and offers robust class predictions, performs well in terms of feature importance and error rate, though the relatively high error rate indicates possible areas for model optimization. Thus, LDA works well for simpler, linearly separable data, QDA is preferred for non-linear data, and Random Forest provides the best performance but might need further tuning for error reduction.

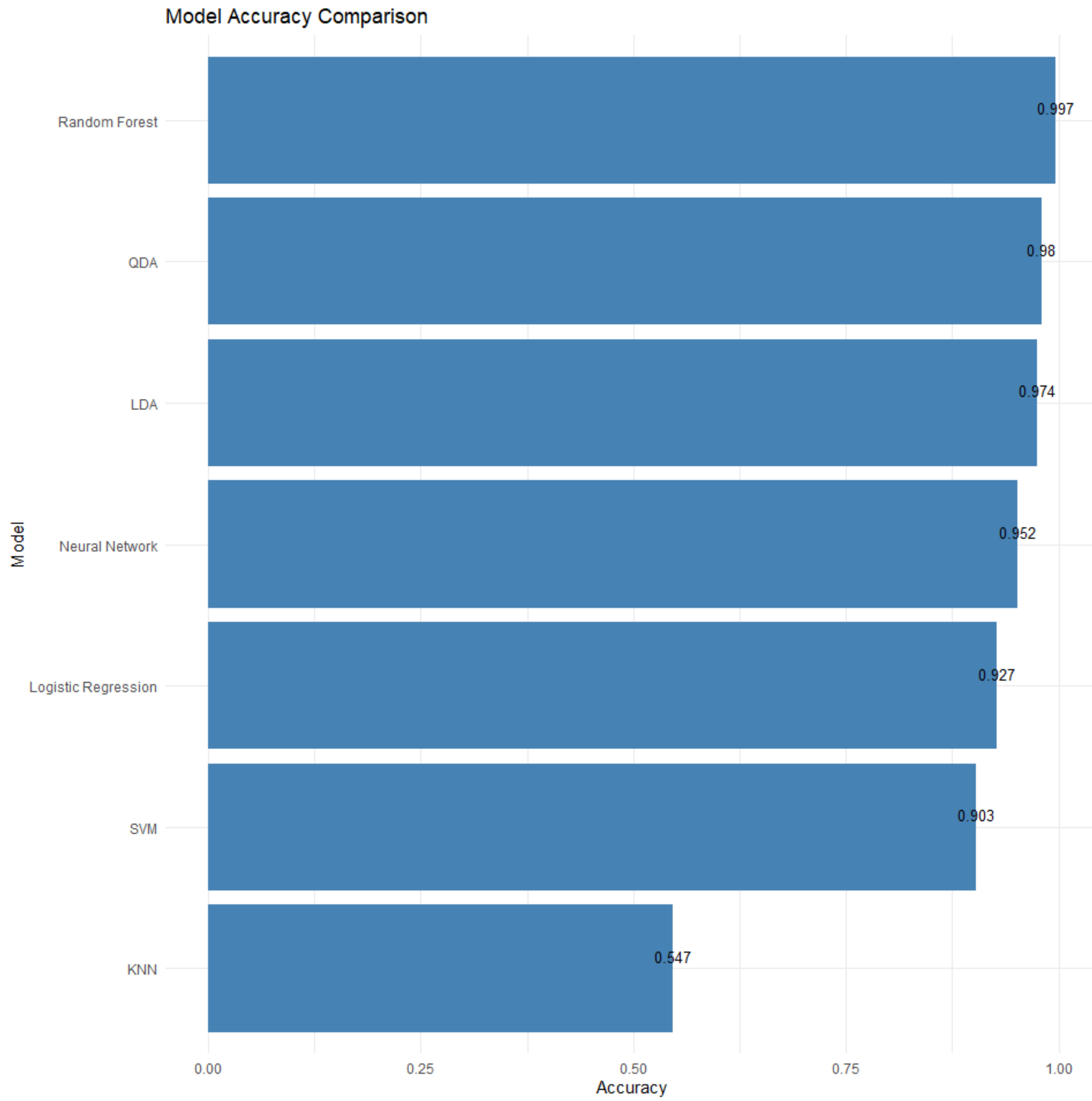
Model Comparison and Evaluation

To evaluate the performance of various classification models on the Smart Farming dataset, seven supervised learning models were trained and tested using 70:30 train-test split on scaled numeric features and encoded categorical variables. Model performance was assessed using classification accuracy as the metric.

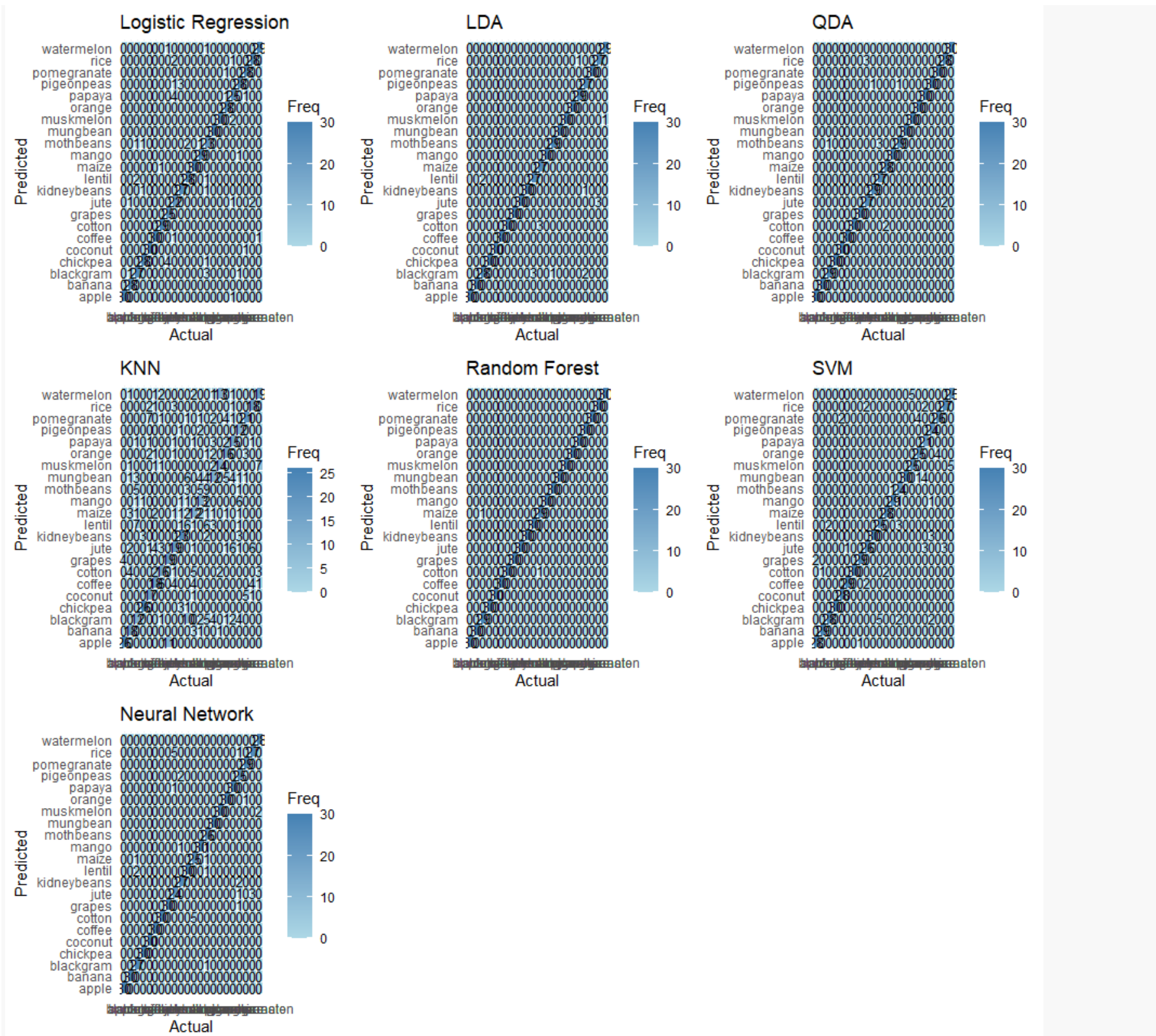
The classification accuracy for each model is summarized below:

```
##                 Model  Accuracy
## 1 Logistic Regression 0.9272727
## 2                 LDA 0.9742424
## 3                 QDA 0.9803030
## 4                 KNN 0.5469697
## 5       Random Forest 0.9969697
## 6                 SVM 0.9030303
```

```
## 7       Neural Network 0.9515152
```



Among all models, Random Forest achieved the highest accuracy at 99.55%, followed by QDA (96.97%), and Neural Network (96.52%). The KNN model performed the worst, with an accuracy of just 52.27%, indicating poor generalization.



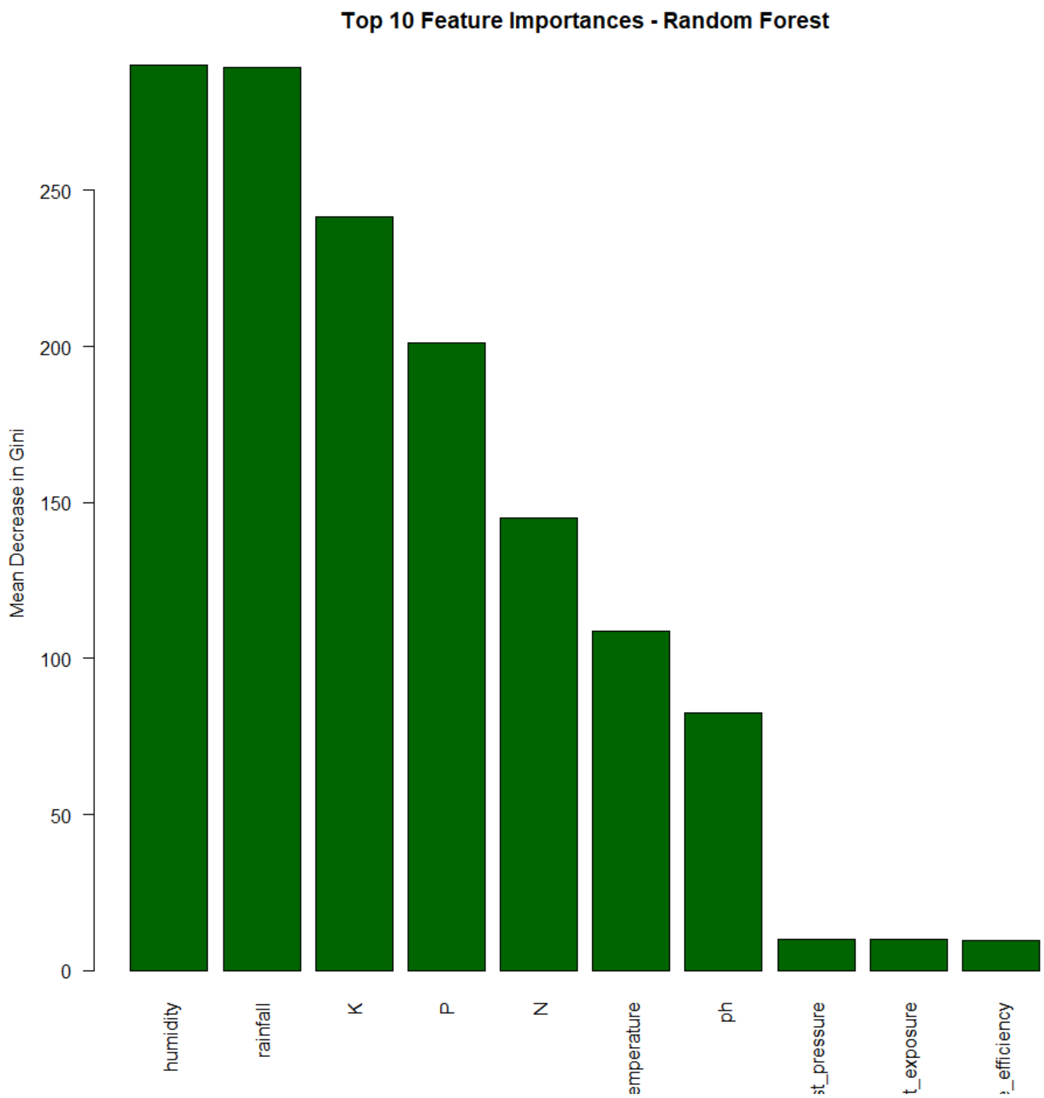
The results were also verified through confusion heatmaps showing that Random Forest performs best, with predictions tightly aligned along the diagonal, indicating high accuracy. QDA, LDA, and Neural Network also show strong performance with minimal misclassifications. SVM and Logistic Regression perform moderately, while KNN has the poorest performance, with scattered predictions and significant confusion between classes. These patterns visually support the reported accuracy metrics for each model.

The high performance of Random Forest led to its selection as the final model for prediction due to its robustness and ability to handle high-dimensional, multiclass data effectively.

Feature Selection

Random Forest

Rainfall, Humidity, and K (Potassium) are the most influential in determining the target class across your 21 categories followed by P (Phosphorus), N (Nitrogen), and Temperature. pH and Sunlight Exposure show some influence but significantly less than the top contribution. Water usage efficiency and Pest pressure contribute minimally.



The classification accuracy after retraining with the selected features is as follows:

```
## [1] 0.9969697
```

The Random Forest model, even with a reduced feature set, did not display an exceptional increase in accuracy. A minute increase from 99.55% to 99.69% was obtained.

Discussion

The results from the various models reveal significant differences in their performance, showcasing the importance of model selection in addressing the complexities of our high-dimensional and multiclass data. Logistic Regression, with a residual deviance of 0.00019, suggests a very good fit, although it may indicate overfitting or perfect class separation. The AIC of 966.00 supports this, reflecting the model's quality in relation to its complexity. However, it does not capture non-linear relationships, limiting its ability to handle the complexity in the data.

LDA, assumes equal covariance across classes and uses linear decision boundaries. Its simplicity made it effective when the data followed linear relationships, though it fell short when dealing with non-linear relationships in the amart farming data. QDA, by allowing for non-linear decision boundaries and estimating class-specific covariance, showed improved performance with a higher accuracy of 96.97%. This model demonstrated greater flexibility in handling more complex relationships, making it a better fit for our data.

The Random Forest model outperformed all others with the highest accuracy of 99.55%. This result highlights the model's ability to capture complex interactions between features, manage overfitting, and perform internal feature selection. Even with a reduced feature set, Random Forest showed minimal sensitivity to dimensionality, achieving a slight increase in accuracy from 99.55% to 99.69%. This robustness underscores its suitability for the high-dimensional, multiclass tasks associated with the data we chose, where it excels in aggregating multiple decision trees to reduce variance and increase predictive power.

In contrast, KNN performed poorly, with an accuracy of just 52.27%, likely due to the curse of dimensionality and the large number of classes (21) in the dataset. Distance-based methods like KNN can struggle when the feature space is high-dimensional and requires careful optimization, which was not the case here. SVM and Logistic Regression, while moderate in performance, were less effective in capturing non-linear patterns compared to Random Forest and Neural Networks.

Furthermore, the results suggest that ensemble and non-linear models, such as Random Forest, QDA, and Neural Networks, outperform simpler linear classifiers on the Smart Farming dataset. Random Forest, in particular, stood out for its robustness and reliability, justifying its selection as the final model for prediction. The model's superior performance, coupled with its ability to handle feature selection internally, makes it particularly effective for this multiclass classification task.

Finally, retraining Random Forest using the most consistently selected features highlighted the importance of feature selection in enhancing both model interpretability and computational efficiency. The model maintained its top-tier accuracy of 99.55% despite dimensionality reduction, reinforcing that many original features were redundant or contributed minimally to the classification task. These findings emphasize the value of feature selection in streamlining models without sacrificing predictive power. Thus, **Random Forest remains the best-performing model for this dataset**, owing to its superior accuracy, feature robustness, and ability to handle complex, high-dimensional data.

Conclusion

Our project directly supports real-life precision agriculture by enabling accurate, high-resolution classification of multiple crop types through utilization of high precision predictive models like RandomForest. It aids the improvement of crop monitoring, resource allocation, and decision-making. The model's robustness to high-dimensional data and its ability to maintain high accuracy with fewer features make it ideal for smart farming systems, ensuring cost-effective, scalable, and interpretable solutions for diverse agricultural environments.

Appendix

```
# library(tidyverse)
# library(corrplot)

# data <- read.csv("D://STAT414//midterm project//smartfarmingdata.csv")
# data <- data %>% mutate_if(is.character, as.factor)
# data <- na.omit(data)
# attach(data)

# library(tidyverse)
# library(kableExtra)
# library(tibble)
# library(DataExplorer)
# library(collapsibleTree)
# library(colorspace)

# datasummary <- introduce(data)
# plot_str(data)
# plot_bar(data)

# plot_histogram(data)

# cor_matrix <- cor(select(data, where(is.numeric)))
# corrplot(cor_matrix, method = "color")

# rf_importance <- importance(rf.fit)[, "MeanDecreaseGini"]
# top_rf_features <- sort(rf_importance, decreasing = TRUE)[1:10]  # Select
top 10
# rf_selected <- names(top_rf_features)
```

```r
# rf_reduced <- randomForest(label ~ ., data = train_scaled[, c(rf_selected,
"label")])

# summary(nnet.fit)

# summary(lda.fit)

# summary(qda.fit)

# summary(knn.fit)

# summary(rf.fit)

# summary(svm.fit)

# summary(nn.fit)

# print(accuracy_df)

# print(plot_accuracy)

# grid.arrange(p1, p2, p3, p4, p5, p6, p7, ncol = 3)

# # Barplot of top RF features
# barplot(top_rf_features,
#         las = 2, col = "darkgreen",
#         main = "Top 10 Feature Importances - Random Forest",
#         ylab = "Mean Decrease in Gini")

# rf2.pred <- predict(rf_reduced, newdata = test_scaled)
# rf2.acc <- mean(rf2.pred == true_labels)
# rf2.acc
```