Parshvanath Charitable Trust's

A. P. SHAH INSTITUTE OF TECHNOLOGY

(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Department of Information Technology

Academic Year: 2020-21
Semester: VIII
Class / Branch: BEIT
Subject: R Programming

Name of Student: Tejas khanted
Student ID: 17104015
Date of Performance: 12/5/2021
Date of Submission: 12/5/2021

MINI PROJECT

Aim: Crop Recommendation in R Programming

Data Set (csv file) -

	A	B	C	D	E	F	G	H
	N	P	K	temperat	humidity	ph	rainfall	Crop
	90	42	43	20.87974	82.00274	6.502985	202.9355	rice
	85	58	41	21.77046	80.31964	7.038096	226.6555	rice
	60	55	44	23.00446	82.32076	7.840207	263.9642	rice
	74	35	40	26.4911	80.15836	6.980401	242.864	rice
	78	42	42	20.13017	81.60487	7.628473	262.7173	rice
	69	37	42	23.05805	83.37012	7.073454	251.055	rice
	69	55	38	22.70884	82.63941	5.700806	271.3249	rice
	94	53	40	20.27774	82.89409	5.718627	241.9742	rice
0	89	54	38	24.51588	83.53522	6.685346	230.4462	rice
1	68	58	38	23.22397	83.03323	6.336254	221.2092	rice
2	91	53	40	26.52724	81.41754	5.386168	264.6149	rice
3	90	46	42	23.97898	81.45062	7.502834	250.0832	rice
4	78	58	44	26.8008	80.88685	5.108682	284.4365	rice
5	93	56	36	24.01498	82.05687	6.984354	185.2773	rice
6	94	50	37	25.66585	80.66385	6.94802	209.587	rice
7	60	48	39	24.28209	80.30026	7.042299	231.0863	rice
8	85	38	41	21.58712	82.78837	6.249051	276.6552	rice
9	91	35	39	23.79392	80.41818	6.97086	206.2612	rice
0	77	38	36	21.86525	80.1923	5.953933	224.555	rice
1	88	35	40	23.57944	83.5876	5.853932	291.2987	rice
2	89	45	36	21.32504	80.47476	6.442475	185.4975	rice

Code -

```
data <- read.csv("F:/Downloads/crop data/Crop_recommendation.csv" , stringsAsFactors =TRUE)

sapply(data, class)

head(data)

summary(data)

library(caret)

# split input and output

x <- data[,1:7]

y <- data[,8]

# boxplot for each attribute on one image

par(mfrow=c(1,7))

for(i in 1:7) {

 boxplot(x[,i], main=names(data)[i])

}

# box and whisker plots for each attribute

featurePlot(x=x, y=y, plot="box")

#Data Partition

#index <- createDataPartition(data$Crop, p = .75, list = FALSE)

#train <- data[index, ]

#test <- data[-index, ]

 #Data Randomly Partiiton

set.seed(2)
```

```
dummy_sep <- rbinom(nrow(data) , 1 ,0.7)

test  <- data[dummy_sep == 0, ]

train <- data[dummy_sep == 1, ]

control <- trainControl(method="cv", number=10)

metric <- "Accuracy"

# CART

set.seed(7)

fit.cart <- train(Crop~., data=data, method="rpart", metric=metric,
trControl=control)

# kNN

set.seed(7)

fit.knn <- train(Crop~., data=data, method="knn", metric=metric,
trControl=control)

# c) advanced algorithms

# SVM

set.seed(7)

fit.svm <- train(Crop~., data=data, method="svmRadial", metric=metric,
trControl=control)

# Random Forest

set.seed(7)

fit.rf <- train(Crop~., data=data, method="rf", metric=metric, trControl=control)

#summarize accuracy of models

results <- resamples(list(cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf))

summary(results)
```

```
# compare accuracy of models

dotplot(results)

# summarize Best Model

print(fit.rf)

# estimate skill of RF on the validation dataset

predictions <- predict(fit.rf, test)

#predictions

test$PredictedCrop <- predictions

newdata <- data.frame(N = 10 , P=26, K= 43 ,temperature = 20 , humidity = 25,ph
= 3.0734 , rainfall = 100.0550 )

newdata1 <-data.frame(N =25    ,P = 78,        K=76,temperature =
17.48042641,humidity =  15.7559405, ph =   7.228963452 , rainfall =
         66.96980581          )

# Predict values

Predited <- predict(fit.rf, newdata1)

Predited

Preditedd <- predict(fit.rf, newdata)

Preditedd

# Predicting values

newdata$PredictedCrop <- Preditedd

newdata1$PredictedCrop <- Predited
```

Output -

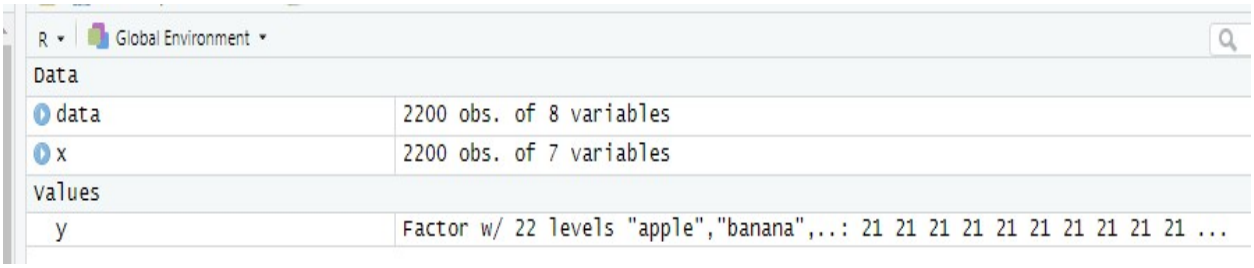
Executing Code -

- **Load the data, to get an idea of the types of the attributes, Peek at Data and Statistical Summary -**

```
> data <- read.csv("F:/Downloads/crop data/Crop_recommendation.csv" , stringsAsFactors =TRUE)
> sapply(data, class)
          N           P           K temperature    humidity          ph    rainfall        Crop
  "integer"   "integer"   "integer"   "numeric"   "numeric"   "numeric"   "numeric"    "factor"
> head(data)
   N  P  K temperature humidity       ph rainfall Crop
1 90 42 43    20.87974 82.00274 6.502985 202.9355 rice
2 85 58 41    21.77046 80.31964 7.038096 226.6555 rice
3 60 55 44    23.00446 82.32076 7.840207 263.9642 rice
4 74 35 40    26.49110 80.15836 6.980401 242.8640 rice
5 78 42 42    20.13017 81.60487 7.628473 262.7173 rice
6 69 37 42    23.05805 83.37012 7.073454 251.0550 rice
> summary(data)
       N                P                K           temperature       humidity           ph
 Min.   :  0.00   Min.   :  5.00   Min.   :  5.00   Min.   : 8.826   Min.   :14.26   Min.   :3.505
 1st Qu.: 21.00   1st Qu.: 28.00   1st Qu.: 20.00   1st Qu.:22.769   1st Qu.:60.26   1st Qu.:5.972
 Median : 37.00   Median : 51.00   Median : 32.00   Median :25.599   Median :80.47   Median :6.425
 Mean   : 50.55   Mean   : 53.36   Mean   : 48.15   Mean   :25.616   Mean   :71.48   Mean   :6.469
 3rd Qu.: 84.25   3rd Qu.: 68.00   3rd Qu.: 49.00   3rd Qu.:28.562   3rd Qu.:89.95   3rd Qu.:6.924
 Max.   :140.00   Max.   :145.00   Max.   :205.00   Max.   :43.675   Max.   :99.98   Max.   :9.935

    rainfall            Crop
 Min.   : 20.21   apple     : 100
 1st Qu.: 64.55   banana    : 100
 Median : 94.87   blackgram: 100
 Mean   :103.46   chickpea  : 100
 3rd Qu.:124.27   coconut   : 100
 Max.   :298.56   coffee    : 100
                  (Other)   :1600
```

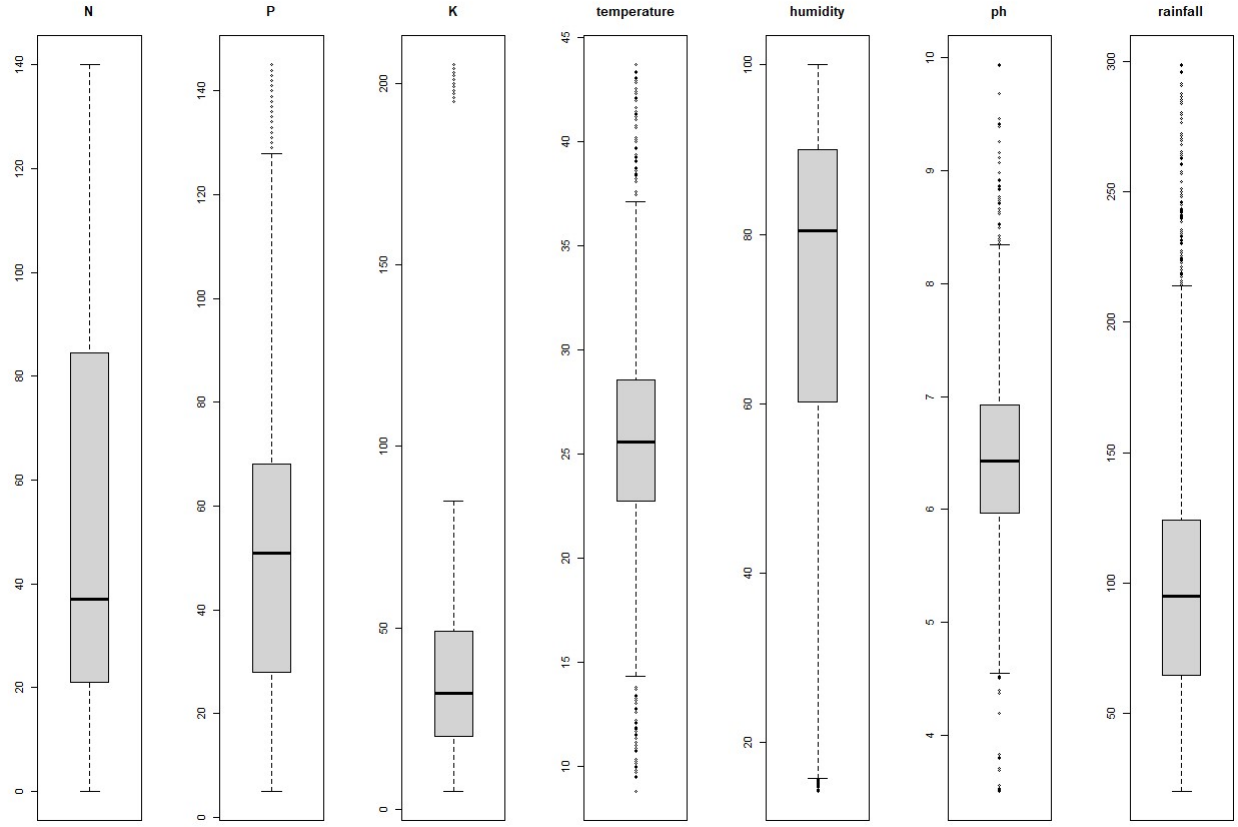
- **Visualize the Data Set via Caret Library -**
  Split the input and output

```
> library(caret)
Loading required package: lattice
Loading required package: ggplot2
> x <- data[,1:7]
> y <- data[,8]
>
```



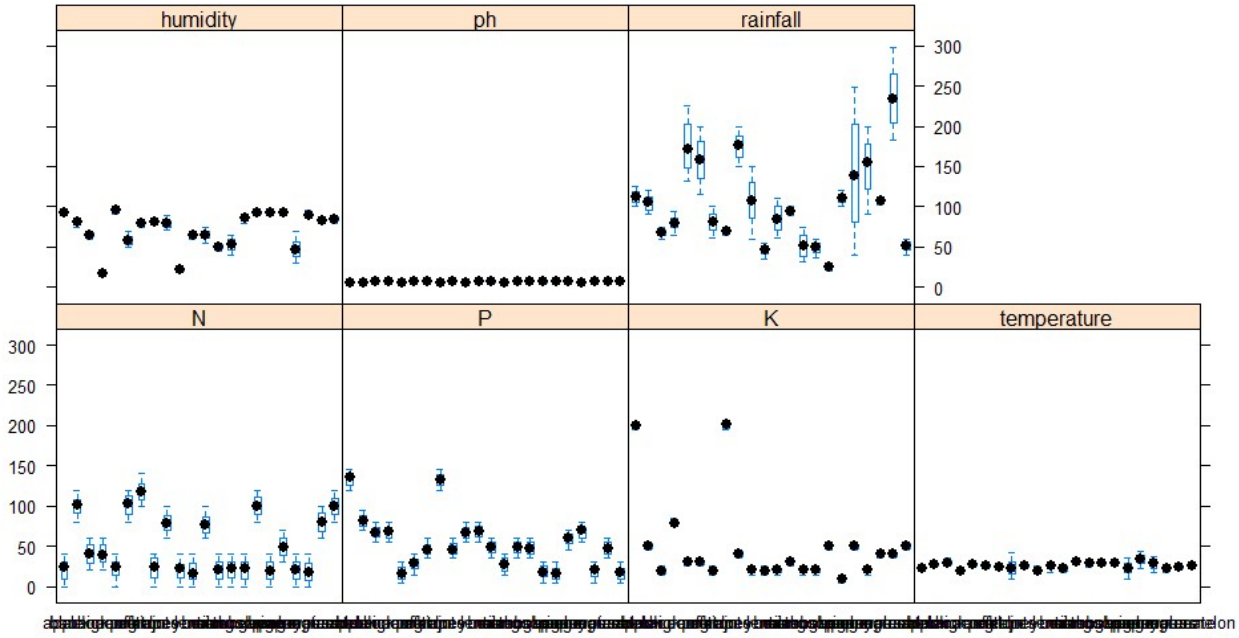
- **Creating boxplot for each attribute on one image -**

```
> # boxplot for each attribute on one image
> par(mfrow=c(1,7))
> for(i in 1:7) {
+   boxplot(x[,i], main=names(data)[i])
+ }
```

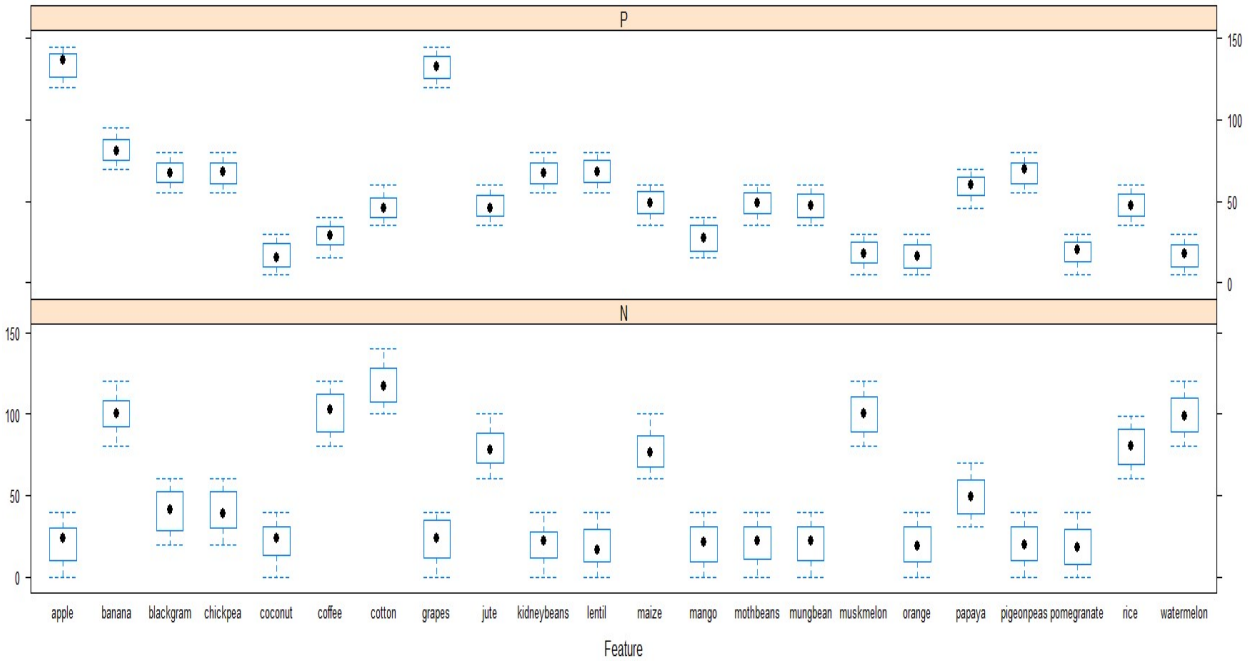


- **Box and whisker plots for each attribute -**

```
> featurePlot(x=x, y=y, plot="box")
> |
```

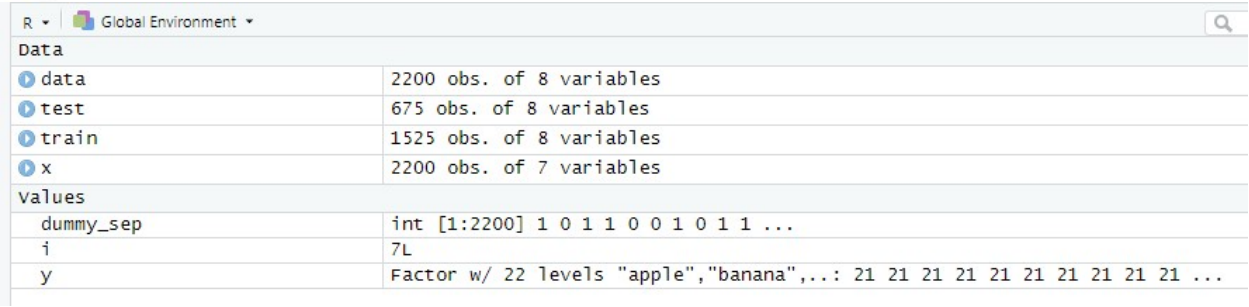


Box and Whisker Plot of data by Class Value



- **Data Randomly Partition for Machine Learning Prediction -**

```
>  #Data Randomly Partiiton
> set.seed(2)
> dummy_sep <- rbinom(nrow(data) , 1 ,0.7)
> test  <- data[dummy_sep == 0, ]
> train <- data[dummy_sep == 1, ]
>
```



R ▾ | Global Environment ▾

Data	
data	2200 obs. of 8 variables
test	675 obs. of 8 variables
train	1525 obs. of 8 variables
x	2200 obs. of 7 variables
Values	
dummy_sep	int [1:2200] 1 0 1 1 0 0 1 0 1 1 ...
i	7L
y	Factor w/ 22 levels "apple","banana",..: 21 21 21 21 21 21 21 21 21 21 ...

- **Test Harness -**
  Run algorithms using 10-fold cross validation

```
> control <- trainControl(method="cv", number=10)
> metric <- "Accuracy"
>
```

- **Build Models -**

```
> # CART
> set.seed(7)
> fit.cart <- train(Crop~., data=data, method="rpart", metric=metric, trControl=control)
> # kNN
> set.seed(7)
> fit.knn <- train(Crop~., data=data, method="knn", metric=metric, trControl=control)
> # c) advanced algorithms
> # SVM
> set.seed(7)
> fit.svm <- train(Crop~., data=data, method="svmRadial", metric=metric, trControl=control)
> # Random Forest
> set.seed(7)
> fit.rf <- train(Crop~., data=data, method="rf", metric=metric, trControl=control)
>
```

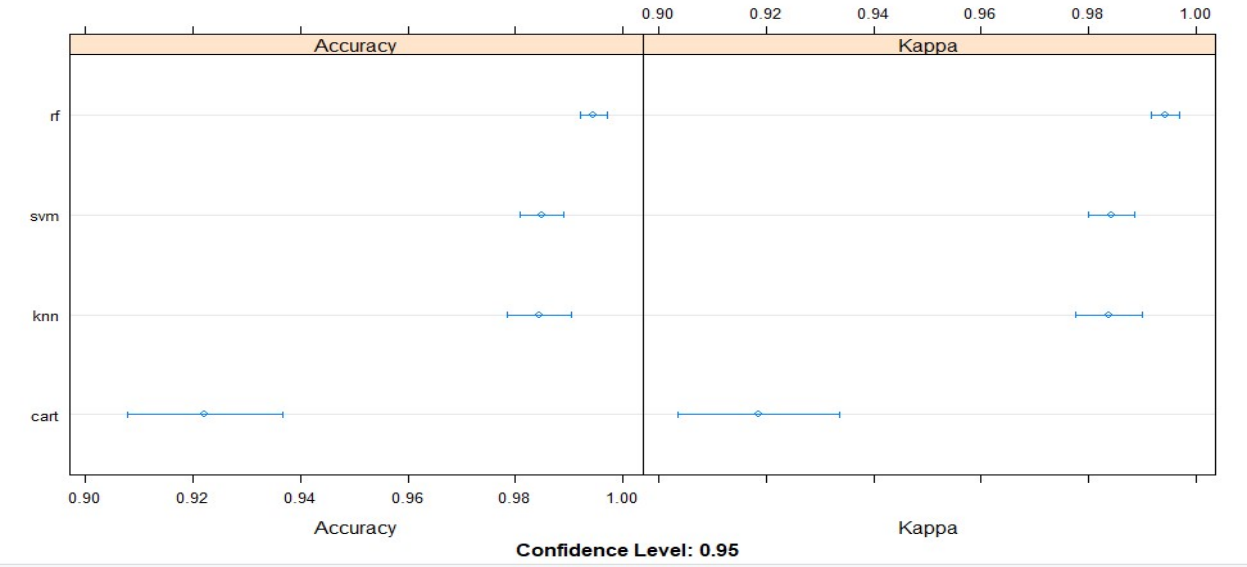
Data	
control	List of 27
data	2200 obs. of 8 variables
fit.cart	List of 23
fit.knn	List of 23
fit.rf	List of 23
fit.svm	Large train (23 elements, 1.8 MB)
results	List of 6
test	675 obs. of 8 variables
train	1525 obs. of 8 variables
x	2200 obs. of 7 variables

- **Summarize accuracy of Models -**

```
> #summarize accuracy of models
> results <- resamples(list(cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf))
> summary(results)

Call:
summary.resamples(object = results)

Models: cart, knn, svm, rf
Number of resamples: 10

Accuracy
          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
cart 0.8863636 0.9159091 0.9250000 0.9222727 0.9306818 0.9545455    0
knn  0.9681818 0.9784091 0.9863636 0.9845455 0.9909091 0.9954545    0
svm  0.9772727 0.9818182 0.9818182 0.9850000 0.9897727 0.9954545    0
rf   0.9863636 0.9954545 0.9954545 0.9945455 0.9954545 1.0000000    0

Kappa
          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
cart 0.8809524 0.9119048 0.9214286 0.9185714 0.9273810 0.9523810    0
knn  0.9666667 0.9773810 0.9857143 0.9838095 0.9904762 0.9952381    0
svm  0.9761905 0.9809524 0.9809524 0.9842857 0.9892857 0.9952381    0
rf   0.9857143 0.9952381 0.9952381 0.9942857 0.9952381 1.0000000    0

>
```

- **Compare accuracy of models -**

```
> # compare accuracy of models
> dotplot(results)
>
```



- **Summarize the best model ie. Random Forest -**

```
> # summarize Best Model
> print(fit.rf)
Random Forest

2200 samples
   7 predictor
  22 classes: 'apple', 'banana', 'blackgram', 'chickpea', 'coconut', 'coffee', 'cotton', 'grapes', 'jute',
 'kidneybeans', 'lentil', 'maize', 'mango', 'mothbeans', 'mungbean', 'muskmelon', 'orange', 'papaya', 'pig
eonpeas', 'pomegranate', 'rice', 'watermelon'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1980, 1980, 1980, 1980, 1980, 1980, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.9945455  0.9942857
  4     0.9927273  0.9923810
  7     0.9927273  0.9923810

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
> |
```

- **Now Prediction of Test data -**

```
> # estimate skill of RF on the validation dataset
> predictions <- predict(fit.rf, test)
> #predictions
> test$PredictedCrop <- predictions
> |
```

	N	P	K	temperature	humidity	ph	rainfall	Crop	PredictedCrop
2	85	58	41	21.77046	80.31964	7.038096	226.65554	rice	rice
5	78	42	42	20.13017	81.60487	7.628473	262.71734	rice	rice
6	69	37	42	23.05805	83.37012	7.073454	251.05500	rice	rice
8	94	53	40	20.27774	82.89409	5.718627	241.97419	rice	rice
13	78	58	44	26.80080	80.88685	5.108682	284.43646	rice	rice
16	60	48	39	24.28209	80.30026	7.042299	231.08633	rice	rice
17	85	38	41	21.58712	82.78837	6.249051	276.65525	rice	rice
23	67	59	41	21.94767	80.97384	6.012633	213.35609	rice	rice
29	60	49	44	20.77576	84.49774	6.244841	240.08106	rice	rice
33	85	37	39	24.52784	82.73686	6.364135	224.67572	rice	rice
34	98	53	38	20.26708	81.63895	5.014507	270.44173	rice	rice
37	99	57	35	26.75754	81.17734	5.960370	272.29991	rice	rice
41	62	42	36	22.78134	82.06719	6.430010	248.71832	rice	rice
45	85	52	45	26.31355	82.36699	7.224286	265.53559	rice	rice
46	91	35	38	24.89728	80.52586	6.134287	183.67932	rice	rice
47	76	49	42	24.95878	84.47963	5.206373	196.95600	rice	rice

- **Prediction on User Defined attributes value -**

```
> newdata <- data.frame(N = 10 , P=26, K= 43 ,temperature = 20 , humidity = 25,ph = 3.0734 , rainfall = 10
0.0550 )
> # Predict values
> Predited <- predict(fit.rf, newdata)
> Predited
[1] kidneybeans
22 Levels: apple banana blackgram chickpea coconut coffee cotton grapes jute kidneybeans ... watermelon
```

Filter

	N	P	K	temperature	humidity	ph	rainfall	PredictedCrop
1	10	26	43	20	25	3.0734	100.055	kidneybeans

```
> newdata1 <-data.frame(N =25   ,P = 78,        K=76,   temperature = 17.48042641,humidity =    15.755940
5, ph = 7.228963452 , rainfall =        66.96980581     )
> # Predict values
> Predited <- predict(fit.rf, newdata1)
> Predited
[1] chickpea
22 Levels: apple banana blackgram chickpea coconut coffee cotton grapes jute kidneybeans ... watermelon
>
```

Filter

	N	P	K	temperature	humidity	ph	rainfall	PredictedCrop
1	25	78	76	17.48043	15.75594	7.228963	66.96981	chickpea

Conclusion - Crop are Recommended by using Random Forest Algorithm on test data and also on User defined Attribute with parameters like N, P, Temperature, Humidity, ph and rainfall.