# SVMRtext

Rae Zhang

11/24/2021

```
library(tm)
```

```
## Loading required package: NLP
```

```
library(stringr)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(SnowballC)
library(arules)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

```
## The following object is masked from 'package:tm':
##
##     inspect
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(cluster)
library(stringi)
library(Matrix)
library(tidytext)
library(plyr)
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(mclust)
```

```
## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.1.5     v purrr   0.3.4
## v tidyr   1.1.3     v dplyr   1.0.5
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x ggplot2::annotate() masks NLP::annotate()
## x dplyr::arrange()    masks plyr::arrange()
## x purrr::compact()    masks plyr::compact()
## x dplyr::count()      masks plyr::count()
## x tidyr::expand()     masks Matrix::expand()
## x dplyr::failwith()   masks plyr::failwith()
## x dplyr::filter()     masks stats::filter()
## x dplyr::id()         masks plyr::id()
## x dplyr::lag()        masks stats::lag()
## x purrr::map()        masks mclust::map()
## x dplyr::mutate()     masks plyr::mutate()
## x tidyr::pack()       masks Matrix::pack()
## x dplyr::recode()     masks arules::recode()
## x dplyr::rename()     masks plyr::rename()
## x dplyr::summarise()  masks plyr::summarise()
## x dplyr::summarize()  masks plyr::summarize()
## x tidyr::unpack()     masks Matrix::unpack()
```

```r
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(caretEnsemble)
```

```
##
## Attaching package: 'caretEnsemble'

## The following object is masked from 'package:ggplot2':
##
##     autoplot
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:mclust':
##
##     sim

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(Amelia)
```

```
## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.0, built: 2021-05-26)
## ## Copyright (C) 2005-2021 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
library(mice)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(e1071)
library(ggthemes)
library(Cairo)
library(network)
```

```
##
## 'network' 1.17.1 (2021-06-12), part of the Statnet Project
## * 'news(package="network")' for changes since last version
## * 'citation("network")' for citation information
## * 'https://statnet.org' for help, support, and other information
```

```
##
## Attaching package: 'network'
```

```
## The following object is masked from 'package:plyr':
##
##     is.discrete
```

```
library(ggtext)
library(readxl)
library(RColorBrewer)
library(slam)
library(proxy)
```

```
##
## Attaching package: 'proxy'
```

```
## The following object is masked from 'package:Matrix':
##
##     as.matrix
```

```
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix
```

```
library(stringr)
library(textmineR)
```

```
##
## Attaching package: 'textmineR'
## The following object is masked from 'package:Matrix':
##
##     update

## The following object is masked from 'package:stats':
##
##     update
```

```
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:network':
##
##     %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,
##     get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite,
##     is.directed, list.edge.attributes, list.vertex.attributes,
##     set.edge.attribute, set.vertex.attribute

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:purrr':
##
##     compose, simplify

## The following object is masked from 'package:tidyr':
##
##     crossing

## The following object is masked from 'package:tibble':
##
##     as_data_frame

## The following object is masked from 'package:arules':
##
##     union

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
library(klaR)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
head(CropDF<-read.csv("/Users/raezh1/Documents/Georgetown/ANLY501/assignment_5new/files/Crop_recommenda
```

```
##    N  P  K temperature humidity humidity_level       ph rainfall label
## 1 90 42 43    20.87974 82.00274           High 6.502985 202.9355  rice
## 2 85 58 41    21.77046 80.31964           High 7.038096 226.6555  rice
## 3 60 55 44    23.00446 82.32076           High 7.840207 263.9642  rice
## 4 74 35 40    26.49110 80.15836           High 6.980401 242.8640  rice
## 5 78 42 42    20.13017 81.60487           High 7.628473 262.7173  rice
## 6 69 37 42    23.05805 83.37012           High 7.073454 251.0550  rice
```

## Make test and train data

## Testing data

### Change data type

```
str(CropDF)
```

```
## 'data.frame':    2200 obs. of  9 variables:
##  $ N             : int  90 85 60 74 78 69 69 94 89 68 ...
##  $ P             : int  42 58 55 35 42 37 55 53 54 58 ...
##  $ K             : int  43 41 44 40 42 42 38 40 38 38 ...
##  $ temperature   : num  20.9 21.8 23 26.5 20.1 ...
##  $ humidity      : num  82 80.3 82.3 80.2 81.6 ...
##  $ humidity_level: chr  "High" "High" "High" "High" ...
##  $ ph            : num  6.5 7.04 7.84 6.98 7.63 ...
##  $ rainfall      : num  203 227 264 243 263 ...
##  $ label         : chr  "rice" "rice" "rice" "rice" ...
```

```
CropDF$N <- as.numeric(CropDF$N)
CropDF$P <- as.numeric(CropDF$P)
CropDF$K <- as.numeric(CropDF$K)
CropDF$label <- as.factor(CropDF$label)
str(CropDF)
```

```
## 'data.frame':    2200 obs. of  9 variables:
##  $ N             : num  90 85 60 74 78 69 69 94 89 68 ...
##  $ P             : num  42 58 55 35 42 37 55 53 54 58 ...
##  $ K             : num  43 41 44 40 42 42 38 40 38 38 ...
##  $ temperature   : num  20.9 21.8 23 26.5 20.1 ...
##  $ humidity      : num  82 80.3 82.3 80.2 81.6 ...
##  $ humidity_level: chr  "High" "High" "High" "High" ...
##  $ ph            : num  6.5 7.04 7.84 6.98 7.63 ...
##  $ rainfall      : num  203 227 264 243 263 ...
##  $ label         : Factor w/ 22 levels "apple","banana",..: 21 21 21 21 21 21 21 21 21 21 ...
```

```
CropDF <- subset(CropDF, select=-c(6))

(Size <- (as.integer(nrow(CropDF)/4)))  ## Test will be 1/4 of the data
```

```
## [1] 550
```

```
SAMPLE <- sample(nrow(CropDF), Size, replace = FALSE)

DF_Test_Crop<-CropDF[SAMPLE, ]
DF_Train_Crop<-CropDF[-SAMPLE, ]
```

### Remove the labels and store them

```
DF_Test_Crop_Labels <- DF_Test_Crop$label
```

### Remove the labels

```
DF_Test_Crop_NL<-DF_Test_Crop[ , -which(names(DF_Test_Crop) %in% c("label"))]
```

### Check size

```
(nrow(DF_Test_Crop_NL))
```

```
## [1] 550
```

## Training data

### Copy the Labels

```
DF_Train_Crop_Labels <- DF_Train_Crop$label
```

### Remove the labels

```
DF_Train_Crop_NL<-DF_Train_Crop[ , -which(names(DF_Train_Crop) %in% c("label"))]
head(DF_Train_Crop_NL)
```

```
##    N  P  K temperature humidity       ph rainfall
## 1 90 42 43    20.87974 82.00274 6.502985 202.9355
## 2 85 58 41    21.77046 80.31964 7.038096 226.6555
## 3 60 55 44    23.00446 82.32076 7.840207 263.9642
## 5 78 42 42    20.13017 81.60487 7.628473 262.7173
## 7 69 55 38    22.70884 82.63941 5.700806 271.3249
## 8 94 53 40    20.27774 82.89409 5.718627 241.9742
```

### Check size

```
(nrow(DF_Train_Crop_NL))
```

```
## [1] 1650
```

## "tune" the SVM by altering the cost

```
tuned_cost <- tune(svm, label~., data=DF_Train_Crop,
                   kernel="linear",
                   ranges=list(cost=c(.01,.1,1,10,100,1000)))
summary(tuned_cost)  ## This shows that the best cost is 100
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##    10
##
## - best performance: 0.01333333
##
## - Detailed performance results:
##    cost      error dispersion
## 1 1e-02 0.20666667 0.04877439
## 2 1e-01 0.02363636 0.01523881
## 3 1e+00 0.01393939 0.01279284
## 4 1e+01 0.01333333 0.01333945
## 5 1e+02 0.01454545 0.01076599
## 6 1e+03 0.01454545 0.01184879
```

## Set up the SVM

### Polynomial Kernel

```r
str(DF_Train_Crop)
```

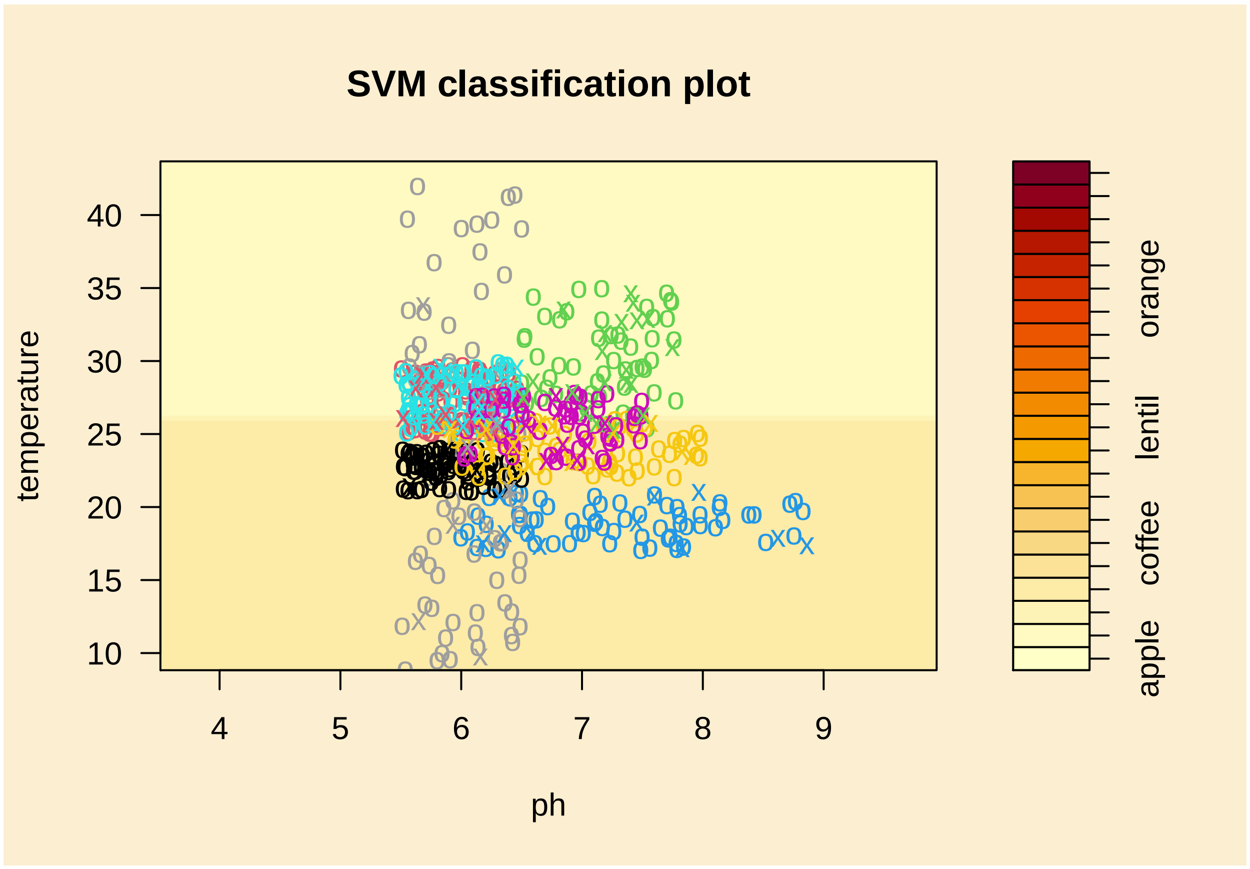
```
## 'data.frame':    1650 obs. of  8 variables:
##  $ N          : num  90 85 60 78 69 94 89 68 90 78 ...
##  $ P          : num  42 58 55 42 55 53 54 58 46 58 ...
##  $ K          : num  43 41 44 42 38 40 38 38 42 44 ...
##  $ temperature: num  20.9 21.8 23 20.1 22.7 ...
##  $ humidity   : num  82 80.3 82.3 81.6 82.6 ...
##  $ ph         : num  6.5 7.04 7.84 7.63 5.7 ...
##  $ rainfall   : num  203 227 264 263 271 ...
##  $ label      : Factor w/ 22 levels "apple","banana",..: 21 21 21 21 21 21 21 21 21 21 ...
```

```r
SVM_fit_P <- svm(label~., data=DF_Train_Crop,
               kernel="polynomial", cost=100,
               scale=FALSE)
print(SVM_fit_P)
```

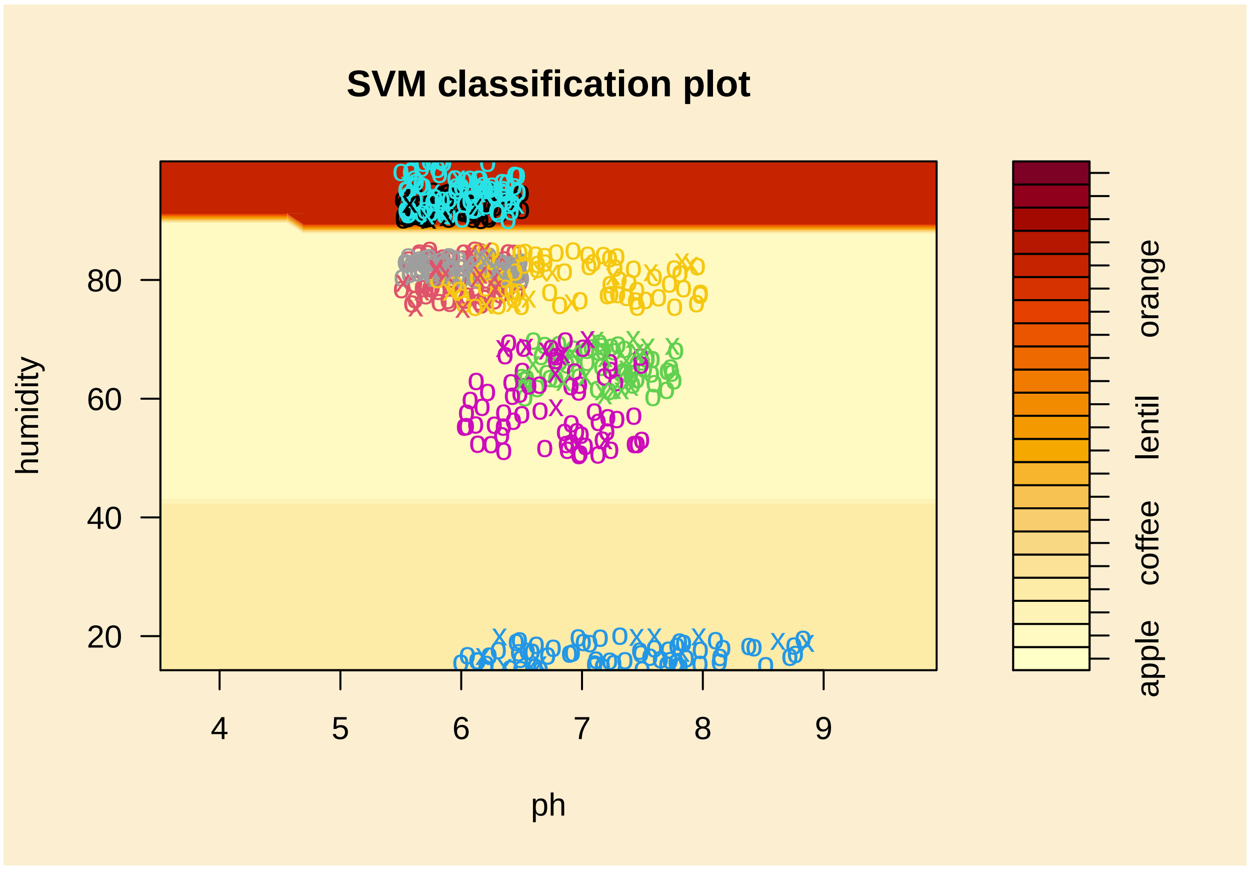
```
##
## Call:
## svm(formula = label ~ ., data = DF_Train_Crop, kernel = "polynomial",
##     cost = 100, scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  100
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  340
```

## Prediction

```r
pred_P <- predict(SVM_fit_P, DF_Test_Crop_NL, type="class")
par(bg = "#fbeed1")
plot(SVM_fit_P, data=DF_Train_Crop, temperature~ph, slice=list(humidity=42,N=70,P=70,K=100,rainfall=150
```



```
plot(SVM_fit_P, data=DF_Train_Crop, humidity~ph, slice=list(temperature=20,N=70,P=70,K=100,rainfall=150)
```



## Create a function that generates heatmap from the confusion matrix

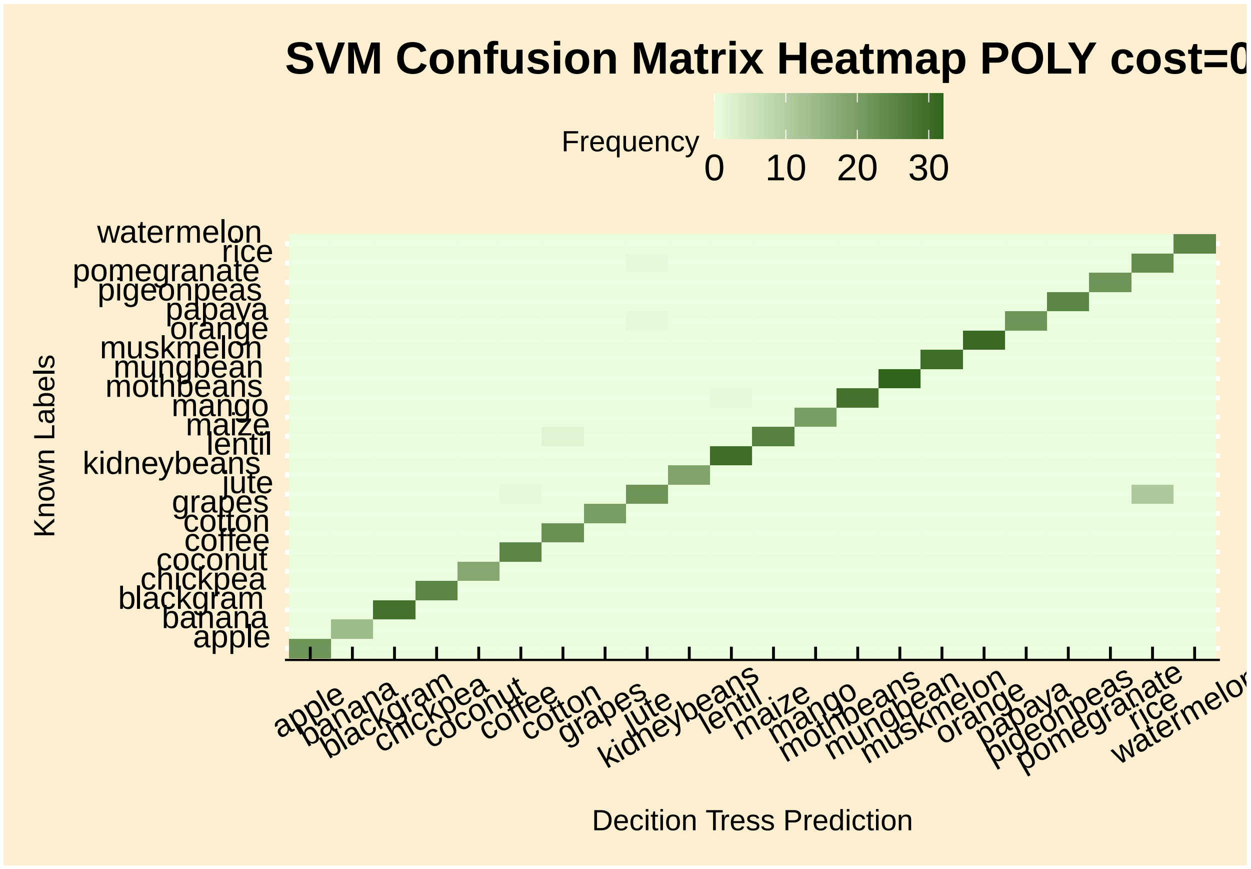
```
get_heatmap <- function(mapname, prediction){
  data <- as.data.frame(table(prediction,DF_Test_Crop_Labels))
plot <- ggplot(data) +
  geom_tile(mapping=aes(x=data[,1], y=data[,2],fill=data[,3])) +
  ylab("Known Labels") +
  xlab("Decition Tress Prediction") +
  theme_economist() +
  ggtitle(mapname) +
  scale_fill_gradient2(name="Frequency",low="#defccf", mid="#e9ffdfe6", high="#32641b") +
  theme(plot.background = element_rect(fill='#fbeed1',color="#fbeed1"),
        legend.background =element_rect(fill='#fbeed1',color="#fbeed1"),
        axis.text.x = element_markdown(size=12, angle = 30, vjust = 0.9, hjust=.6),
        axis.text.y = element_markdown(size=12, angle = 0, vjust = 0.2, hjust=1.1))
return(plot)
}
```

### Confusion Matrix

```
(Ptable <- table(pred_P, DF_Test_Crop_Labels))
```

```
##                DF_Test_Crop_Labels
## pred_P          apple banana blackgram chickpea coconut coffee cotton grapes jute
##   apple          22      0         0        0       0      0      0      0    0
##   banana          0     14         0        0       0      0      0      0    0
##   blackgram       0      0        29        0       0      0      0      0    0
##   chickpea        0      0         0       25       0      0      0      0    0
##   coconut         0      0         0        0      18      0      0      0    0
##   coffee          0      0         0        0       0     25      0      0    1
##   cotton          0      0         0        0       0      0     23      0    0
##   grapes          0      0         0        0       0      0      0     20    0
##   jute            0      0         0        0       0      0      0      0   22
##   kidneybeans     0      0         0        0       0      0      0      0    0
##   lentil          0      0         0        0       0      0      0      0    0
##   maize           0      0         0        0       0      0      0      0    0
##   mango           0      0         0        0       0      0      0      0    0
##   mothbeans       0      0         0        0       0      0      0      0    0
##   mungbean        0      0         0        0       0      0      0      0    0
##   muskmelon       0      0         0        0       0      0      0      0    0
##   orange          0      0         0        0       0      0      0      0    0
##   papaya          0      0         0        0       0      0      0      0    0
##   pigeonpeas      0      0         0        0       0      0      0      0    0
##   pomegranate     0      0         0        0       0      0      0      0    0
##   rice            0      0         0        0       0      0      0      0   11
##   watermelon      0      0         0        0       0      0      0      0    0
##              DF_Test_Crop_Labels
## pred_P        kidneybeans lentil maize mango mothbeans mungbean muskmelon
##   apple                 0      0     0     0         0        0         0
##   banana                0      0     0     0         0        0         0
##   blackgram             0      0     0     0         0        0         0
##   chickpea              0      0     0     0         0        0         0
##   coconut               0      0     0     0         0        0         0
##   coffee                0      0     0     0         0        0         0
##   cotton                0      0     2     0         0        0         0
##   grapes                0      0     0     0         0        0         0
##   jute                  0      0     0     0         0        0         0
##   kidneybeans          19      0     0     0         0        0         0
##   lentil                0     30     0     0         1        0         0
##   maize                 0      0    26     0         0        0         0
##   mango                 0      0     0    20         0        0         0
##   mothbeans             0      0     0     0        29        0         0
##   mungbean              0      0     0     0         0       32         0
##   muskmelon             0      0     0     0         0        0        30
##   orange                0      0     0     0         0        0         0
##   papaya                0      0     0     0         0        0         0
##   pigeonpeas            0      0     0     0         0        0         0
##   pomegranate           0      0     0     0         0        0         0
##   rice                  0      0     0     0         0        0         0
##   watermelon            0      0     0     0         0        0         0
##              DF_Test_Crop_Labels
## pred_P        orange papaya pigeonpeas pomegranate rice watermelon
##   apple            0      0          0           0    0          0
##   banana           0      0          0           0    0          0
##   blackgram        0      0          0           0    0          0
##   chickpea         0      0          0           0    0          0
##   coconut          0      0          0           0    0          0
##   coffee           0      0          0           0    0          0
##    cotton           0       0           0           0    0           0
##    grapes           0       0           0           0    0           0
##    jute             0       1           0           0    1           0
##    kidneybeans      0       0           0           0    0           0
##    lentil           0       0           0           0    0           0
##    maize            0       0           0           0    0           0
##    mango            0       0           0           0    0           0
##    mothbeans        0       0           0           0    0           0
##    mungbean         0       0           0           0    0           0
##    muskmelon        0       0           0           0    0           0
##    orange          31       0           0           0    0           0
##    papaya           0      22           0           0    0           0
##    pigeonpeas       0       0          25           0    0           0
##    pomegranate      0       0           0          22    0           0
##    rice             0       0           0           0   24           0
##    watermelon       0       0           0           0    0          25
```

```
get_heatmap("SVM Confusion Matrix Heatmap POLY cost=0.1", pred_P)
```



## Misclassification Rate for Polynomial

```
(MR_P <- 1 - sum(diag(Ptable))/sum(Ptable))
```

```
## [1] 0.03090909
```

## Linear Kernel…

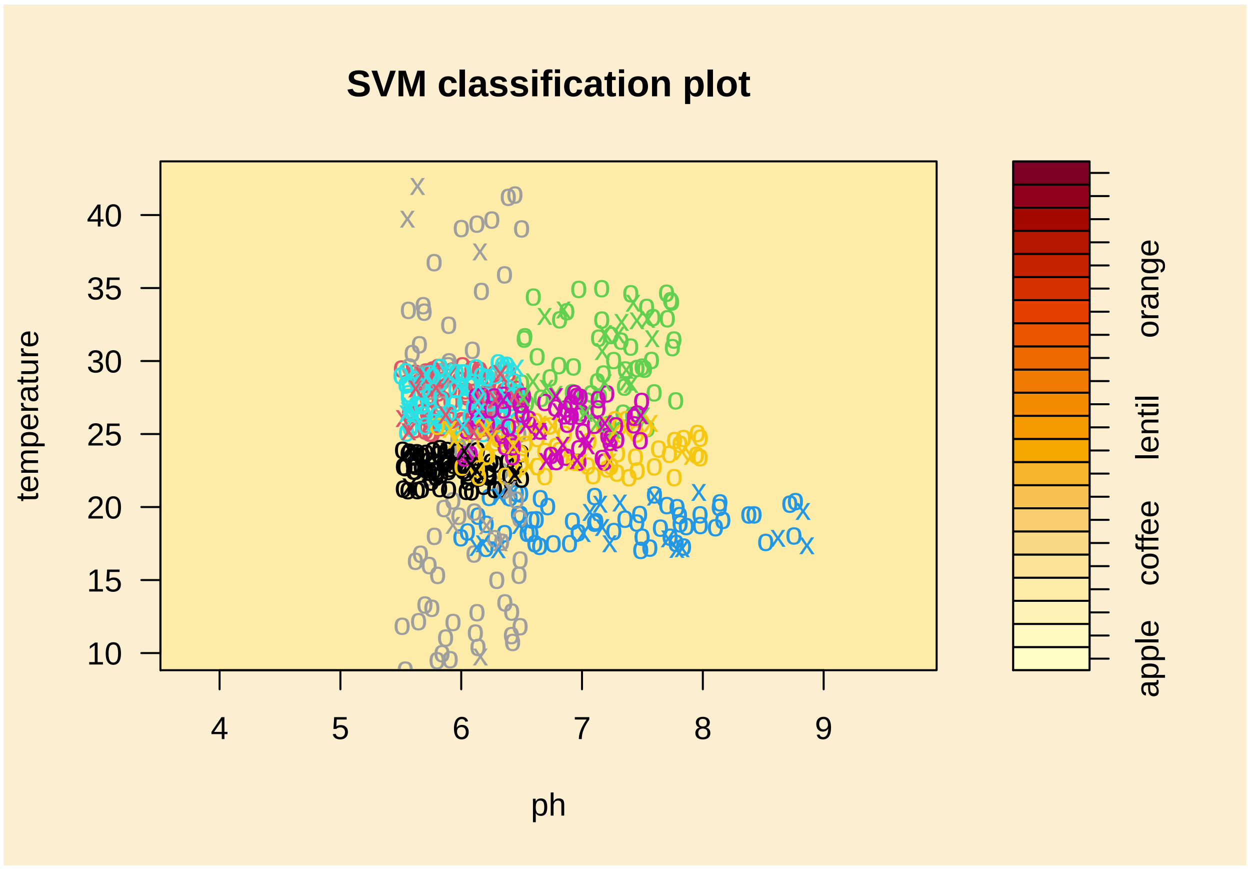
```
SVM_fit_L <- svm(label~., data=DF_Train_Crop,
                 kernel="linear", cost=10,
                 scale=FALSE)
print(SVM_fit_L)
```

```
##
## Call:
## svm(formula = label ~ ., data = DF_Train_Crop, kernel = "linear",
##     cost = 10, scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##
## Number of Support Vectors:  382
```

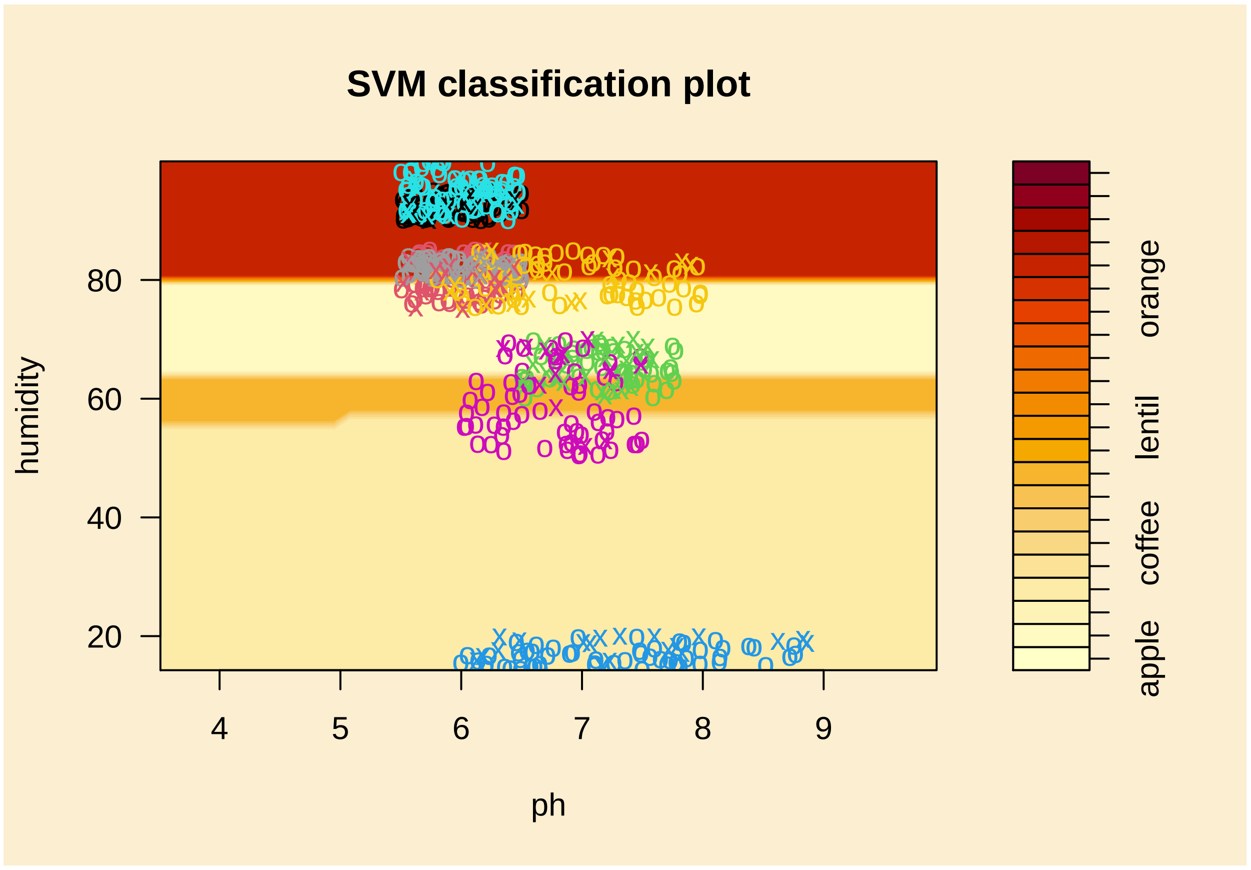
## Prediction

```
pred_L <- predict(SVM_fit_L, DF_Test_Crop_NL, type="class")

par(bg = "#fbeed1")
plot(SVM_fit_L, data=DF_Train_Crop, temperature~ph, slice=list(humidity=42,N=70,P=70,K=100,rainfall=150
```



```
plot(SVM_fit_L, data=DF_Train_Crop, humidity~ph, slice=list(temperature=20,N=70,P=70,K=100,rainfall=150)
```

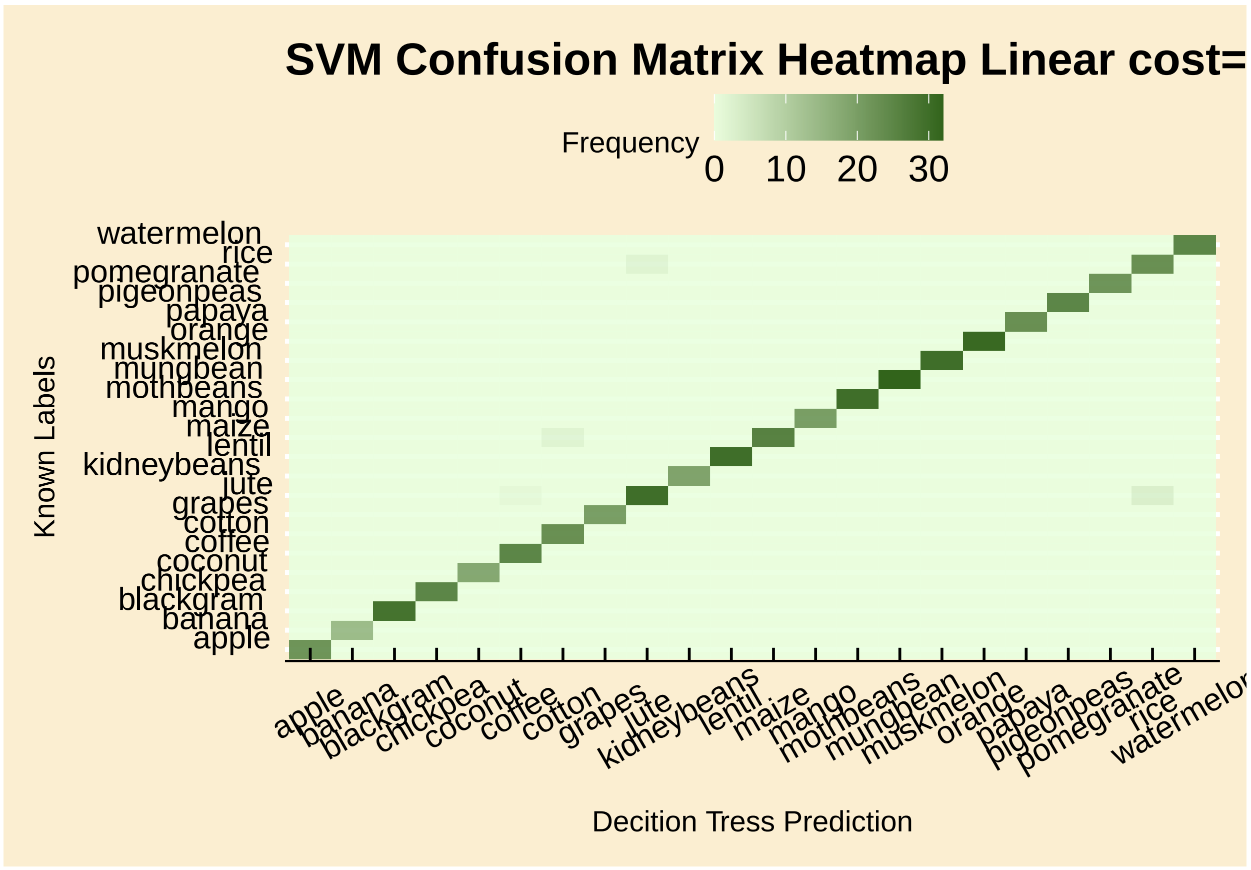


## Confusion Matrix

```
(Ltable <- table(pred_L, DF_Test_Crop_Labels))
```

```
##                DF_Test_Crop_Labels
## pred_L          apple banana blackgram chickpea coconut coffee cotton grapes jute
##   apple            22      0         0        0       0      0      0      0    0
##   banana            0     14         0        0       0      0      0      0    0
##   blackgram         0      0        29        0       0      0      0      0    0
##   chickpea          0      0         0       25       0      0      0      0    0
##   coconut           0      0         0        0      18      0      0      0    0
##   coffee            0      0         0        0       0     25      0      0    1
##   cotton            0      0         0        0       0      0     23      0    0
##   grapes            0      0         0        0       0      0      0     20    0
##   jute              0      0         0        0       0      0      0      0   30
##   kidneybeans       0      0         0        0       0      0      0      0    0
##   lentil            0      0         0        0       0      0      0      0    0
##   maize             0      0         0        0       0      0      0      0    0
##   mango             0      0         0        0       0      0      0      0    0
##   mothbeans         0      0         0        0       0      0      0      0    0
##   mungbean          0      0         0        0       0      0      0      0    0
##   muskmelon         0      0         0        0       0      0      0      0    0
##   orange            0      0         0        0       0      0      0      0    0
##   papaya            0      0         0        0       0      0      0      0    0
##   pigeonpeas        0      0         0        0       0      0      0      0    0
##   pomegranate       0      0         0        0       0      0      0      0    0
##   rice              0      0         0        0       0      0      0      0    3
##   watermelon      0      0         0        0       0      0      0      0    0
##              DF_Test_Crop_Labels
## pred_L        kidneybeans lentil maize mango mothbeans mungbean muskmelon
##   apple                 0      0     0     0         0        0         0
##   banana                0      0     0     0         0        0         0
##   blackgram             0      0     0     0         0        0         0
##   chickpea              0      0     0     0         0        0         0
##   coconut               0      0     0     0         0        0         0
##   coffee                0      0     0     0         0        0         0
##   cotton                0      0     2     0         0        0         0
##   grapes                0      0     0     0         0        0         0
##   jute                  0      0     0     0         0        0         0
##   kidneybeans          19      0     0     0         0        0         0
##   lentil                0     30     0     0         0        0         0
##   maize                 0      0    26     0         0        0         0
##   mango                 0      0     0    20         0        0         0
##   mothbeans             0      0     0     0        30        0         0
##   mungbean              0      0     0     0         0       32         0
##   muskmelon             0      0     0     0         0        0        30
##   orange                0      0     0     0         0        0         0
##   papaya                0      0     0     0         0        0         0
##   pigeonpeas            0      0     0     0         0        0         0
##   pomegranate           0      0     0     0         0        0         0
##   rice                  0      0     0     0         0        0         0
##   watermelon            0      0     0     0         0        0         0
##              DF_Test_Crop_Labels
## pred_L        orange papaya pigeonpeas pomegranate rice watermelon
##   apple            0      0          0           0    0          0
##   banana           0      0          0           0    0          0
##   blackgram        0      0          0           0    0          0
##   chickpea         0      0          0           0    0          0
##   coconut          0      0          0           0    0          0
##   coffee           0      0          0           0    0          0
##   cotton           0      0          0           0    0          0
##   grapes           0      0          0           0    0          0
##   jute             0      0          0           0    2          0
##   kidneybeans      0      0          0           0    0          0
##   lentil           0      0          0           0    0          0
##   maize            0      0          0           0    0          0
##   mango            0      0          0           0    0          0
##   mothbeans        0      0          0           0    0          0
##   mungbean         0      0          0           0    0          0
##   muskmelon        0      0          0           0    0          0
##   orange          31      0          0           0    0          0
##   papaya           0     23          0           0    0          0
##   pigeonpeas       0      0         25           0    0          0
##   pomegranate      0      0          0          22    0          0
##   rice             0      0          0           0   23          0
##   watermelon       0      0          0           0    0         25
```

```
get_heatmap("SVM Confusion Matrix Heatmap Linear cost=0.1", pred_L)
```



### Misclassification Rate for Linear

```
(MR_L <- 1 - sum(diag(Ltable))/sum(Ltable))
```

```
## [1] 0.01454545
```

## Radial Kernel

```
SVM_fit_R <- svm(label~., data=DF_Train_Crop,
                 kernel="radial", cost=.1,
                 scale=FALSE)
print(SVM_fit_R)
```

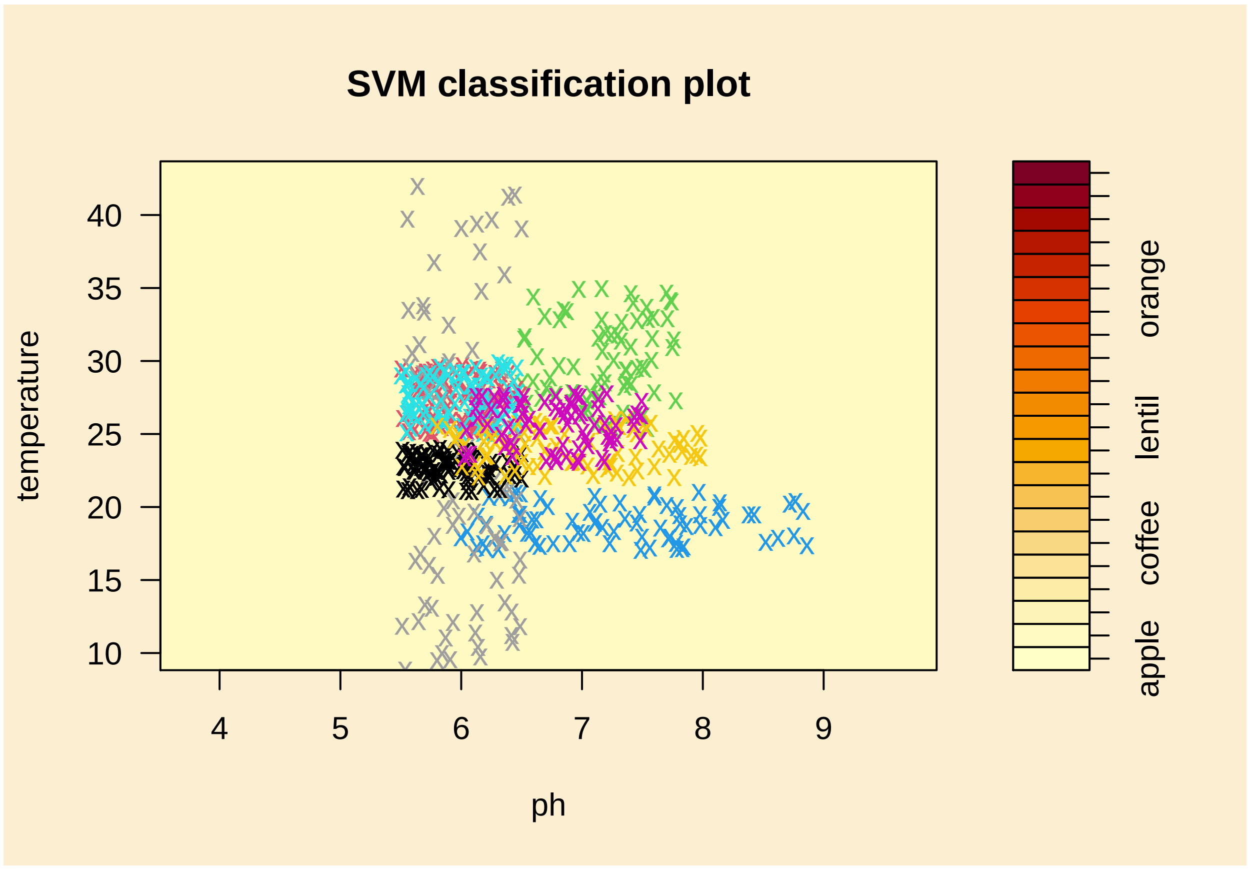
```
##
## Call:
## svm(formula = label ~ ., data = DF_Train_Crop, kernel = "radial",
##     cost = 0.1, scale = FALSE)
##
##
## Parameters:
```

```
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  0.1
##
## Number of Support Vectors:  1650
```

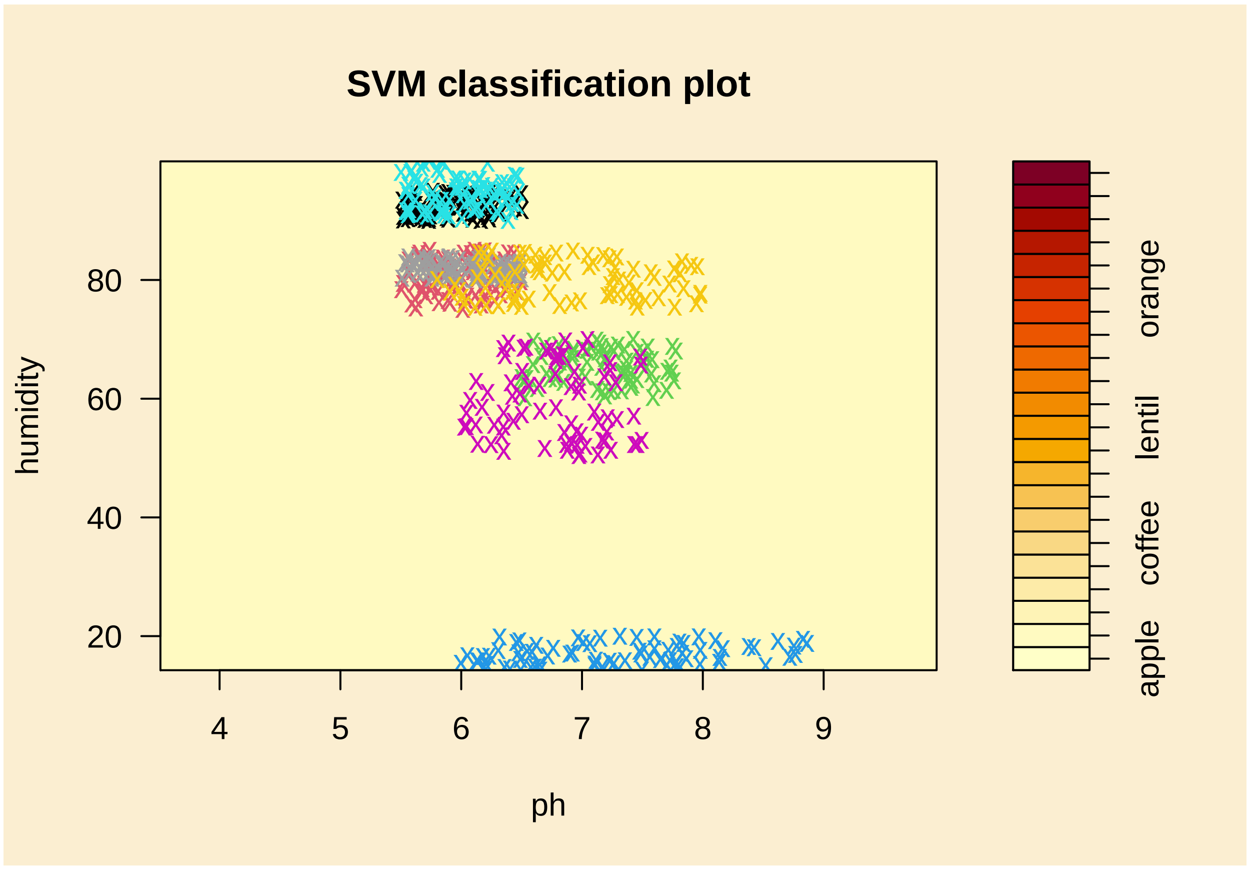
## Prediction

```
pred_R <- predict(SVM_fit_R, DF_Test_Crop_NL, type="class")

par(bg = "#fbeed1")
plot(SVM_fit_R, data=DF_Train_Crop, temperature~ph, slice=list(humidity=42,N=70,P=70,K=100,rainfall=150)
```



```
plot(SVM_fit_R, data=DF_Train_Crop, humidity~ph, slice=list(temperature=20,N=70,P=70,K=100,rainfall=150)
```

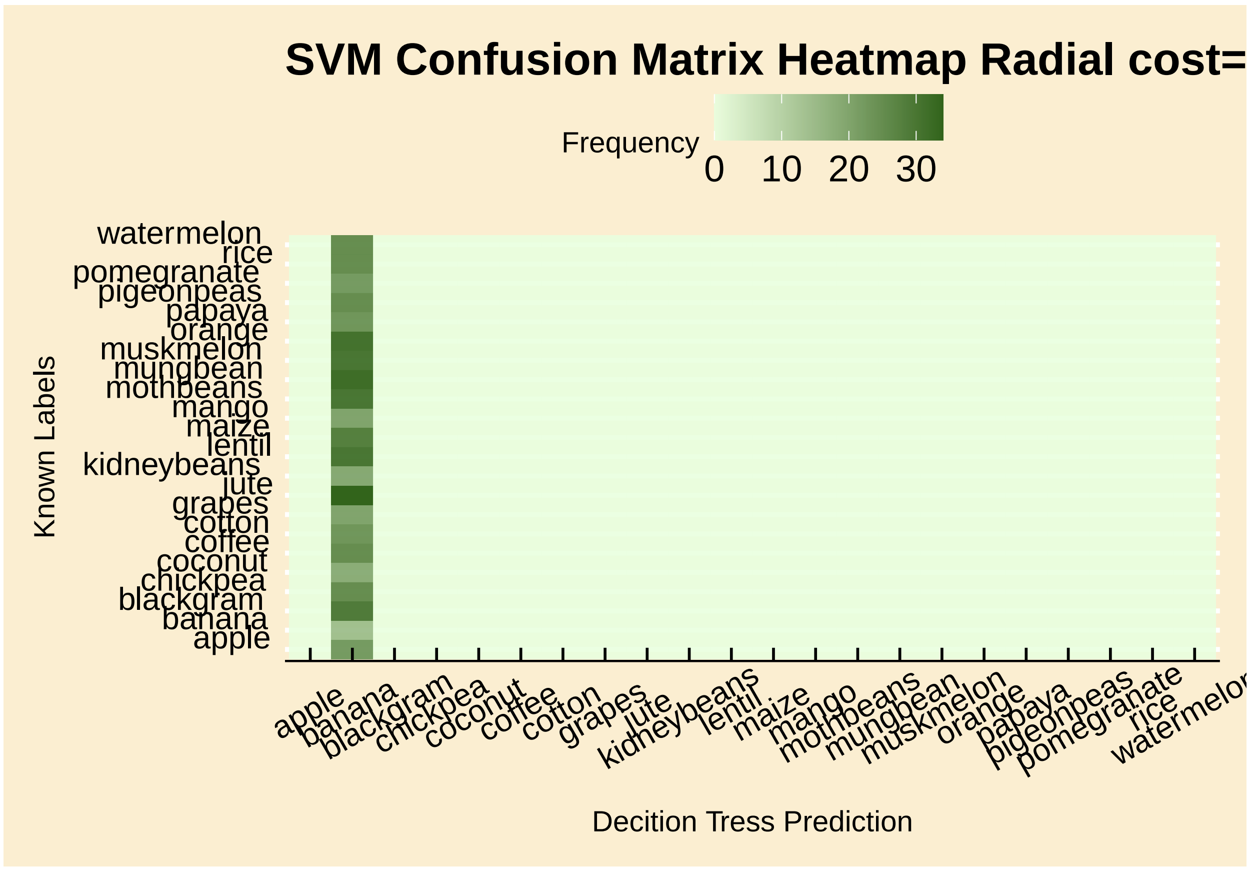


## Confusion Matrix

```
(Rtable <- table(pred_R, DF_Test_Crop_Labels))
```

```
##                DF_Test_Crop_Labels
## pred_R          apple banana blackgram chickpea coconut coffee cotton grapes jute
##   apple             0      0         0        0       0      0      0      0    0
##   banana           22     14        29       25      18     25     23     20   34
##   blackgram         0      0         0        0       0      0      0      0    0
##   chickpea          0      0         0        0       0      0      0      0    0
##   coconut           0      0         0        0       0      0      0      0    0
##   coffee            0      0         0        0       0      0      0      0    0
##   cotton            0      0         0        0       0      0      0      0    0
##   grapes            0      0         0        0       0      0      0      0    0
##   jute              0      0         0        0       0      0      0      0    0
##   kidneybeans       0      0         0        0       0      0      0      0    0
##   lentil            0      0         0        0       0      0      0      0    0
##   maize             0      0         0        0       0      0      0      0    0
##   mango             0      0         0        0       0      0      0      0    0
##   mothbeans         0      0         0        0       0      0      0      0    0
##   mungbean          0      0         0        0       0      0      0      0    0
##   muskmelon         0      0         0        0       0      0      0      0    0
##   orange            0      0         0        0       0      0      0      0    0
##   papaya            0      0         0        0       0      0      0      0    0
##   pigeonpeas        0      0         0        0       0      0      0      0    0
##   pomegranate       0      0         0        0       0      0      0      0    0
##   rice              0      0         0        0       0      0      0      0    0
##   watermelon      0      0         0        0       0      0      0      0    0
##              DF_Test_Crop_Labels
## pred_R        kidneybeans lentil maize mango mothbeans mungbean muskmelon
##   apple                 0      0     0     0         0        0         0
##   banana               19     30    28    20        30       32        30
##   blackgram             0      0     0     0         0        0         0
##   chickpea              0      0     0     0         0        0         0
##   coconut               0      0     0     0         0        0         0
##   coffee                0      0     0     0         0        0         0
##   cotton                0      0     0     0         0        0         0
##   grapes                0      0     0     0         0        0         0
##   jute                  0      0     0     0         0        0         0
##   kidneybeans           0      0     0     0         0        0         0
##   lentil                0      0     0     0         0        0         0
##   maize                 0      0     0     0         0        0         0
##   mango                 0      0     0     0         0        0         0
##   mothbeans             0      0     0     0         0        0         0
##   mungbean              0      0     0     0         0        0         0
##   muskmelon             0      0     0     0         0        0         0
##   orange                0      0     0     0         0        0         0
##   papaya                0      0     0     0         0        0         0
##   pigeonpeas            0      0     0     0         0        0         0
##   pomegranate           0      0     0     0         0        0         0
##   rice                  0      0     0     0         0        0         0
##   watermelon            0      0     0     0         0        0         0
##              DF_Test_Crop_Labels
## pred_R        orange papaya pigeonpeas pomegranate rice watermelon
##   apple            0      0          0           0    0          0
##   banana          31     23         25          22   25         25
##   blackgram        0      0          0           0    0          0
##   chickpea         0      0          0           0    0          0
##   coconut          0      0          0           0    0          0
##   coffee           0      0          0           0    0          0
##   cotton           0      0          0           0    0          0
##   grapes           0      0          0           0    0          0
##   jute             0      0          0           0    0          0
##   kidneybeans      0      0          0           0    0          0
##   lentil           0      0          0           0    0          0
##   maize            0      0          0           0    0          0
##   mango            0      0          0           0    0          0
##   mothbeans        0      0          0           0    0          0
##   mungbean         0      0          0           0    0          0
##   muskmelon        0      0          0           0    0          0
##   orange           0      0          0           0    0          0
##   papaya           0      0          0           0    0          0
##   pigeonpeas       0      0          0           0    0          0
##   pomegranate      0      0          0           0    0          0
##   rice             0      0          0           0    0          0
##   watermelon       0      0          0           0    0          0
```

```r
get_heatmap("SVM Confusion Matrix Heatmap Radial cost=0.1", pred_R)
```



## Misclassification Rate for Radial

```
(MR_R <- 1 - sum(diag(Rtable))/sum(Rtable))
```

```
## [1] 0.9745455
```