2. KAGGLE Problem- Otto Group Product Classification Challenge

Classify products into the correct category

The Otto Group is one of the world's biggest e-commerce companies, with subsidiaries in more than 20 countries, including Crate & Barrel (USA), Otto.de (Germany) and 3 Suisses (France).
They are selling millions of products worldwide every day, with several thousand products being added to our product line.

A consistent analysis of the performance of their products is crucial. However, due to our diverse global infrastructure, many identical products get classified differently. Therefore, the quality of our product analysis depends heavily on the ability to accurately cluster similar products. The better the classification, the more insights we can generate about our product range.

For this competition, a dataset is provided with 93 features for more than 200,000 products. The objective is to build a predictive model which is able to distinguish between the main product categories.

File descriptions

trainData.csv - the training set
testData.csv - the test set
sampleSubmission.csv - a sample submission file in the correct format

Data fields

- id - an anonymous id unique to a product.
- feat_1, feat_2, ..., feat_93 - the various features of a product.
- target - the class of a product.

Data Set Description

Each row corresponds to a single product. There are a total of 93 numerical features, which represent counts of different events. All features have been obfuscated and will not be defined any further.

There are nine categories for all products. Each target category represents one of our most important product categories (like fashion, electronics, etc.). The products for the training and testing sets are selected randomly.

IMPLEMENTATION:

1. RANDOM FOREST GENERATION

```
install.packages('randomForest')                    # install randomForest package.
library(randomForest)                               # include random forest library.
  set.seed(12)                  # set a unique seed number so as to get the same results everytime we run
                                                                                the below model.
fit <- randomForest(as.factor(target) ~ ., data=train2, importance=TRUE, ntree=100)
 # create a random forest model using the target field as the response and all 93 features as inputs
varImpPlot(fit)    # create a dotchart of variable/feature importance as measured by a Random Forest
pred <- predict(fit,test,type="prob")                    # use the random forest model to create a prediction
submit <- data.frame(id = test$id, pred)
write.csv(submit, file = "random_forest_submit.csv", row.names = FALSE)
```

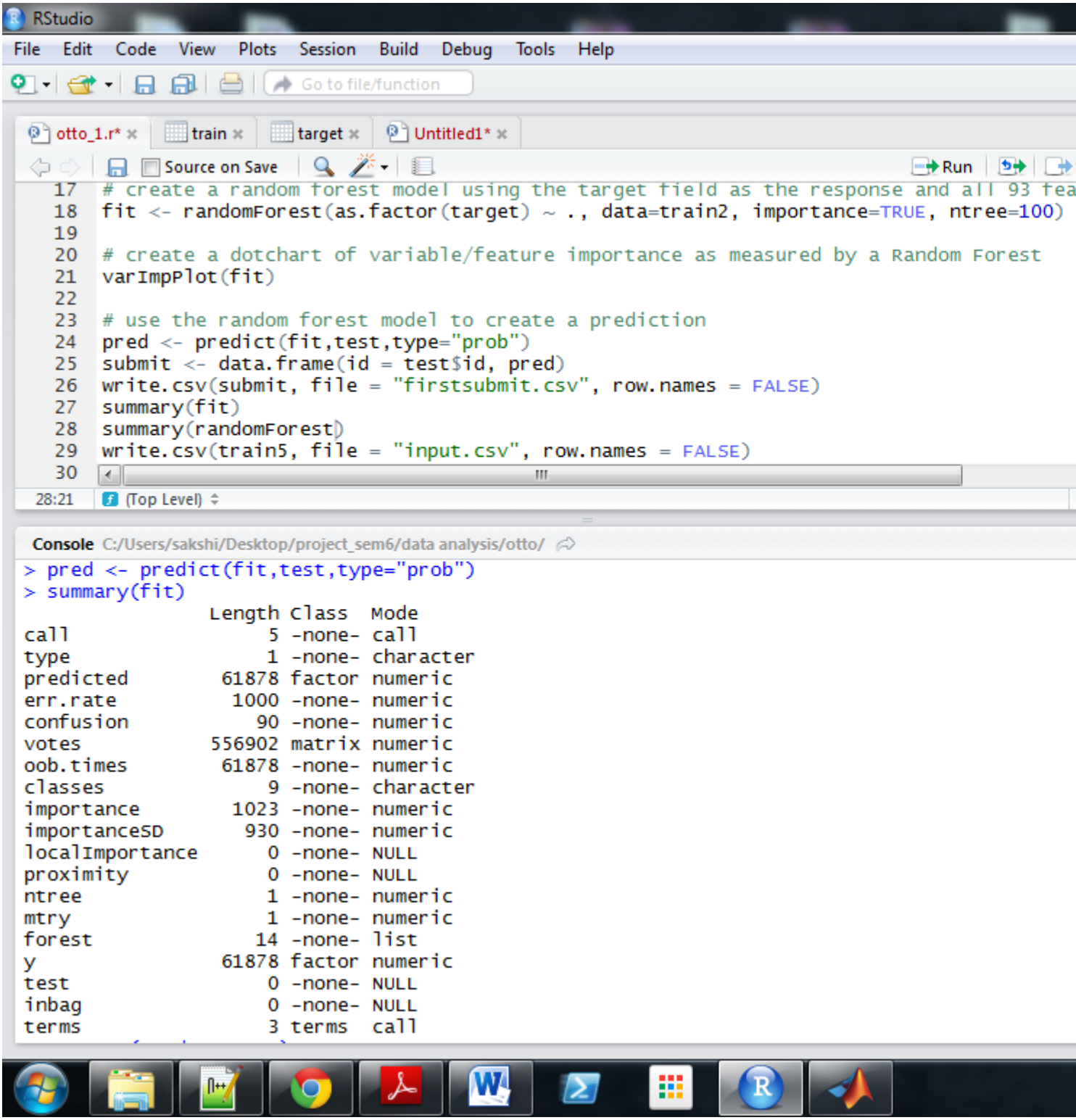
```
summary(fit)
               Length Class  Mode
call                5 -none- call
type                1 -none- character
predicted       61878 factor numeric
err.rate         1000 -none- numeric
confusion          90 -none- numeric
votes          556902 matrix numeric
oob.times       61878 -none- numeric
classes             9 -none- character
importance       1023 -none- numeric
importanceSD      930 -none- numeric
localImportance     0 -none- NULL
proximity           0 -none- NULL
ntree               1 -none- numeric
mtry                1 -none- numeric
forest             14 -none- list
y               61878 factor numeric
test                0 -none- NULL
inbag               0 -none- NULL
terms               3 terms  call
```

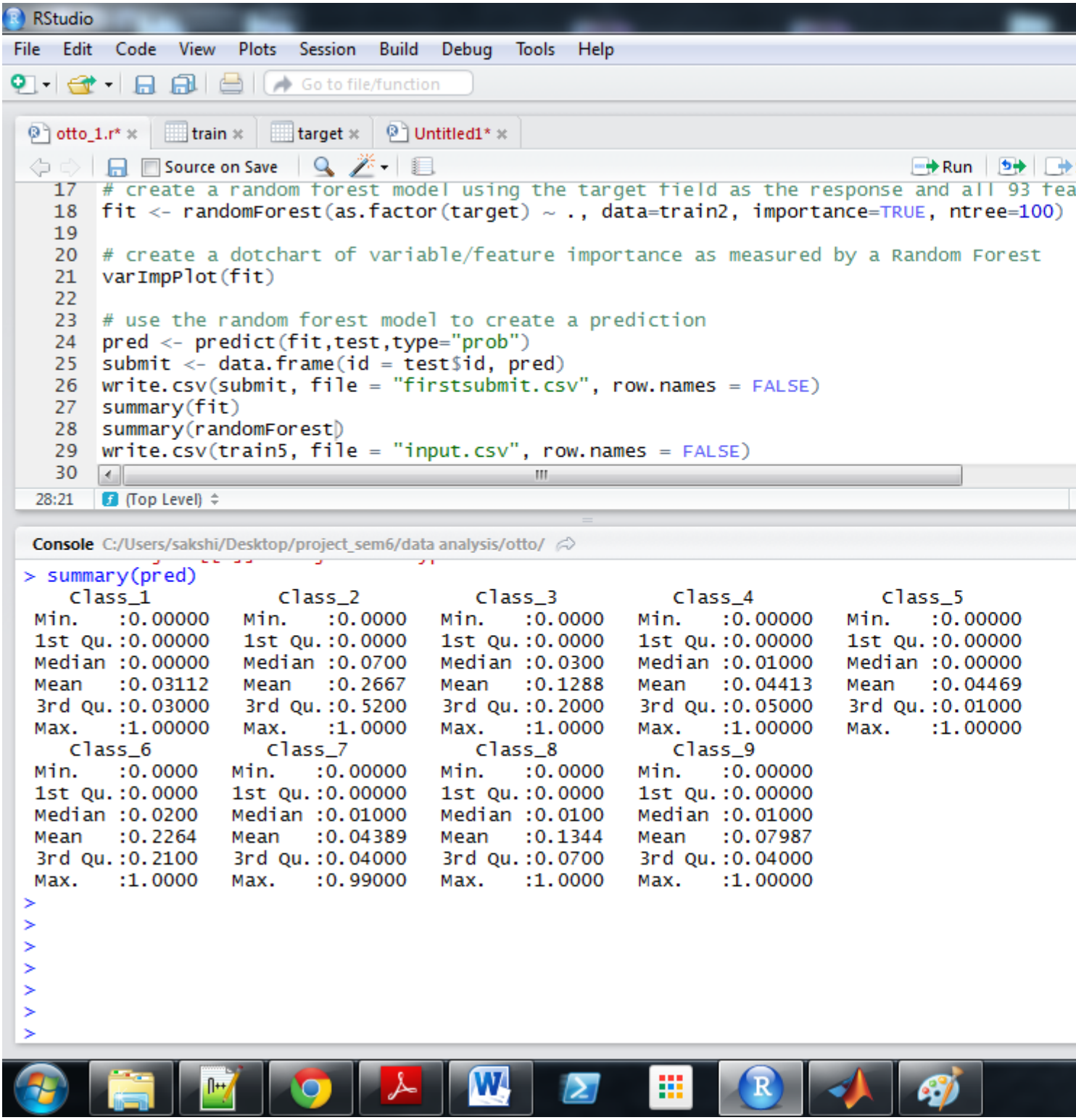
Hence, we have generated random forest using length class mode of 1 none character type.

```
summary(pred)
   Class_1          Class_2          Class_3          Class_4          Class_5
 Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
 1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
 Median :0.00000   Median :0.0700   Median :0.0300   Median :0.01000   Median :0.00000
```

Mean :0.03112 Mean :0.2667 Mean :0.1288 Mean :0.04413 Mean :0.04469
3rd Qu.:0.03000 3rd Qu.:0.5200 3rd Qu.:0.2000 3rd Qu.:0.05000 3rd Qu.:0.01000
Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.00000 Max. :1.00000
Class_6 Class_7 Class_8 Class_9
Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.00000
1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.00000
Median :0.0200 Median :0.01000 Median :0.0100 Median :0.01000
Mean :0.2264 Mean :0.04389 Mean :0.1344 Mean :0.07987
3rd Qu.:0.2100 3rd Qu.:0.04000 3rd Qu.:0.0700 3rd Qu.:0.04000
Max. :1.0000 Max. :0.99000 Max. :1.0000 Max. :1.00000

Mean and Median of all quartiles in each class is derived for the prediction.


RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
otto_1.r* train target Untitled1*
Source on Save
Run
17 # create a random forest model using the target field as the response and all 93 fea
18 fit <- randomForest(as.factor(target) ~ ., data=train2, importance=TRUE, ntree=100)
19
20 # create a dotchart of variable/feature importance as measured by a Random Forest
21 varImpPlot(fit)
22
23 # use the random forest model to create a prediction
24 pred <- predict(fit,test,type="prob")
25 submit <- data.frame(id = test$id, pred)
26 write.csv(submit, file = "firstsubmit.csv", row.names = FALSE)
27 summary(fit)
28 summary(randomForest)
29 write.csv(train5, file = "input.csv", row.names = FALSE)
30
28:21 (Top Level)
Console C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/
> pred <- predict(fit,test,type="prob")
> summary(fit)
Length Class Mode
call 5 -none- call
type 1 -none- character
predicted 61878 factor numeric
err.rate 1000 -none- numeric
confusion 90 -none- numeric
votes 556902 matrix numeric
oob.times 61878 -none- numeric
classes 9 -none- character
importance 1023 -none- numeric
importanceSD 930 -none- numeric
localImportance 0 -none- NULL
proximity 0 -none- NULL
ntree 1 -none- numeric
mtry 1 -none- numeric
forest 14 -none- list
y 61878 factor numeric
test 0 -none- NULL
inbag 0 -none- NULL
terms 3 terms call

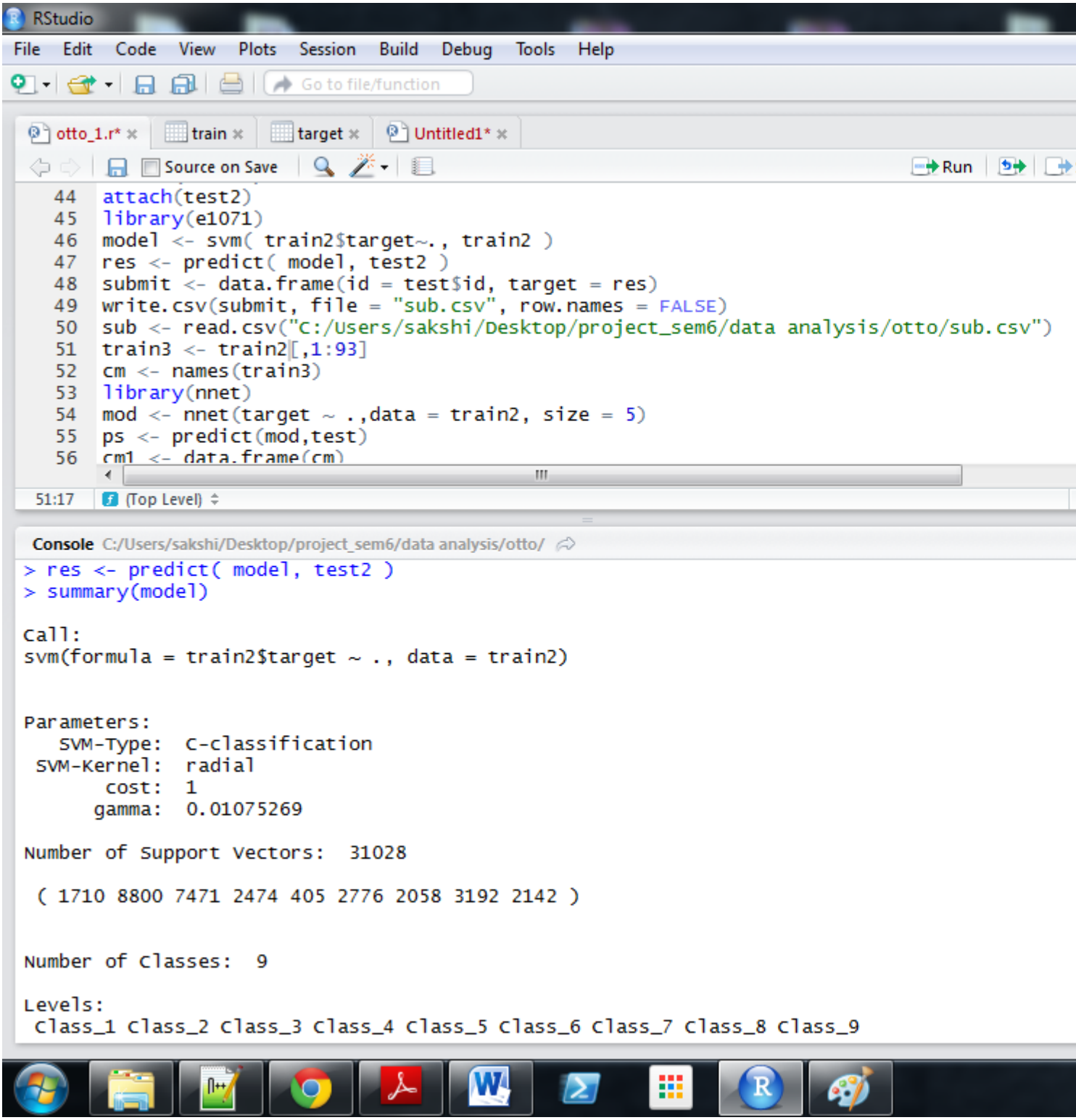

RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
otto_1.r* train target Untitled1*
Source on Save
Run
17 # create a random forest model using the target field as the response and all 93 fea
18 fit <- randomForest(as.factor(target) ~ ., data=train2, importance=TRUE, ntree=100)
19
20 # create a dotchart of variable/feature importance as measured by a Random Forest
21 varImpPlot(fit)
22
23 # use the random forest model to create a prediction
24 pred <- predict(fit,test,type="prob")
25 submit <- data.frame(id = test$id, pred)
26 write.csv(submit, file = "firstsubmit.csv", row.names = FALSE)
27 summary(fit)
28 summary(randomForest)
29 write.csv(train5, file = "input.csv", row.names = FALSE)
30
28:21 (Top Level)
Console C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/
> summary(pred)
Class_1 Class_2 Class_3 Class_4 Class_5
Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.00000 Min. :0.00000
1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.00000
Median :0.00000 Median :0.0700 Median :0.0300 Median :0.01000 Median :0.00000
Mean :0.03112 Mean :0.2667 Mean :0.1288 Mean :0.04413 Mean :0.04469
3rd Qu.:0.03000 3rd Qu.:0.5200 3rd Qu.:0.2000 3rd Qu.:0.05000 3rd Qu.:0.01000
Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.00000 Max. :1.00000
Class_6 Class_7 Class_8 Class_9
Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.00000
1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.00000
Median :0.0200 Median :0.01000 Median :0.0100 Median :0.01000
Mean :0.2264 Mean :0.04389 Mean :0.1344 Mean :0.07987
3rd Qu.:0.2100 3rd Qu.:0.04000 3rd Qu.:0.0700 3rd Qu.:0.04000
Max. :1.0000 Max. :0.99000 Max. :1.0000 Max. :1.00000
>

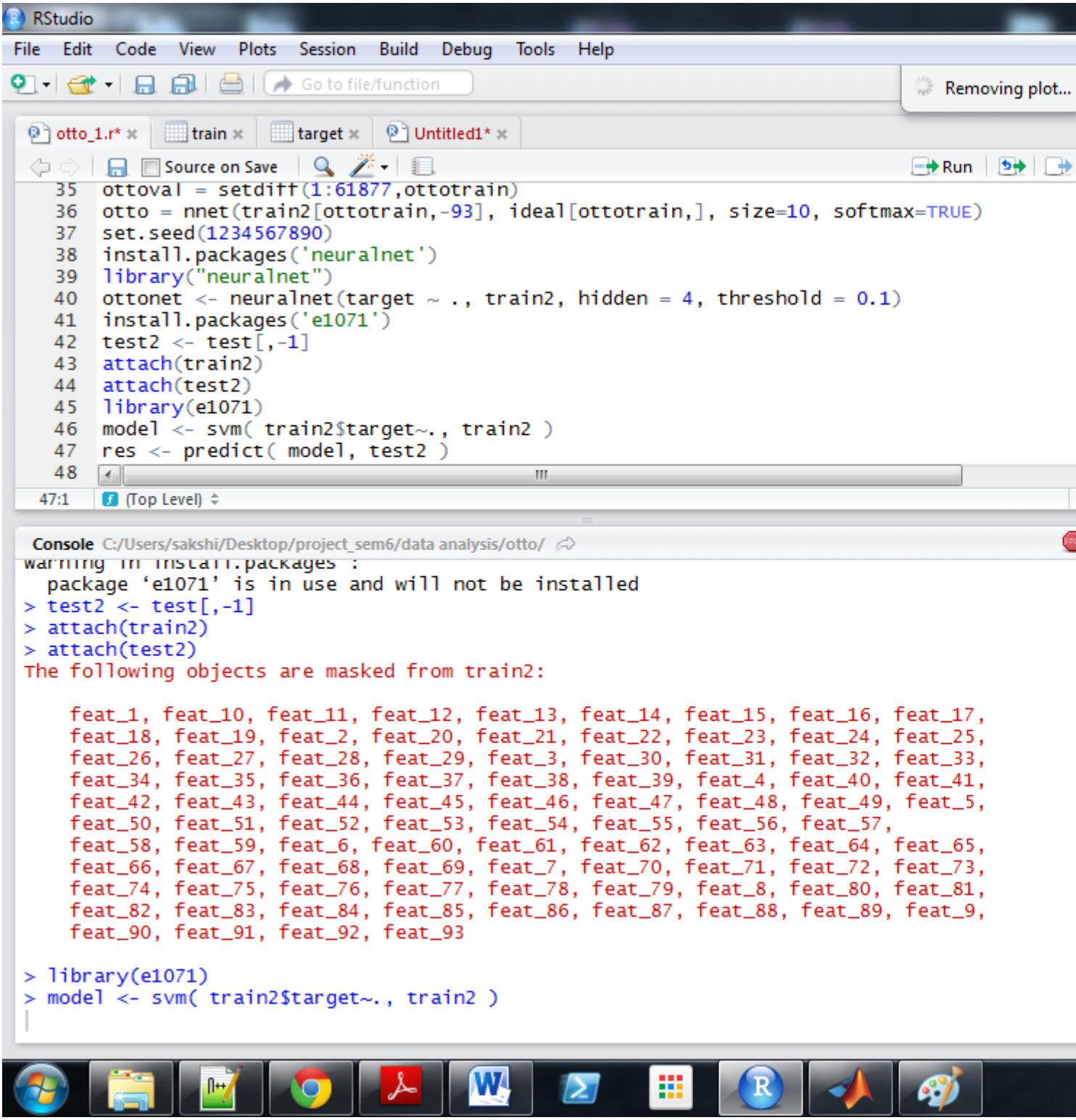
2. SUPPORT VECTOR MACHINE (SVM)

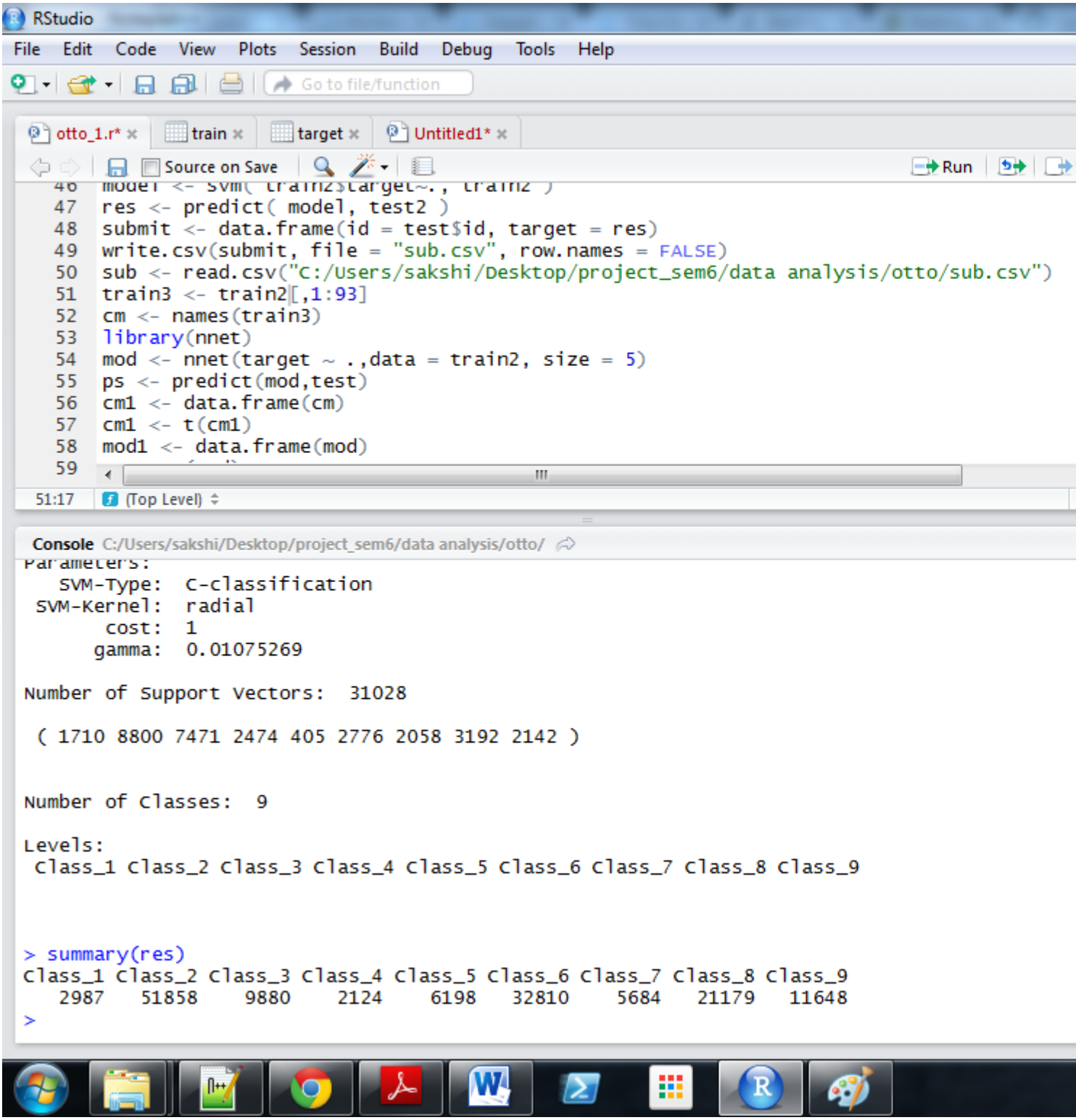
```
install.packages('e1071')                    # install SVM package
test2 <- test[,-1]                           # removing the id column
attach(train2)
attach(test2)
library(e1071)                               #include SVM library
model <- svm( train2$target~., train2 )      # train the data model with class named 'target'
res <- predict( model, test2 )               # predict the classification for test data
   submit <- data.frame(id = test$id, target = res)          # create a data frame by re combining the result
                                                                with id column
write.csv(submit, file = "sub.csv", row.names = FALSE)
sub <- read.csv("C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/sub.csv")

 summary(model)

Call:
svm(formula = train2$target ~ ., data = train2)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.01075269

Number of Support Vectors:  31028

 ( 1710 8800 7471 2474 405 2776 2058 3192 2142 )


Number of Classes:  9

Levels:
 Class_1 Class_2 Class_3 Class_4 Class_5 Class_6 Class_7 Class_8 Class_9
```

Hence, we have used C-classification type of svm with radial kernel for the classification and 31028 support vectors are derived.

```
 summary(res)
Class_1 Class_2 Class_3 Class_4 Class_5 Class_6 Class_7 Class_8 Class_9
   2987   51858    9880    2124    6198   32810    5684   21179   11648
```


RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
otto_1.r* train target Untitled1*
Source on Save
Run
attach(test2)
library(e1071)
model <- svm(train2$target~., train2)
res <- predict(model, test2)
submit <- data.frame(id = test$id, target = res)
write.csv(submit, file = "sub.csv", row.names = FALSE)
sub <- read.csv("C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/sub.csv")
train3 <- train2[,1:93]
cm <- names(train3)
library(nnet)
mod <- nnet(target ~ .,data = train2, size = 5)
ps <- predict(mod,test)
cm1 <- data.frame(cm)
51:17 (Top Level)
Console C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/
> res <- predict(model, test2)
> summary(model)
Call:
svm(formula = train2$target ~ ., data = train2)
Parameters:
SVM-Type: C-classification
SVM-Kernel: radial
cost: 1
gamma: 0.01075269
Number of Support Vectors: 31028
(1710 8800 7471 2474 405 2776 2058 3192 2142)
Number of Classes: 9
Levels:
Class_1 Class_2 Class_3 Class_4 Class_5 Class_6 Class_7 Class_8 Class_9


RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Removing plot...
otto_1.r* train target Untitled1*
Source on Save
Run
35 ottoval = setdiff(1:61877,ottotrain)
36 otto = nnet(train2[ottotrain,-93], ideal[ottotrain,], size=10, softmax=TRUE)
37 set.seed(1234567890)
38 install.packages('neuralnet')
39 library("neuralnet")
40 ottonet <- neuralnet(target ~ ., train2, hidden = 4, threshold = 0.1)
41 install.packages('e1071')
42 test2 <- test[,-1]
43 attach(train2)
44 attach(test2)
45 library(e1071)
46 model <- svm(train2$target~., train2)
47 res <- predict(model, test2)
48
47:1 (Top Level)
Console C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/
package 'e1071' is in use and will not be installed
> test2 <- test[,-1]
> attach(train2)
> attach(test2)
The following objects are masked from train2:
feat_1, feat_10, feat_11, feat_12, feat_13, feat_14, feat_15, feat_16, feat_17,
feat_18, feat_19, feat_2, feat_20, feat_21, feat_22, feat_23, feat_24, feat_25,
feat_26, feat_27, feat_28, feat_29, feat_3, feat_30, feat_31, feat_32, feat_33,
feat_34, feat_35, feat_36, feat_37, feat_38, feat_39, feat_4, feat_40, feat_41,
feat_42, feat_43, feat_44, feat_45, feat_46, feat_47, feat_48, feat_49, feat_5,
feat_50, feat_51, feat_52, feat_53, feat_54, feat_55, feat_56, feat_57,
feat_58, feat_59, feat_6, feat_60, feat_61, feat_62, feat_63, feat_64, feat_65,
feat_66, feat_67, feat_68, feat_69, feat_7, feat_70, feat_71, feat_72, feat_73,
feat_74, feat_75, feat_76, feat_77, feat_78, feat_79, feat_8, feat_80, feat_81,
feat_82, feat_83, feat_84, feat_85, feat_86, feat_87, feat_88, feat_89, feat_9,
feat_90, feat_91, feat_92, feat_93
> library(e1071)
> model <- svm(train2$target~., train2)


RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
otto_1.r* train target Untitled1*
Source on Save
Run
model <- svm(train2$target~., train2)
res <- predict(model, test2)
submit <- data.frame(id = test$id, target = res)
write.csv(submit, file = "sub.csv", row.names = FALSE)
sub <- read.csv("C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/sub.csv")
train3 <- train2[,1:93]
cm <- names(train3)
library(nnet)
mod <- nnet(target ~ .,data = train2, size = 5)
ps <- predict(mod,test)
cm1 <- data.frame(cm)
cm1 <- t(cm1)
mod1 <- data.frame(mod)
51:17 (Top Level)
Console C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/
Parameters:
SVM-Type: C-classification
SVM-Kernel: radial
cost: 1
gamma: 0.01075269
Number of Support Vectors: 31028
(1710 8800 7471 2474 405 2776 2058 3192 2142)
Number of Classes: 9
Levels:
Class_1 Class_2 Class_3 Class_4 Class_5 Class_6 Class_7 Class_8 Class_9
> summary(res)
Class_1 Class_2 Class_3 Class_4 Class_5 Class_6 Class_7 Class_8 Class_9
2987 51858 9880 2124 6198 32810 5684 21179 11648
>

3. NEURAL NETWORKS

```
train3 <- train2[,1:93]                          # include all columns except the class label column
cm <- names(train3)                              # get the names of all the columns
library(nnet)                                    # include the neural network library  ie. nnet
   mod <- nnet(target ~ .,data = train2, size = 5)   # generate a neural network model with number
                                                     of hidden layers as 5 and train the data.
ps <- predict(mod,test)                          # predict the test data with the model generated.
cm1 <- data.frame(cm)
cm1 <- t(cm1)                                    # take the transpose of the result
mod1 <- data.frame(mod)
summary(mod)
write.csv(ps1, file = "subneuralfin.csv", row.names = FALSE)
```

```
mod <- nnet(target ~ .,data = train2, size = 5)
# weights:  524
initial  value 140521.042097
iter  10 value 64189.244074
iter  20 value 51283.502796
iter  30 value 49777.149148
iter  40 value 48315.724506
iter  50 value 46965.225870
iter  60 value 45911.909257
iter  70 value 45208.936264
iter  80 value 44719.998361
iter  90 value 44343.831812
iter 100 value 43675.211675
final  value 43675.211675
stopped after 100 iterations
```

Hence, the training of neural network goes till 100 iterations in which our input matrix consist of 93x61487 dimension and target matrix of 9x61487 with 524 weights using softmax modelling.

```
a 93-5-9 network with 524 weights
options were - softmax modelling
  b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9->h1 i10->h1
  -4.42    0.40   -0.25    0.11    0.97    1.19   -0.31    2.82   -0.22    3.56    0.05
i11->h1 i12->h1 i13->h1 i14->h1 i15->h1 i16->h1 i17->h1 i18->h1 i19->h1 i20->h1 i21->h1
  -6.06   -2.82    3.56    1.57    3.64   -0.75   -5.78    1.22    0.87   -0.12    1.31
i22->h1 i23->h1 i24->h1 i25->h1 i26->h1 i27->h1 i28->h1 i29->h1 i30->h1 i31->h1 i32->h1
   0.41   -2.76   -0.73    2.98   -2.74   -2.85    0.87   -1.03    0.53   -1.20   -2.45
i33->h1 i34->h1 i35->h1 i36->h1 i37->h1 i38->h1 i39->h1 i40->h1 i41->h1 i42->h1 i43->h1
   4.93    3.25    4.67   -1.97    0.94    1.89   -5.09    3.28    2.47   -2.36   -5.66
i44->h1 i45->h1 i46->h1 i47->h1 i48->h1 i49->h1 i50->h1 i51->h1 i52->h1 i53->h1 i54->h1
  -2.30   -1.24    0.22    1.41    1.00   -2.58   -2.81    0.70   -0.96   -6.95    0.84
```

i55->h1 i56->h1 i57->h1 i58->h1 i59->h1 i60->h1 i61->h1 i62->h1 i63->h1 i64->h1 i65->h1
-3.20 -6.63 -3.73 3.68 -5.94 -6.27 -3.37 -0.37 1.35 -0.19 0.85
i66->h1 i67->h1 i68->h1 i69->h1 i70->h1 i71->h1 i72->h1 i73->h1 i74->h1 i75->h1 i76->h1
-2.39 -0.09 2.93 3.24 -0.53 1.83 2.31 0.57 -0.09 4.11 0.44
i77->h1 i78->h1 i79->h1 i80->h1 i81->h1 i82->h1 i83->h1 i84->h1 i85->h1 i86->h1 i87->h1
-2.63 1.24 -4.13 2.13 -3.29 -3.92 0.61 0.57 -5.16 -1.14 -2.12
i88->h1 i89->h1 i90->h1 i91->h1 i92->h1 i93->h1
-2.26 3.05 6.43 -1.21 0.54 -1.51
b->h2 i1->h2 i2->h2 i3->h2 i4->h2 i5->h2 i6->h2 i7->h2 i8->h2 i9->h2 i10->h2
-12.77 -5.37 4.00 0.67 0.19 -8.69 -5.41 -4.40 -9.55 2.18 -2.43
i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2 i18->h2 i19->h2 i20->h2 i21->h2
8.45 -1.88 1.48 -6.23 -1.36 0.37 0.93 1.34 5.39 1.94 1.43
i22->h2 i23->h2 i24->h2 i25->h2 i26->h2 i27->h2 i28->h2 i29->h2 i30->h2 i31->h2 i32->h2
1.27 3.16 -3.18 -3.42 2.44 -1.12 2.28 2.21 -12.67 0.34 2.22
i33->h2 i34->h2 i35->h2 i36->h2 i37->h2 i38->h2 i39->h2 i40->h2 i41->h2 i42->h2 i43->h2
-3.21 -1.51 3.65 -1.28 -4.22 2.10 10.98 -14.01 -0.31 8.54 -2.51
i44->h2 i45->h2 i46->h2 i47->h2 i48->h2 i49->h2 i50->h2 i51->h2 i52->h2 i53->h2 i54->h2
-0.02 3.21 0.32 10.31 -2.55 3.25 6.46 -0.42 -0.89 -3.34 0.68
i55->h2 i56->h2 i57->h2 i58->h2 i59->h2 i60->h2 i61->h2 i62->h2 i63->h2 i64->h2 i65->h2
1.50 -3.75 6.48 4.49 -0.17 12.85 -0.49 -6.17 2.17 -7.33 0.32
i66->h2 i67->h2 i68->h2 i69->h2 i70->h2 i71->h2 i72->h2 i73->h2 i74->h2 i75->h2 i76->h2
-3.34 -0.86 -10.30 3.70 0.11 1.40 -13.06 4.08 6.08 3.77 -2.37
i77->h2 i78->h2 i79->h2 i80->h2 i81->h2 i82->h2 i83->h2 i84->h2 i85->h2 i86->h2 i87->h2
-8.46 -6.34 -4.50 2.06 2.40 1.20 4.01 -12.70 -7.90 0.84 2.14
i88->h2 i89->h2 i90->h2 i91->h2 i92->h2 i93->h2
-4.50 -1.89 2.82 -0.73 -0.84 5.02
b->h3 i1->h3 i2->h3 i3->h3 i4->h3 i5->h3 i6->h3 i7->h3 i8->h3 i9->h3 i10->h3
0.15 0.44 -3.23 2.28 -0.09 -0.73 -3.09 2.68 0.43 -2.66 -1.04
i11->h3 i12->h3 i13->h3 i14->h3 i15->h3 i16->h3 i17->h3 i18->h3 i19->h3 i20->h3 i21->h3
2.66 -0.03 -0.50 -2.89 -3.34 -0.07 2.84 0.64 0.31 0.78 2.43
i22->h3 i23->h3 i24->h3 i25->h3 i26->h3 i27->h3 i28->h3 i29->h3 i30->h3 i31->h3 i32->h3
1.07 -1.39 0.49 -2.63 0.83 1.18 4.94 2.83 0.02 4.90 -1.71
i33->h3 i34->h3 i35->h3 i36->h3 i37->h3 i38->h3 i39->h3 i40->h3 i41->h3 i42->h3 i43->h3
-1.20 -4.93 1.47 1.06 0.51 -0.20 -3.98 -4.56 7.18 0.67 -13.15
i44->h3 i45->h3 i46->h3 i47->h3 i48->h3 i49->h3 i50->h3 i51->h3 i52->h3 i53->h3 i54->h3
1.38 -10.45 0.71 2.27 -1.05 -0.75 -3.31 3.87 -3.99 5.13 -0.02
i55->h3 i56->h3 i57->h3 i58->h3 i59->h3 i60->h3 i61->h3 i62->h3 i63->h3 i64->h3 i65->h3
2.14 -9.53 3.24 3.92 5.27 2.71 3.79 3.19 -1.98 -0.88 0.49
i66->h3 i67->h3 i68->h3 i69->h3 i70->h3 i71->h3 i72->h3 i73->h3 i74->h3 i75->h3 i76->h3
-0.52 1.35 3.31 1.23 -1.17 1.76 -2.07 1.80 1.73 3.09 2.70
i77->h3 i78->h3 i79->h3 i80->h3 i81->h3 i82->h3 i83->h3 i84->h3 i85->h3 i86->h3 i87->h3
0.02 3.12 -0.33 -1.43 1.89 5.28 -0.36 1.66 0.00 -3.39 -1.17
i88->h3 i89->h3 i90->h3 i91->h3 i92->h3 i93->h3
-2.00 0.28 6.24 1.60 3.53 3.97
b->h4 i1->h4 i2->h4 i3->h4 i4->h4 i5->h4 i6->h4 i7->h4 i8->h4 i9->h4 i10->h4
-2.22 -0.12 1.11 -0.30 0.15 -2.73 -6.02 -1.49 0.21 -0.79 -0.66
i11->h4 i12->h4 i13->h4 i14->h4 i15->h4 i16->h4 i17->h4 i18->h4 i19->h4 i20->h4 i21->h4
2.55 0.21 0.09 0.35 1.83 0.27 -1.71 0.67 -1.40 -1.42 1.38
i22->h4 i23->h4 i24->h4 i25->h4 i26->h4 i27->h4 i28->h4 i29->h4 i30->h4 i31->h4 i32->h4

0.04 -2.47 -0.53 1.89 5.04 0.64 -1.63 1.00 1.18 -0.01 -1.12
i33->h4 i34->h4 i35->h4 i36->h4 i37->h4 i38->h4 i39->h4 i40->h4 i41->h4 i42->h4 i43->h4
0.17 -1.96 -2.06 0.71 -2.72 -0.48 -1.79 0.23 0.96 2.63 0.98
i44->h4 i45->h4 i46->h4 i47->h4 i48->h4 i49->h4 i50->h4 i51->h4 i52->h4 i53->h4 i54->h4
0.13 -1.80 0.49 -0.84 0.29 0.75 -1.79 0.29 -1.60 -0.11 0.53
i55->h4 i56->h4 i57->h4 i58->h4 i59->h4 i60->h4 i61->h4 i62->h4 i63->h4 i64->h4 i65->h4
-1.40 -2.48 1.62 -0.35 0.48 4.25 2.20 0.16 0.68 0.09 -1.18
i66->h4 i67->h4 i68->h4 i69->h4 i70->h4 i71->h4 i72->h4 i73->h4 i74->h4 i75->h4 i76->h4
0.13 0.36 -1.60 -0.58 -0.65 -2.59 1.02 -0.01 0.89 -0.75 -0.75
i77->h4 i78->h4 i79->h4 i80->h4 i81->h4 i82->h4 i83->h4 i84->h4 i85->h4 i86->h4 i87->h4
-0.91 -2.98 0.17 0.39 -0.55 0.13 -0.55 -2.57 0.37 -0.12 0.73
i88->h4 i89->h4 i90->h4 i91->h4 i92->h4 i93->h4
0.71 -0.79 -1.71 1.70 1.43 2.14
b->h5 i1->h5 i2->h5 i3->h5 i4->h5 i5->h5 i6->h5 i7->h5 i8->h5 i9->h5 i10->h5
-0.13 -1.09 1.65 -1.01 -0.47 -1.18 -0.60 -1.69 0.34 3.72 0.30
i11->h5 i12->h5 i13->h5 i14->h5 i15->h5 i16->h5 i17->h5 i18->h5 i19->h5 i20->h5 i21->h5
-1.74 0.55 0.00 0.34 -0.01 0.70 -0.03 0.04 -2.03 -1.79 0.43
i22->h5 i23->h5 i24->h5 i25->h5 i26->h5 i27->h5 i28->h5 i29->h5 i30->h5 i31->h5 i32->h5
-0.49 -3.95 0.05 -0.12 -9.46 -0.46 -0.34 1.79 2.52 -1.37 -1.33
i33->h5 i34->h5 i35->h5 i36->h5 i37->h5 i38->h5 i39->h5 i40->h5 i41->h5 i42->h5 i43->h5
0.33 -1.47 1.44 1.77 0.14 -0.58 0.25 0.61 1.16 -5.20 2.96
i44->h5 i45->h5 i46->h5 i47->h5 i48->h5 i49->h5 i50->h5 i51->h5 i52->h5 i53->h5 i54->h5
-1.46 -0.57 0.81 -2.54 1.11 -1.23 -0.77 -1.71 2.25 2.58 -0.10
i55->h5 i56->h5 i57->h5 i58->h5 i59->h5 i60->h5 i61->h5 i62->h5 i63->h5 i64->h5 i65->h5
0.66 -0.30 -5.14 -1.85 3.33 -4.60 -2.43 1.79 0.40 0.14 0.97
i66->h5 i67->h5 i68->h5 i69->h5 i70->h5 i71->h5 i72->h5 i73->h5 i74->h5 i75->h5 i76->h5
-0.57 0.11 -1.20 -2.29 0.02 -1.20 0.45 0.01 -0.81 -0.44 -0.73
i77->h5 i78->h5 i79->h5 i80->h5 i81->h5 i82->h5 i83->h5 i84->h5 i85->h5 i86->h5 i87->h5
6.41 -4.03 -0.72 -0.55 -1.97 0.37 3.81 -3.97 0.52 0.48 -1.51
i88->h5 i89->h5 i90->h5 i91->h5 i92->h5 i93->h5
0.23 1.81 -1.79 1.30 -0.99 0.42
b->o1 h1->o1 h2->o1 h3->o1 h4->o1 h5->o1
-0.41 -1.85 -0.95 3.69 -0.76 0.68
b->o2 h1->o2 h2->o2 h3->o2 h4->o2 h5->o2
0.62 0.96 -0.56 -2.43 1.50 1.88
b->o3 h1->o3 h2->o3 h3->o3 h4->o3 h5->o3
0.90 0.03 -0.68 -3.33 0.61 2.38
b->o4 h1->o4 h2->o4 h3->o4 h4->o4 h5->o4
0.69 -0.38 -1.03 -3.11 2.93 -0.75
b->o5 h1->o5 h2->o5 h3->o5 h4->o5 h5->o5
-0.26 5.15 -1.46 -4.66 -4.25 -4.47
b->o6 h1->o6 h2->o6 h3->o6 h4->o6 h5->o6
0.27 -2.13 2.58 1.67 2.39 -1.58
b->o7 h1->o7 h2->o7 h3->o7 h4->o7 h5->o7
2.93 -2.86 1.72 -1.08 -1.52 -0.07
b->o8 h1->o8 h2->o8 h3->o8 h4->o8 h5->o8
-0.54 1.59 1.31 3.66 -2.21 -0.64
b->o9 h1->o9 h2->o9 h3->o9 h4->o9 h5->o9
-3.92 -0.80 -1.66 5.59 1.56 2.67

```
summary(ps)
   Class_1           Class_2           Class_3           Class_4
Min.   :0.0003175   Min.   :0.0004027   Min.   :0.0000802   Min.   :0.0000436
1st Qu.:0.0007202   1st Qu.:0.0005650   1st Qu.:0.0001154   1st Qu.:0.0005764
Median :0.0046876   Median :0.0331889   Median :0.0174847   Median :0.0092618
Mean   :0.0275962   Mean   :0.2562176   Mean   :0.1332563   Mean   :0.0458221
3rd Qu.:0.0109750   3rd Qu.:0.6104054   3rd Qu.:0.2462515   3rd Qu.:0.0583670
Max.   :0.4460607   Max.   :0.7104585   Max.   :0.5125156   Max.   :0.5673775
   Class_5           Class_6           Class_7           Class_8
Min.   :0.0000000   Min.   :0.0004977   Min.   :0.0003894   Min.   :0.0002611
1st Qu.:0.0000001   1st Qu.:0.0016959   1st Qu.:0.0021075   1st Qu.:0.0008596
Median :0.0001054   Median :0.0217462   Median :0.0073123   Median :0.0090453
Mean   :0.0430220   Mean   :0.2294358   Mean   :0.0462082   Mean   :0.1365114
3rd Qu.:0.0006822   3rd Qu.:0.1708350   3rd Qu.:0.0154313   3rd Qu.:0.0233542
Max.   :0.9113590   Max.   :0.9739285   Max.   :0.8554974   Max.   :0.9658130
   Class_9
Min.   :0.0000296
1st Qu.:0.0022342
Median :0.0033974
Mean   :0.0819304
3rd Qu.:0.0097583
Max.   :0.8958361
```

For each class mean and median of each quartile is derived after the prediction.

RStudio

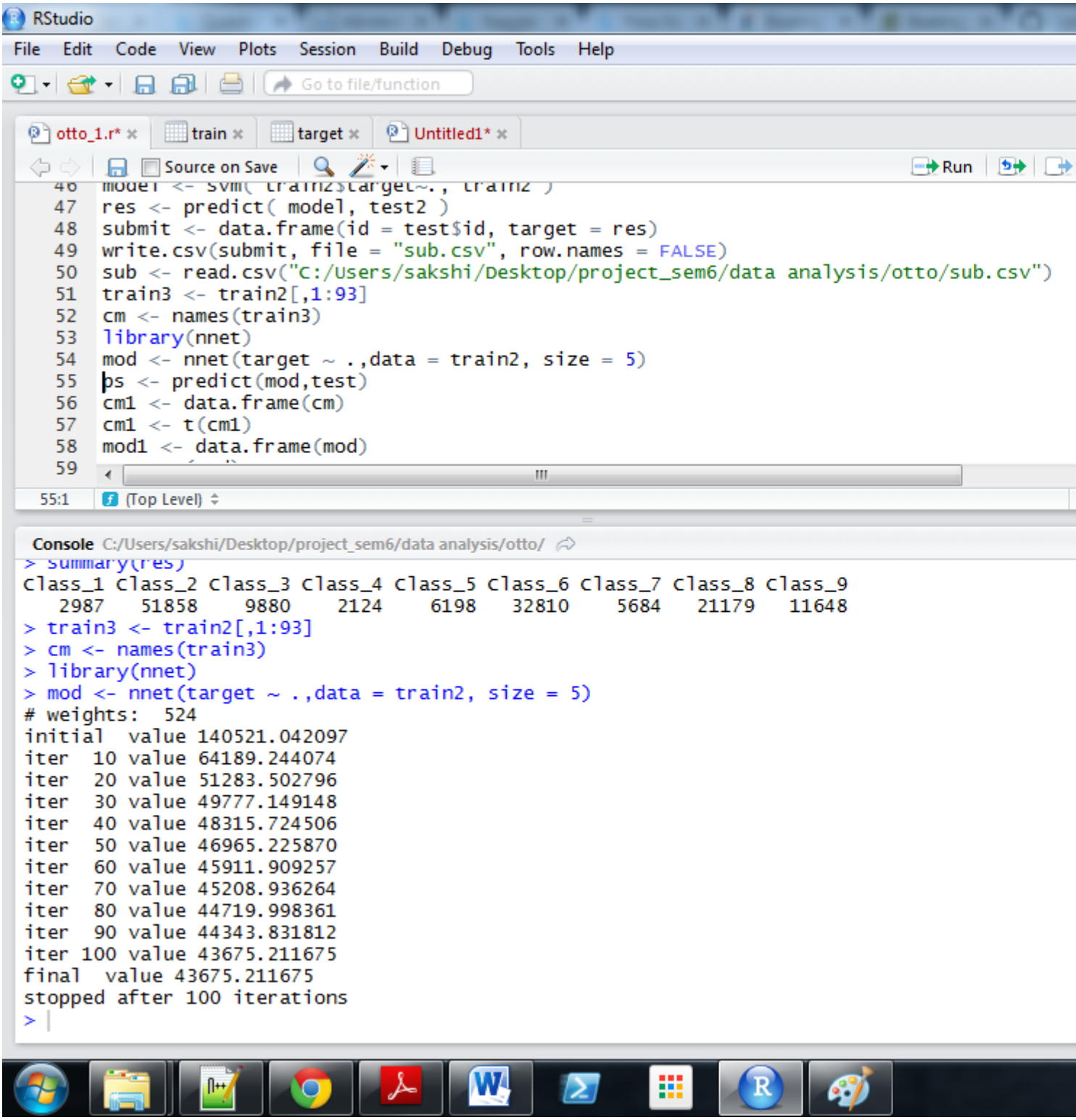
File Edit Code View Plots Session Build Debug Tools Help

Go to file/function

Source

Console C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/

```
# weights:  524
initial  value 144757.897082
iter  10 value 64689.034043
iter  20 value 51962.212856
iter  30 value 47240.122457
iter  40 value 46007.884662
iter  50 value 45311.901219
iter  60 value 44887.390533
iter  70 value 44480.603307
iter  80 value 44165.083182
iter  90 value 43916.004247
iter 100 value 43696.945086
final  value 43696.945086
stopped after 100 iterations
> ps <- predict(mod,test)
> summary(ps)
    Class_1           Class_2           Class_3           Class_4
 Min.   :0.0003175   Min.   :0.0004027   Min.   :0.0000802   Min.   :0.0000436
 1st Qu.:0.0007202   1st Qu.:0.0005650   1st Qu.:0.0001154   1st Qu.:0.0005764
 Median :0.0046876   Median :0.0331889   Median :0.0174847   Median :0.0092618
 Mean   :0.0275962   Mean   :0.2562176   Mean   :0.1332563   Mean   :0.0458221
 3rd Qu.:0.0109750   3rd Qu.:0.6104054   3rd Qu.:0.2462515   3rd Qu.:0.0583670
 Max.   :0.4460607   Max.   :0.7104585   Max.   :0.5125156   Max.   :0.5673775
    Class_5           Class_6           Class_7           Class_8
 Min.   :0.0000000   Min.   :0.0004977   Min.   :0.0003894   Min.   :0.0002611
 1st Qu.:0.0000001   1st Qu.:0.0016959   1st Qu.:0.0021075   1st Qu.:0.0008596
 Median :0.0001054   Median :0.0217462   Median :0.0073123   Median :0.0090453
 Mean   :0.0430220   Mean   :0.2294358   Mean   :0.0462082   Mean   :0.1365114
 3rd Qu.:0.0006822   3rd Qu.:0.1708350   3rd Qu.:0.0154313   3rd Qu.:0.0233542
 Max.   :0.9113590   Max.   :0.9739285   Max.   :0.8554974   Max.   :0.9658130
    Class_9
 Min.   :0.0000296
 1st Qu.:0.0022342
 Median :0.0033974
 Mean   :0.0819304
 3rd Qu.:0.0097583
 Max.   :0.8958361
>
```


RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
otto_1.r*
train
target
Untitled1*
Source on Save
Run
model <- svm(train2$target~., train2)
res <- predict(model, test2)
submit <- data.frame(id = test$id, target = res)
write.csv(submit, file = "sub.csv", row.names = FALSE)
sub <- read.csv("C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/sub.csv")
train3 <- train2[,1:93]
cm <- names(train3)
library(nnet)
mod <- nnet(target ~ .,data = train2, size = 5)
ps <- predict(mod,test)
cm1 <- data.frame(cm)
cm1 <- t(cm1)
mod1 <- data.frame(mod)
55:1
(Top Level)
Console C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/
> summary(res)
Class_1 Class_2 Class_3 Class_4 Class_5 Class_6 Class_7 Class_8 Class_9
2987 51858 9880 2124 6198 32810 5684 21179 11648
> train3 <- train2[,1:93]
> cm <- names(train3)
> library(nnet)
> mod <- nnet(target ~ .,data = train2, size = 5)
weights: 524
initial value 140521.042097
iter 10 value 64189.244074
iter 20 value 51283.502796
iter 30 value 49777.149148
iter 40 value 48315.724506
iter 50 value 46965.225870
iter 60 value 45911.909257
iter 70 value 45208.936264
iter 80 value 44719.998361
iter 90 value 44343.831812
iter 100 value 43675.211675
final value 43675.211675
stopped after 100 iterations
>

RStudio

File Edit Code View Plots Session Build Debug Tools Help

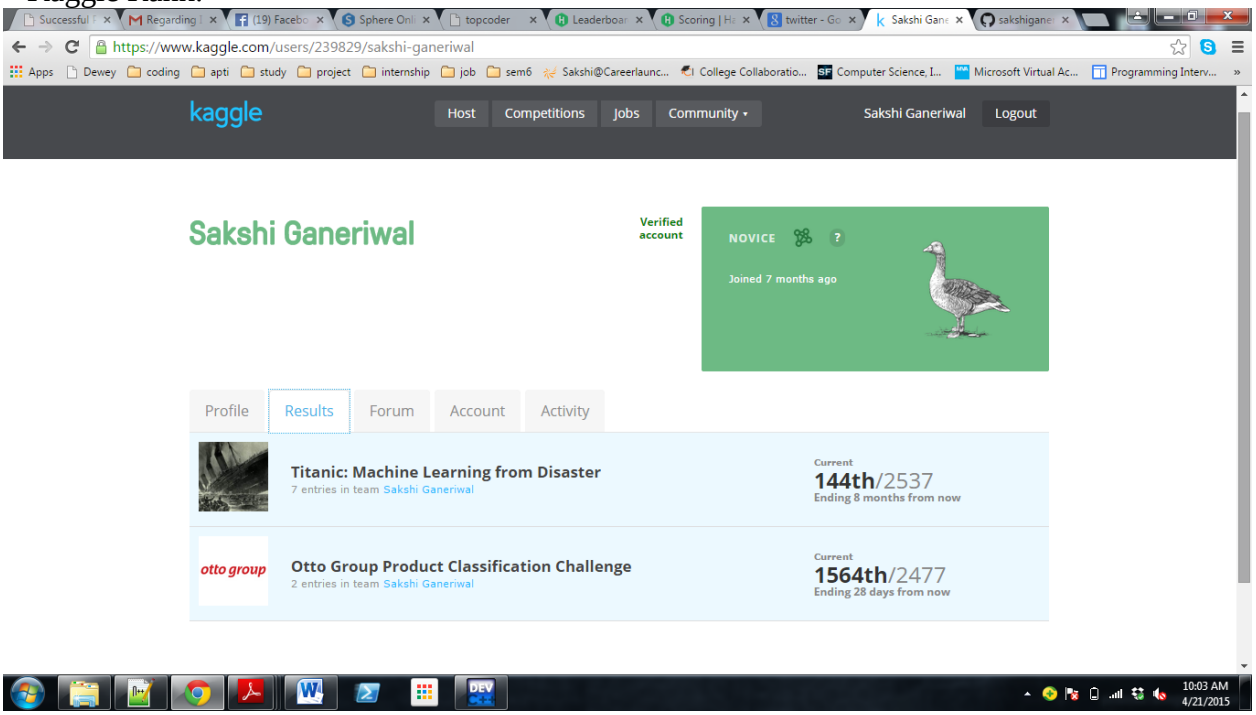
Go to file/function

Source

Console C:/Users/sakshi/Desktop/project_sem6/data analysis/otto/

```
> summary(mod)
a 93-5-9 network with 524 weights
options were - softmax modelling
  b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9->h1 i10->h1
  -4.42    0.40   -0.25    0.11    0.97    1.19   -0.31    2.82   -0.22    3.56    0.05
i11->h1 i12->h1 i13->h1 i14->h1 i15->h1 i16->h1 i17->h1 i18->h1 i19->h1 i20->h1 i21->h1
  -6.06   -2.82    3.56    1.57    3.64   -0.75   -5.78    1.22    0.87   -0.12    1.31
i22->h1 i23->h1 i24->h1 i25->h1 i26->h1 i27->h1 i28->h1 i29->h1 i30->h1 i31->h1 i32->h1
   0.41   -2.76   -0.73    2.98   -2.74   -2.85    0.87   -1.03    0.53   -1.20   -2.45
i33->h1 i34->h1 i35->h1 i36->h1 i37->h1 i38->h1 i39->h1 i40->h1 i41->h1 i42->h1 i43->h1
   4.93    3.25    4.67   -1.97    0.94    1.89   -5.09    3.28    2.47   -2.36   -5.66
i44->h1 i45->h1 i46->h1 i47->h1 i48->h1 i49->h1 i50->h1 i51->h1 i52->h1 i53->h1 i54->h1
  -2.30   -1.24    0.22    1.41    1.00   -2.58   -2.81    0.70   -0.96   -6.95    0.84
i55->h1 i56->h1 i57->h1 i58->h1 i59->h1 i60->h1 i61->h1 i62->h1 i63->h1 i64->h1 i65->h1
  -3.20   -6.63   -3.73    3.68   -5.94   -6.27   -3.37   -0.37    1.35   -0.19    0.85
i66->h1 i67->h1 i68->h1 i69->h1 i70->h1 i71->h1 i72->h1 i73->h1 i74->h1 i75->h1 i76->h1
  -2.39   -0.09    2.93    3.24   -0.53    1.83    2.31    0.57   -0.09    4.11    0.44
i77->h1 i78->h1 i79->h1 i80->h1 i81->h1 i82->h1 i83->h1 i84->h1 i85->h1 i86->h1 i87->h1
  -2.63    1.24   -4.13    2.13   -3.29   -3.92    0.61    0.57   -5.16   -1.14   -2.12
i88->h1 i89->h1 i90->h1 i91->h1 i92->h1 i93->h1
  -2.26    3.05    6.43   -1.21    0.54   -1.51
  b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9->h2 i10->h2
 -12.77   -5.37    4.00    0.67    0.19   -8.69   -5.41   -4.40   -9.55    2.18   -2.43
i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2 i18->h2 i19->h2 i20->h2 i21->h2
   8.45   -1.88    1.48   -6.23   -1.36    0.37    0.93    1.34    5.39    1.94    1.43
i22->h2 i23->h2 i24->h2 i25->h2 i26->h2 i27->h2 i28->h2 i29->h2 i30->h2 i31->h2 i32->h2
   1.27    3.16   -3.18   -3.42    2.44   -1.12    2.28    2.21  -12.67    0.34    2.22
i33->h2 i34->h2 i35->h2 i36->h2 i37->h2 i38->h2 i39->h2 i40->h2 i41->h2 i42->h2 i43->h2
  -3.21   -1.51    3.65   -1.28   -4.22    2.10   10.98  -14.01   -0.31    8.54   -2.51
i44->h2 i45->h2 i46->h2 i47->h2 i48->h2 i49->h2 i50->h2 i51->h2 i52->h2 i53->h2 i54->h2
  -0.02    3.21    0.32   10.31   -2.55    3.25    6.46   -0.42   -0.89   -3.34    0.68
i55->h2 i56->h2 i57->h2 i58->h2 i59->h2 i60->h2 i61->h2 i62->h2 i63->h2 i64->h2 i65->h2
   1.50   -3.75    6.48    4.49   -0.17   12.85   -0.49   -6.17    2.17   -7.33    0.32
i66->h2 i67->h2 i68->h2 i69->h2 i70->h2 i71->h2 i72->h2 i73->h2 i74->h2 i75->h2 i76->h2
  -3.34   -0.86  -10.30    3.70    0.11    1.40  -13.06    4.08    6.08    3.77   -2.37
i77->h2 i78->h2 i79->h2 i80->h2 i81->h2 i82->h2 i83->h2 i84->h2 i85->h2 i86->h2 i87->h2
  -8.46   -6.34   -4.50    2.06    2.40    1.20    4.01  -12.70   -7.90    0.84    2.14
i88->h2 i89->h2 i90->h2 i91->h2 i92->h2 i93->h2
  -4.50   -1.89    2.82   -0.73   -0.84    5.02
```

Kaggle Rank:


https://www.kaggle.com/users/239829/sakshi-ganeriwal
kaggle
Host
Competitions
Jobs
Community
Sakshi Ganeriwal
Logout
Sakshi Ganeriwal
Verified account
NOVICE
Joined 7 months ago
Profile
Results
Forum
Account
Activity
Titanic: Machine Learning from Disaster
7 entries in team Sakshi Ganeriwal
Current
144th/2537
Ending 8 months from now
otto group
Otto Group Product Classification Challenge
2 entries in team Sakshi Ganeriwal
Current
1564th/2477
Ending 28 days from now
10:03 AM
4/21/2015