```
> #import dataset
> data=read.csv(file.choose(),sep = ',',header = T)
> # View dataset
> View(data)
> #feature selection
> data=data[,3:16]
> # After feature selection view the dataset
> View(data)
> # Understand the dataset
> str(data)
'data.frame':  731 obs. of  14 variables:
 $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ weekday   : int  6 0 1 2 3 4 5 6 0 1 ...
 $ workingday: int  0 0 1 1 1 1 1 0 0 1 ...
 $ weathersit: int  2 2 1 1 1 1 2 2 1 1 ...
 $ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
 $ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
 $ hum       : num  0.806 0.696 0.437 0.59 0.437 ...
 $ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
 $ casual    : int  331 131 120 108 82 88 148 68 54 41 ...
 $ registered: int  654 670 1229 1454 1518 1518 1362 891 768 1280 ...
 $ cnt       : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
> #Attach the datset with the system so we can use directly variable
> attach(data)
> #View the dataset on console
> head(data)
  season yr mnth holiday weekday workingday weathersit     temp    atemp
1      1  0    1       0       6          0          2 0.344167 0.363625
2      1  0    1       0       0          0          2 0.363478 0.353739
3      1  0    1       0       1          1          1 0.196364 0.189405
4      1  0    1       0       2          1          1 0.200000 0.212122
5      1  0    1       0       3          1          1 0.226957 0.229270
6      1  0    1       0       4          1          1 0.204348 0.233209
       hum windspeed casual registered  cnt
1 0.805833 0.1604460    331        654  985
2 0.696087 0.2485390    131        670  801
3 0.437273 0.2483090    120       1229 1349
4 0.590435 0.1602960    108       1454 1562
5 0.436957 0.1869000     82       1518 1600
6 0.518261 0.0895652     88       1518 1606
> # Normalization of the few columns  like casual and registered
> data$casual=(data$casual-min(data$casual))/(max(data$casual)-min(data$casua
l))
> data$registered=(data$registered-min(data$registered))/(max(data$registered
)-min(data$registered))
> head(data)
  season yr mnth holiday weekday workingday weathersit     temp    atemp
1      1  0    1       0       6          0          2 0.344167 0.363625
2      1  0    1       0       0          0          2 0.363478 0.353739
3      1  0    1       0       1          1          1 0.196364 0.189405
4      1  0    1       0       2          1          1 0.200000 0.212122
5      1  0    1       0       3          1          1 0.226957 0.229270
6      1  0    1       0       4          1          1 0.204348 0.233209
       hum windspeed     casual registered  cnt
```

```
1 0.805833 0.1604460 0.09653756 0.09153913  985
2 0.696087 0.2485390 0.03785211 0.09384926  801
3 0.437273 0.2483090 0.03462441 0.17455963 1349
4 0.590435 0.1602960 0.03110329 0.20704591 1562
5 0.436957 0.1869000 0.02347418 0.21628646 1600
6 0.518261 0.0895652 0.02523474 0.21628646 1606
> # Devide dataset in the train and test case
> ind= sample(2,nrow(data),replace=T,prob=c(0.8 ,0.2))
> train=data[ind==1,]
> test=data[ind==2,]
> # Apply linear regresson on the dataset
> model=lm(cnt~.,train)
> #find rmse value
> rmse <- function(error)
+ {
+   sqrt(mean(error^2))
+ }
> error <- model$residuals
> predictionRMSE <- rmse(error)
> predictionRMSE
[1] 3.508913e-12
> # Predict the dataset
> predictedy=predict(model,test)
> predictedy
   4    5    8   11   16   20   21   24   31   32   50   59   65   67
1562 1600  959 1263 1204 1927 1543 1416 1501 1360 1635 1446  605 2133
  68   87   88   89  104  106  107  111  114  118  126  132  137  139
1891 2028 2425 1536 3267  795 3744 4189 4191 4058 4608 4864 4123 4575
 145  151  173  179  181  189  190  193  195  202  206  219  220  222
4978 3982 4507 4648 5515 4040 5336 4258 5084 3784 3840 3785 4326 4780
 230  238  240  248  249  260  261  262  264  271  277  294  296  297
3805 4661 4334 3351 2710 4511 4274 4539 4352 3907 4456 4304 4381 4187
 316  317  320  321  327  330  334  340  347  352  356  360  363  373
4067 3717 1817 3053 2566 3068 3613 2594 3523 2431 3068 1317 2423 3425
 376  380  386  391  400  401  403  412  417  425  430  431  434  445
2177 2311 1301 4075 2832 2947 4375 3005 3777 1834 3333 3956 4569 6093
 456  457  461  470  472  482  485  490  491  496  509  513  520  527
6235 6041 6457 7460 6370 5026 6304 6296 6883 6572 5260 6591 7641 6598
 529  531  534  538  541  543  546  548  549  554  562  568  570  572
4972 7363 6978 5905 6891 7442 5463 5531 6227 4840 6031 4459 6966 8173
 575  581  582  583  585  589  593  596  599  606  608  614  617  619
6685 7175 6824 5464 7273 6299 7347 7865 7006 7040 7713 7112 5976 7525
 621  622  628  632  637  639  643  652  656  657  661  662  667  677
7870 7804 7591 7907 7415 6889 7328 7109 7461 7509 7058 7466 4459 5035
 683  684  685  701  704  719  724  727
4094 5495 5445 5191 6606 5267  920 2114
> #plot the model
> plot(model)
Hit <Return> to see next plot: #View the whole summary of the model
Hit <Return> to see next plot: summary.lm(model)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
> (Tab=table(predictedy,test$cnt))

predictedy         605 795 920 959 1204 1263 1301 1317 1360 1416
  605.000000000003   1   0   0   0    0    0    0    0    0    0
  795.000000000003   0   1   0   0    0    0    0    0    0    0
```

920.000000000004	0	0	1	0	0	0	0	0	0	0
959.000000000002	0	0	0	1	0	0	0	0	0	0
1204	0	0	0	0	1	0	0	0	0	0
1263	0	0	0	0	0	1	0	0	0	0

predictedy	1446	1501	1536	1543	1562	1600	1635	1817	1834
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	1891	1927	2028	2114	2133	2177	2311	2423	2425
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	2431	2566	2594	2710	2832	2947	3005	3053	3068
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	3267	3333	3351	3425	3523	3613	3717	3744	3777
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	3784	3785	3805	3840	3907	3956	3982	4040	4058
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	4067	4075	4094	4123	4187	4189	4191	4258	4274
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	4304	4326	4334	4352	4375	4381	4456	4459	4507
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0

959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	4511	4539	4569	4575	4608	4648	4661	4780	4840
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	4864	4972	4978	5026	5035	5084	5191	5260	5267
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	5336	5445	5463	5464	5495	5515	5531	5905	5976
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	6031	6041	6093	6227	6235	6296	6299	6304	6370
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	6457	6572	6591	6598	6606	6685	6824	6883	6889
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	6891	6966	6978	7006	7040	7058	7109	7112	7175
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0	0	0
1263	0	0	0	0	0	0	0	0	0

predictedy	7273	7328	7347	7363	7415	7442	7460	7461	7466
605.000000000003	0	0	0	0	0	0	0	0	0
795.000000000003	0	0	0	0	0	0	0	0	0
920.000000000004	0	0	0	0	0	0	0	0	0
959.000000000002	0	0	0	0	0	0	0	0	0

```
  1204                0    0    0    0    0    0    0    0    0
  1263                0    0    0    0    0    0    0    0    0

predictedy         7509 7525 7591 7641 7713 7804 7865 7870 7907
  605.000000000003    0    0    0    0    0    0    0    0    0
  795.000000000003    0    0    0    0    0    0    0    0    0
  920.000000000004    0    0    0    0    0    0    0    0    0
  959.000000000002    0    0    0    0    0    0    0    0    0
  1204                0    0    0    0    0    0    0    0    0
  1263                0    0    0    0    0    0    0    0    0

predictedy         8173
  605.000000000003    0
  795.000000000003    0
  920.000000000004    0
  959.000000000002    0
  1204                0
  1263                0
 [ reached getOption("max.print") -- omitted 140 rows ]
> (SVMPERFORMANCE=sum(diag(Tab))/sum(Tab)*100)
[1] 100
```