PredictionWriteup_smartDevices

SS

23/06/2021

Loading Libraries

```
library(caret)
library(knitr)
library(rpart)
library(rpart.plot)
library(randomForest)
```

Download the dataset using predefined urls

```
trainUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"

testUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
```

Load the dataset into variables

Post Cleaning of data variables i.e marking unusable fields as NA from #DIV/0!

```
training_Data <- read.csv(url(trainUrl), na.strings = c("NA", "#DIV/0!", ""))

testing_Data <- read.csv(url(testUrl), na.strings = c("NA", "#DIV/0!", ""))
```

Data Transformations i.e cleaning of data

Getting rid of unwanted data elements i.e NA

```
training_Data <- training_Data[, colSums(is.na(training_Data)) == 0]

testing_Data <- testing_Data[, colSums(is.na(testing_Data)) == 0]

head(training_Data)
head(testing_Data)
```

Deleting Columns which are not related

Removing Columns which are not required for prediction purpose

```
training_Data <- training_Data[, -c(1:7)]

testing_Data <- testing_Data[, -c(1:7)]
```

Final Snapshot of Data to be used as input to models

```
head(training_Data)
head(testing_Data)
```

Partioning the training set into two different dataset

Splitting the datasets into two parts with 70 percent in training set and 30 percent in testing data set.

```
traning_Partition_Data <- createDataPartition(training_Data$classe,  p = 0.7,
list = F)

training_DataSet <- training_Data[traning_Partition_Data, ]

testing_DataSet <- training_Data[-traning_Partition_Data, ]
```

Checking Dimesnions for both sets

```
dim(training_Data)

dim(testing_DataSet)
```

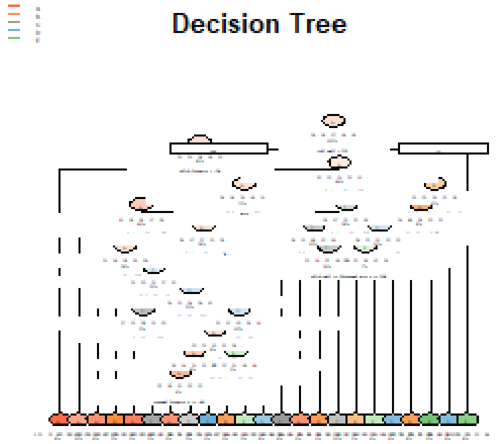
# Prediction Model 1 - using Decision Tree

```
decision_Tree_Model <- rpart(classe ~ ., data = training_DataSet, method =
"class")

decision_Tree_Prediction <- predict(decision_Tree_Model, testing_DataSet,
type = "class")
```

# Ploting Decision Tree

```
rpart.plot(decision_Tree_Model, main = "Decision Tree", under = T, faclen =
0)
```



Decision Tree.

# Applying confusion matrix to test results

```
confusionMatrix(factor(decision_Tree_Prediction),
factor(testing_DataSet$classe))

Overall Statistics

               Accuracy : 0.7336
                 95% CI : (0.7221, 0.7448)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6613
```

# Prediction model 2 -using Random Forest

```
training_DataSet$classe = factor(training_DataSet$classe)

random_Forest_Model <- randomForest(classe ~. , data = training_DataSet,
method = "class")

random_Forest_Prediction <- predict(random_Forest_Model, testing_DataSet,
type = "class")

confusionMatrix(factor(random_Forest_Prediction),
factor(testing_DataSet$classe))

Overall Statistics

               Accuracy : 0.9949
                 95% CI : (0.9927, 0.9966)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9936
```

Final Prediction using RF method

```
Final_Prediction <- predict(random_Forest_Model, testing_DataSet, type =
"class")

Final_Prediction
```

Conclusion

Accuracy level of Random Forest Model is better than that of decision tree model as it is evident from the model statistics