Project by Olivia

In this project, my goal is to build a prediction algorithem based on "train.csv" to better forcast the "classe" value of "test.csv".In part 1, I have use PCA to shrink the number of predictors from 160 to 52. In part 2, i introduced 1) Tree 2) Linear Discrimination 3) General Boosting algorithem to build my model. And I found that:1) PCA did not help improve the accuracy of our prediction.(2).For dataset without PCA and with PCA, the rank among my models is : GBM > LDS > Tree. Finaly, I used GBM model to forcast the value of classe on testing set is: B A B A A E D B A A B C B A E E A B B B

Part 1. Introduction to dataset (training and testing)

```
library(ggplot2);library(caret);library(caret);library(gridExtra);library(rpart.plot);
```

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

```
## Warning: package 'caret' was built under R version 3.1.3
```

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

```
## Warning: package 'lattice' was built under R version 3.1.3
```

```
## Loading required package: grid
```

```
## Warning: package 'rpart.plot' was built under R version 3.1.2
```

```
## Loading required package: rpart
```

```
## Warning: package 'rpart' was built under R version 3.1.2
```

```
library(rattle);library(gbm); library(survival);require(MASS);require(plyr);require(knitr)
```

```
## Warning: package 'rattle' was built under R version 3.1.2
```

```
## Rattle: A free graphical interface for data mining with R.
## Version 3.4.1 Copyright (c) 2006-2014 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
## Warning: package 'gbm' was built under R version 3.1.3
```

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 3.1.2
```

```
##
## Attaching package: 'survival'
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.1
```

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

```
## Warning: package 'MASS' was built under R version 3.1.3
```

```
## Loading required package: plyr
```

```
## Warning: package 'plyr' was built under R version 3.1.3
```

```
## Loading required package: knitr
```

```
f1=download.file('https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv',
                 destfile = '/Users/apple/Desktop/Cousera/Data Science/4.Machine Learning JHopkins/trai
```

```
## Warning in download.file("https://d396qusza40orc.cloudfront.net/
## predmachlearn/pml-training.csv", : download had nonzero exit status
```

```
train=read.csv('train.csv',na.strings=c("NA","#DIV/0!", ""))

f2=download.file('https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv',
                 destfile = '/Users/apple/Desktop/Cousera/Data Science/4.Machine Learning JHopkins/test
```

```
## Warning in download.file("https://d396qusza40orc.cloudfront.net/
## predmachlearn/pml-testing.csv", : download had nonzero exit status
```

```
test=read.csv('test.csv',na.strings=c("NA","#DIV/0!", ""))

dim(train); dim(test)
```

```
## [1] 19622   160
```

```
## [1]  20 160
```

1.Training dataset : 19622 observations and 160 features.
2.Testing dataset : 20 observations and 160 features.
3.Variable'classe' : 5 levels: A, B, C, D , E.
4.Features(Reducing irrelavant variables)
(1)Are the features in training set are the same with testing set? If not, which features are not the same?

```
all.equal(colnames(train),colnames(test))                 # "1 string mismatch"
(colnames(train))[which(colnames(train)!=colnames(test))] # "classe"
(colnames(test))[which(colnames(train)!=colnames(test))]  # "problem_id"
```

(2)Which featrues are correlated to 'Classe'?
First, I delete the first 7 columns, they are irrelavant to the 'Classe'

```
train=train[,-c(1:7)];test =test[,-c(1:7)]
```

Second, Delete columns with all missing values. Then we will have 53 non-missing features.

```
train<-train[,colSums(is.na(train)) == 0]; test <-test[,colSums(is.na(test)) == 0]
which(colnames(train)=='classe');which(colnames(test)=='problem_id')     # 53
```

Third, Using PCA to reduce the colinearity among our 53 features.

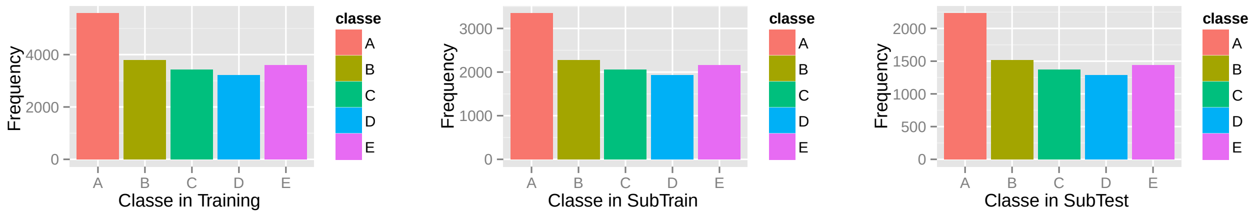
```
PCA<- preProcess(train[,-53],method="pca")
TrainPca <- predict(PCA,train[,-53])   ;   TestPca <- predict(PCA,test[,-53])
TrainPca$classe=train$classe             # dim(TrainPca)  dim(TestPca)
#Finally, after cleaning data, I have 25 PCA, 52 predictors and 1 response 'classe'.
```

5.Partition the training set: subtrain set as 11776 obs(60%) and subtest as 7846 obs(40%).

```
sub <- createDataPartition(y=TrainPca$classe, p=0.6, list=FALSE)
subTrain <- TrainPca[sub, ]     ;    subTest <- TrainPca[-sub, ]
```

6.After partitioning training set,another question is : Whether 'classe' in both set has the same distribution?

```
# The following histogram show you the distribution of 'classe' in training dataset.
g1=ggplot(train, aes(classe,fill=classe))+geom_histogram(binwidth=1)
g1=g1+xlab("Classe in Training ")+ylab("Frequency ")
g2=ggplot(subTrain, aes(classe,fill=classe))+geom_histogram(binwidth=1)
g2=g2+xlab("Classe in SubTrain ")+ylab("Frequency")
g3=ggplot(subTest, aes(classe,fill=classe))+geom_histogram(binwidth=1)
g3=g3+xlab("Classe in SubTest ")+ylab("Frequency")
grid.arrange(g1,g2,g3,ncol=3)
```



The above 2 graphs show us the weight among A,B,C,D,E is around the same in both subtrain and subtest.Thus it is relible that we validate our classification model(buillt by subtrain) ,on subtest set.

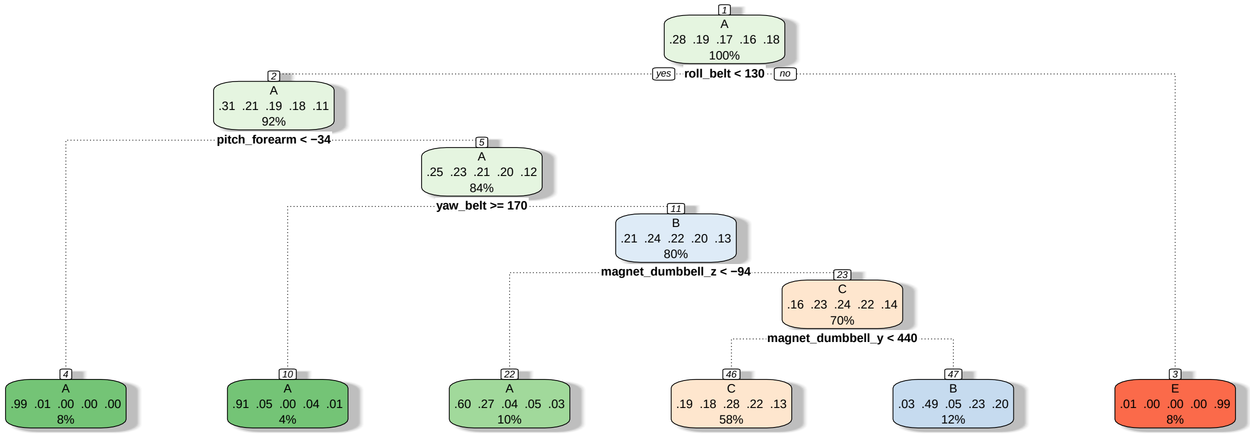
Part 2. Classification with subtraining and subtesting data

1.1 Prediction with Trees, using PCA

```
control=trainControl(method ="cv",number=7)
mTree=train(classe~.,data=subTrain,trControl=control,method='rpart')
a1=confusionMatrix(predict(mTree,subTrain), subTrain$classe)$overall['Accuracy']
a2=confusionMatrix(predict(mTree,subTest) , subTest$classe)$overall['Accuracy']
```

1.2. Prediction with Trees,without PCA

```
sub2<-createDataPartition(y=train$classe,p=0.6,list=F)
subTrain2 <- train[sub2, ] ; subTest2 <- train[-sub2, ]
mTree2=train(classe~., data=subTrain2,trControl=control,method='rpart')
b1=confusionMatrix(predict(mTree2,subTrain2),subTrain2$classe)$overall['Accuracy']
b2=confusionMatrix(predict(mTree2,subTest2),subTest2$classe)$overall['Accuracy']
fancyRpartPlot(mTree2$finalModel,sub='Tree without PCA')
```



Tree without PCA

To sum up, Tree without PCA has accuracy 0.4959 in subtest set, higher than that with PCA(0.3776).

2.1 Prediction using Linear Discrimination, with PCA

```
mLds=train(classe~., data=subTrain,trControl=control,method='lda',verbose=F)
c1=confusionMatrix(predict(mLds,subTrain), subTrain$classe)$overall['Accuracy']# Accuracy(subTrain)
c2=confusionMatrix(predict(mLds,subTest)  , subTest$classe)$overall['Accuracy'] # Accuracy(subTest)
```

2.2 Prediction using Linear Discrimination, without PCA

```
mLds2=train(classe~., data=subTrain2,trControl=control,method='lda')
d1=confusionMatrix(predict(mLds2,subTrain2), subTrain2$classe)$overall['Accuracy']
d2=confusionMatrix(predict(mLds2,subTest2) , subTest2$classe)$overall['Accuracy']
```

3.1 Prediction using Generalized Boosted Regression Model, with PCA

```
mGbm=train(classe~., data=subTrain,method='gbm',trControl=control,verbose=F)
f1=confusionMatrix(predict(mGbm,subTrain), subTrain$classe)$overall['Accuracy'];# Accuracy(subTrain)
f2=confusionMatrix(predict(mGbm,subTest)  , subTest$classe)$overall['Accuracy'] # Accuracy(subTest)
```

3.2 Prediction using Generalized Boosted Regression Model, without PCA

```
mGbm2=train(classe~., data=subTrain2,method='gbm',trControl=control,verbose=F)
g1=confusionMatrix(predict(mGbm2,subTrain2), subTrain2$classe)$overall['Accuracy']
g2=confusionMatrix(predict(mGbm2,subTest2) , subTest2$classe)$overall['Accuracy']
confusionMatrix(predict(mGbm2,subTest2),subTest2$classe)
```

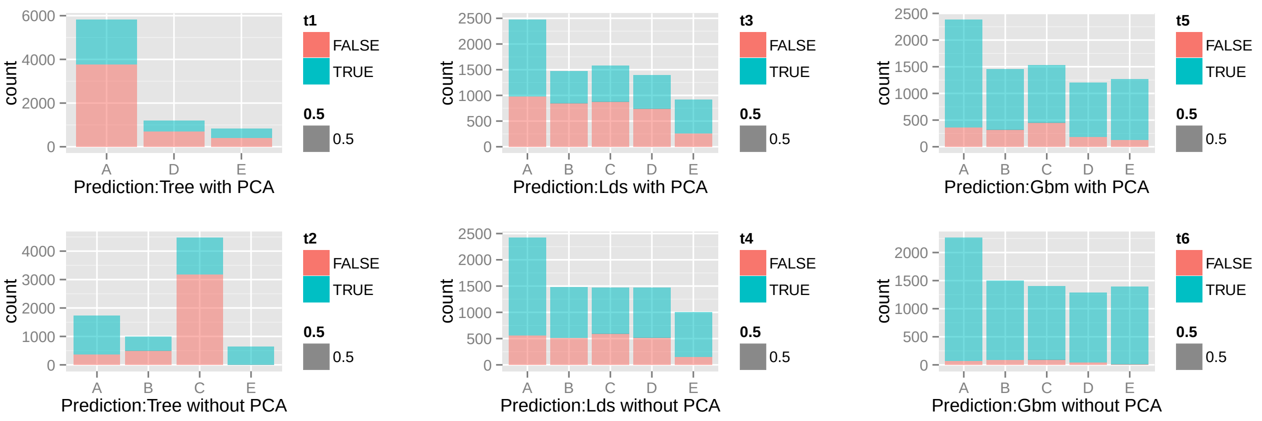
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2198   60    0    1    4
##          B   25 1411   39    4   21
##          C    6   41 1311   33   11
##          D    3    3   16 1242   19
##          E    0    3    2    6 1387
##
## Overall Statistics
##
##                Accuracy : 0.9621
##                  95% CI : (0.9577, 0.9663)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9521
##  Mcnemar's Test P-Value : 8.67e-09
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9848   0.9295   0.9583   0.9658   0.9619
## Specificity            0.9884   0.9859   0.9860   0.9938   0.9983
## Pos Pred Value         0.9713   0.9407   0.9351   0.9680   0.9921
## Neg Pred Value         0.9939   0.9831   0.9912   0.9933   0.9915
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2801   0.1798   0.1671   0.1583   0.1768
## Detection Prevalence   0.2884   0.1912   0.1787   0.1635   0.1782
## Balanced Accuracy      0.9866   0.9577   0.9721   0.9798   0.9801
```

Summary * Combined Tree, LDS and GBM together, I got 3 conclusions:
(1). The PCA did not significantly improve the accuracy of our prediction on subtesting set.
(2). For dataset without PCA, the rank among my models is :
GBM (96%) > LDS (70%) > Tree(50%).
For dataset with PCA, the rank among my models is :
GBM (82%) > LDS (53%) > Tree(38%).
(3). Model selection can significantly improve the accuracy of prediction.
(4). General Boosting Model is the best model for this classification.

4. Visualize the Prediction of Different Model

```
pTree=predict(mTree,subTest) ; pTree2=predict(mTree2,subTest2)
pLds=predict(mLds,subTest)   ; pLds2=predict(mLds2,subTest2)
pGbm=predict(mGbm,subTest)   ; pGbm2=predict(mGbm2,subTest2)
dat=data.frame(pTree,pTree2,pLds,pLds2,pGbm,pGbm2,y=subTest2$classe)
dat$t1=(pTree==subTest$classe);dat$t2=(pTree2==subTest2$classe);dat$t3=(pLds==subTest$classe)
```

```
dat$t4=(pLds2==subTest2$classe);dat$t5=(pGbm==subTest$classe);  dat$t6=(pGbm2==subTest2$classe)
g1=ggplot(dat, aes(pTree, fill=t1,alpha=0.5))+geom_histogram()+xlab("Prediction:Tree with PCA")
g2=ggplot(dat, aes(pLds,  fill=t3,alpha=0.5))+geom_histogram()+xlab("Prediction:Lds with PCA")
g3=ggplot(dat, aes(pGbm,  fill=t5,alpha=0.5))+geom_histogram()+xlab("Prediction:Gbm with PCA")
g4=ggplot(dat, aes(pTree2,fill=t2,alpha=0.5))+geom_histogram()+xlab("Prediction:Tree without PCA")
g5=ggplot(dat, aes(pLds2, fill=t4,alpha=0.5))+geom_histogram()+xlab("Prediction:Lds without PCA")
g6=ggplot(dat, aes(pGbm2, fill=t6,alpha=0.5))+geom_histogram()+xlab("Prediction:Gbm without PCA")
grid.arrange(g1,g2,g3,g4,g5,g6,ncol=3)
```



Part 3. Prediction on Test Sets

1. Accuracy on SubTrain Sets

```
T11=confusionMatrix(predict(mTree,subTrain), subTrain$classe)$overall['Accuracy'];
T21=confusionMatrix(predict(mTree2,subTrain2),subTrain2$classe)$overall['Accuracy'];
T12=confusionMatrix(predict(mLds,subTrain), subTrain$classe)$overall['Accuracy'];
T22=confusionMatrix(predict(mLds2,subTrain2), subTrain2$classe)$overall['Accuracy'];
T13=confusionMatrix(predict(mGbm,subTrain), subTrain$classe)$overall['Accuracy'];
T23=confusionMatrix(predict(mGbm2,subTrain2), subTrain2$classe)$overall['Accuracy']
Train=round( data.frame(Tree=c(T11,T21),Lds=c(T12,T22),GBM=c(T13,T23)),3)
row.names(Train)=c("PCA","WithoutPCA")
kable(Train)
```

	Tree	Lds	GBM
PCA	0.383	0.526	0.861
WithoutPCA	0.481	0.703	0.975

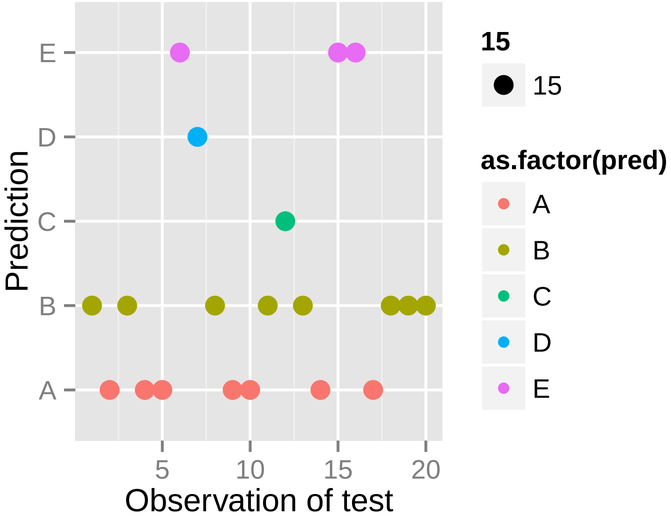
2. Accuracy on SubTest Sets

```
t11=confusionMatrix(predict(mTree,subTest) , subTest$classe)$overall['Accuracy']
t21=confusionMatrix(predict(mTree2,subTest2),subTest2$classe)$overall['Accuracy']
t12=confusionMatrix(predict(mLds,subTest)  , subTest$classe)$overall['Accuracy']
t22=confusionMatrix(predict(mLds2,subTest2) , subTest2$classe)$overall['Accuracy']
t13=confusionMatrix(predict(mGbm,subTest)  , subTest$classe)$overall['Accuracy']
t23=confusionMatrix(predict(mGbm2,subTest2) , subTest2$classe)$overall['Accuracy']
```

```
Test=round( data.frame(Tree=c(t11,t21),Lds=c(t12,t22),GBM=c(t13,t23)) ,3)
row.names(Test)=c("PCA","WithoutPCA")
kable(Test)
```

	Tree	Lds	GBM
PCA	0.380	0.528	0.815
WithoutPCA	0.484	0.703	0.962

```
# This graph show you my classification for the 20 test points.
pred=predict(mGbm2,test)
qplot(c(1:20),pred,col=as.factor(pred),size=15)+xlab('Observation of test')+ylab('Prediction')
```



```
pred
```

```
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

Part 4. Out of Sample Error

In Sameple Errors are errors by applying algorithem model built by subtrain set to itself.

```
In11=1-confusionMatrix(predict(mTree,subTrain), subTrain$classe)$overall['Accuracy'];
In21=1-confusionMatrix(predict(mTree2,subTrain2),subTrain2$classe)$overall['Accuracy'];
In12=1-confusionMatrix(predict(mLds,subTrain), subTrain$classe)$overall['Accuracy'];
In22=1-confusionMatrix(predict(mLds2,subTrain2), subTrain2$classe)$overall['Accuracy'];
In13=1-confusionMatrix(predict(mGbm,subTrain), subTrain$classe)$overall['Accuracy'];
In23=1-confusionMatrix(predict(mGbm2,subTrain2), subTrain2$classe)$overall['Accuracy']
InErr=round( data.frame(Tree=c(In11,In21),Lds=c(In12,In22),GBM=c(In13,In23)),3)
row.names(InErr)=c("PCA","WithoutPCA")
kable(InErr)
```

	Tree	Lds	GBM
PCA	0.617	0.474	0.139
WithoutPCA	0.519	0.297	0.025

Out Sample Errors are errors by applying algorithem model built by subtrain set to subtest set.

```
Out11=1-confusionMatrix(predict(mTree,subTest) , subTest$classe)$overall['Accuracy']
Out21=1-confusionMatrix(predict(mTree2,subTest2),subTest2$classe)$overall['Accuracy']
Out12=1-confusionMatrix(predict(mLds,subTest)  , subTest$classe)$overall['Accuracy']
Out22=1-confusionMatrix(predict(mLds2,subTest2) , subTest2$classe)$overall['Accuracy']
Out13=1-confusionMatrix(predict(mGbm,subTest)  , subTest$classe)$overall['Accuracy']
Out23=1-confusionMatrix(predict(mGbm2,subTest2) , subTest2$classe)$overall['Accuracy']
OutErr=round( data.frame(Tree=c(Out11,Out21),Lds=c(Out12,Out22),GBM=c(Out13,Out23)) ,3)
row.names(OutErr)=c("PCA","WithoutPCA")
kable(OutErr)
```

	Tree	Lds	GBM
PCA	0.620	0.472	0.185
WithoutPCA	0.516	0.297	0.038