```
#Linear Regression Method 1
library(mlbench)
data("BostonHousing")
fit=lm(medv~., BostonHousing)
print(fit)
predictions=predict(fit, BostonHousing)
mse=mean((BostonHousing$medv-predictions)^2)
print(mse)

#Linear Regression Method 2
(caret)library
(mlbench)
data("BostonHousing")
set.seed(7)
trainControl=trainControl(method="cv", number=5)
fit.lm=train(medv~., data=BostonHousing, method="lm",
metric="RMSE",preProcess=c("center","scale"),
trainControl=trainControl)
print(fit.lm)

#Logistic regression Method 1
library(mlbench)
data("PimaIndiansDiabetes")
fit=glm(diabetes~., data=PimaIndiansDiabetes,
family=binomial(link='logit'))
print(fit)
probabilities=predict(fit, PimaIndiansDiabetes[,1:8], type='response')
predictions=ifelse(probabilities>0.5, 'pos','neg')
table(predictions, PimaIndiansDiabetes$diabetes)

#Logistic regression Method 2
library(caret)
library(mlbench)
data("PimaIndiansDiabetes")
set.seed(7)
trainControl=trainControl(method="cv", number=5)
fit.glm=train(diabetes~., data=PimaIndiansDiabetes,
method="glm",metric="Accuracy",preProcess=c("center","scale"),
trControl=trainControl)
print(fit.glm)

#Linear Discriminant Analysis Method 1
library(MASS)
library(mlbench)
data("PimaIndiansDiabetes")
fit=lda(diabetes~., data=PimaIndiansDiabetes)
print(fit)
predictions=predict(fit, PimaIndiansDiabetes[,1:8])$class
table(predictions,PimaIndiansDiabetes$diabetes)
```

```
#Linear Discriminant Analysis Method 2
library(caret)
library(mlbench)
data("PimaIndiansDiabetes")
set.seed(7)
trainControl=trainControl(method="cv",number=5)fit.lda=train(diabetes~
., data=PimaIndiansDiabetes, method="lda", metric="Acurracy",
preProcess=c("center","scale"), trControl=trainControl)
print(fit.lda)

#Regularized Regression Method 1
library(glmnet)
library(mlbench)
data("PimaIndiansDiabetes")
x=as.matrix(PimaIndiansDiabetes[,1:8])
y=as.matrix(PimaIndiansDiabetes[,9])
fit=glmnet(x,y,family="binomial", alpha = 0.5, lambda = 0.001)
print(fit)
predictions=predict(fit,x,type="class")
table(predictions, PimaIndiansDiabetes$diabetes)

#Regularized Regression Method 2
library(glmnet)
library(mlbench)
data("BostonHousing")
BostonHousing$chas=as.numeric(as.character(BostonHousing$chas))
x=as.matrix(BostonHousing[,1:13])
y=as.matrix(BostonHousing[,14])
fit=glmnet(x,y,family="gaussian", alpha=0.5, lambda=0.001)
print(fit)
predictions=predict(fit,x,type="link")
mse=mean((y-predictions)^2)
print(mse)

#Regularized Regression Method 3
library(caret)
library(mlbench)
library(glmnet)
data("PimaIndiansDiabetes")
set.seed(7)
trainControl=trainControl(method="cv", number=5)
fit.glmnet=train(diabetes~., data=PimaIndiansDiabetes,
method="glmnet",metric="Accuracy",
preProcess=c("center","scale"),trControl=trainControl)
print(fit.glmnet)

#Regularized Regression Method 4
library(caret)
library(mlbench)
library(glmnet)
```

```
data("BostonHousing")
set.seed(7)
trainControl=trainControl(method="cv",number=5)fit.glmnet=train(medv~.
, data=BostonHousing, method="glmnet", metric="RMSE",
preProcess=c("center","scale"), trControl=trainControl)
print(fit.glmnet)

#K-Nearest Neighbors Method 1
library(caret)
library(mlbench)
data("PimaIndiansDiabetes")
fit=knn3(diabetes~., data=PimaIndiansDiabetes, k=3)
print(fit)
predictions=predict(fit, PimaIndiansDiabetes[,1:8], type="class")
table(predictions, PimaIndiansDiabetes$diabetes)

#K-Nearest Neighbors Method 2
library(caret)
library(mlbench)
data("BostonHousing")
Boston$chas=as.numeric(as.character(Boston$chas))
x=as.matrix(BostonHousing[,1:13])
y=as.matrix(BostonHousing[,14])
fit=knnreg(x,y,k=3)
print(fit)
predictions=predict(fit,x)
mse=mean(BostonHousing$medv-predictions)^2
print(mse)

#K-Nearest Neighbors Method 3
library(caret)
library(mlbench)
data("PimaIndiansDiabetes")
set.seed(7)
trainControl=trainControl(method="cv", number=5)
fit.knn=train(diabetes~., data=PimaIndiansDiabetes, method="knn",
metric="Accuracy", preProcess=c("center","scale"),
trControl=trainControl)
print(fit.knn)

#Naive Bayes Method 1
library(e1071)
library(mlbench)
data("PimaIndiansDiabetes")
fit=naiveBayes(diabetes~., data = PimaIndiansDiabetes)
print(fit)
predictions=predict(fit, PimaIndiansDiabetes[,1:8])
table(predictions, PimaIndiansDiabetes$diabetes)

#Naive Bayes Method 1
```

```
library(caret)
library(mlbench)
data("PimaIndiansDiabetes")
set.seed(7)
trainControl=trainControl(method="cv",number=5)fit.nb=train(diabetes~.
, data=PimaIndiansDiabetes, method="nb",
metric="Accuracy",trControl=trainControl)
print(fit.nb)

#Support Vector Machine Method 1
library(kernlab)
library(mlbench)
data("PimaIndiansDiabetes")
fit=ksvm(diabetes~., data=PimaIndiansDiabetes,
kernel="rbfdot")#Summarize the fitprint(fit)
predictions=predict(fit, PimaIndiansDiabetes[,1:8], type="response")
table(predictions, PimaIndiansDiabetes$diabetes)

#Support Vector Machine Method 2
library(kernlab)
library(mlbench)
data("BostonHousing")
fit=ksvm(medv~., BostonHousing, kernel="rbfdot")
print(fit)
predictions=predict(fit,BostonHousing)
mse=mean(BostonHousing$medv-predictions)^2
print(mse)

#Support Vector Machine Method 3
library(caret)
library(mlbench)
data("PimaIndiansDiabetes")
set.seed(7)
trainControl=trainControl(method="cv",number=5)fit.svmRadia=train(diab
etes~., data=PimaIndiansDiabetes, method="svmRadial",
metric="Accuracy", trControl=trainControl)
print(fit.svmRadia)

#Support Vector Machine Method 4
library(caret)
library(mlbench)
data("BostonHousing")
set.seed(7)
trainControl=trainControl(method="cv",
number=5)fit.svmRadia=train(medv~., data=BostonHousing,
method="svmRadial", metric="RMSE", trControl=trainControl)
print(fit.svmRadia)

#Classification Trees
library(rpart)
library(mlbench)
data(PimaIndiansDiabetes)
fit=rpart(diabetes~., data=PimaIndiansDiabetes)
print(fit)
predictions=predict(fit, PimaIndiansDiabetes[,1:8], type="class")
table(predictions, PimaIndiansDiabetes$diabetes)

#Regression Trees
library(rpart)
library(mlbench)
data(BostonHousing)
fit=rpart(medv~., data=BostonHousing,
control=rpart.control(minsplit=5))
print(fit)
predictions=predict(fit, BostonHousing[,1:13])
mse=mean(BostonHousing$medv-predictions)^2)
print(mse)

#Classification Trees
library(caret)
library(mlbench)
data(PimaIndiansDiabetes)
set.seed(7)
trainControl=trainControl(method="cv",
number=5)fit.rpart=train(diabetes~., data=PimaIndiansDiabetes,
method="rpart", metric="Accuracy", trControl=trainControl)
print(fit.rpart)

#Regression Trees
library(caret)
library(mlbench)
data(BostonHousing)
set.seed(7)
trainControl=trainControl(method="cv",
number=2)fit.rpart=train(medv~., data=BostonHousing, method="rpart",
metric="RMSE", trControl=trainControl)
print(fit.rpart)
```