Dimensionality Reduction

Load in data and packages

```
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
library(class)
library(tree)
library(MASS)

df <- read.csv("~/Desktop/diamonds.csv")
```

Normalize data for linear regression

```
df$carat <- factor(df$carat)
df$cut <- factor(df$cut)
df$color <- factor(df$color)
df$clarity <- factor(df$clarity)
df$depth <- factor(df$depth)
df$price <- factor(df$price)

levels(df$cut) <- c("Fair", "Good", "Very Good", "Premium", "Ideal")
levels(df$color) <- c("D", "E", "F", "G", "H", "I", "J")
levels(df$clarity) <- c("I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "VVS1")
```

Run PCA on the data

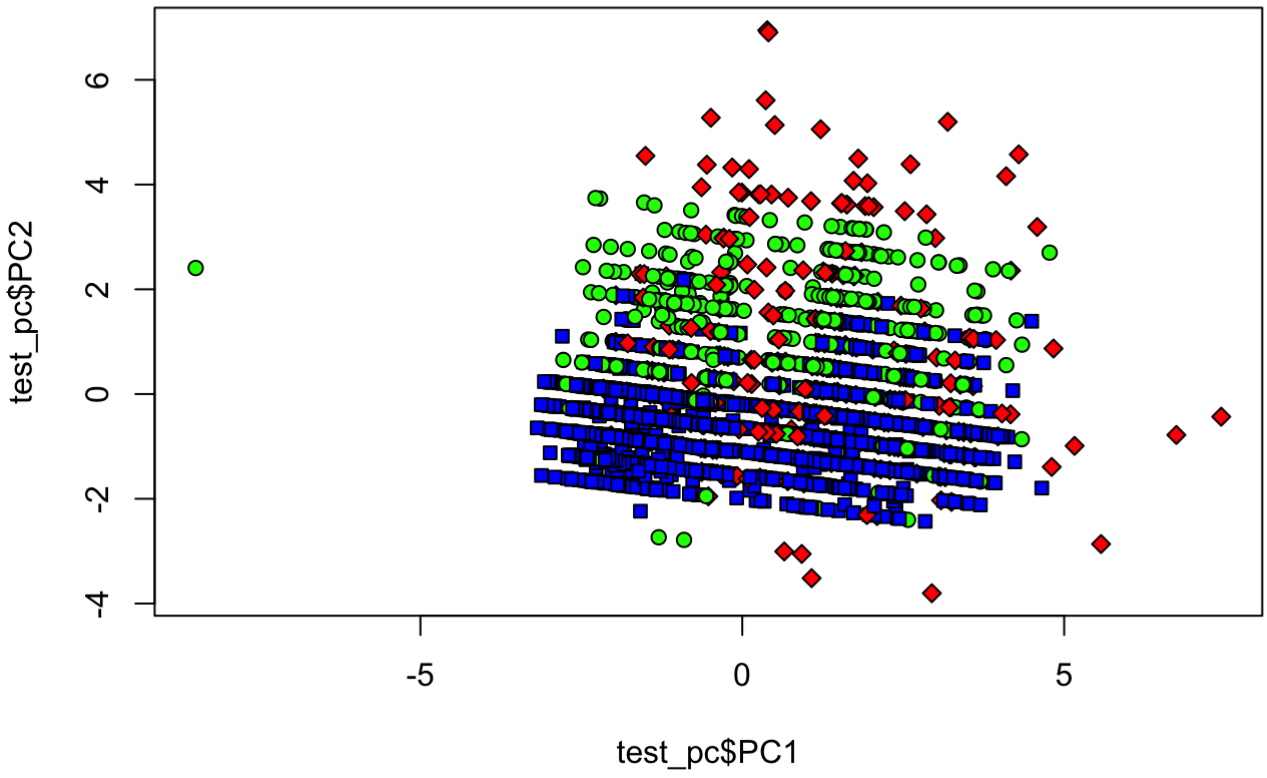
```
set.seed(1234)
i <-sample(1:nrow(df), .80*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

```
pca_out <- preProcess(train[,1:ncol(train)], method=c("center", "scale", "pca"))
pca_out
```

```
## Created from 43152 samples and 11 variables
##
## Pre-processing:
##   - centered (5)
##   - ignored (6)
##   - principal component signal extraction (5)
##   - scaled (5)
##
## PCA needed 3 components to capture 95 percent of the variance
```

Plotting PCA

```
train_pc <- predict(pca_out, train[, 1:ncol(train)])
test_pc <- predict(pca_out, test[,])
plot(test_pc$PC1, test_pc$PC2, pch=c(23,21,22)[unclass(test_pc$cut)], bg=c("red","gree
n","blue")[unclass(test$cut)])
```



PCA data in knn

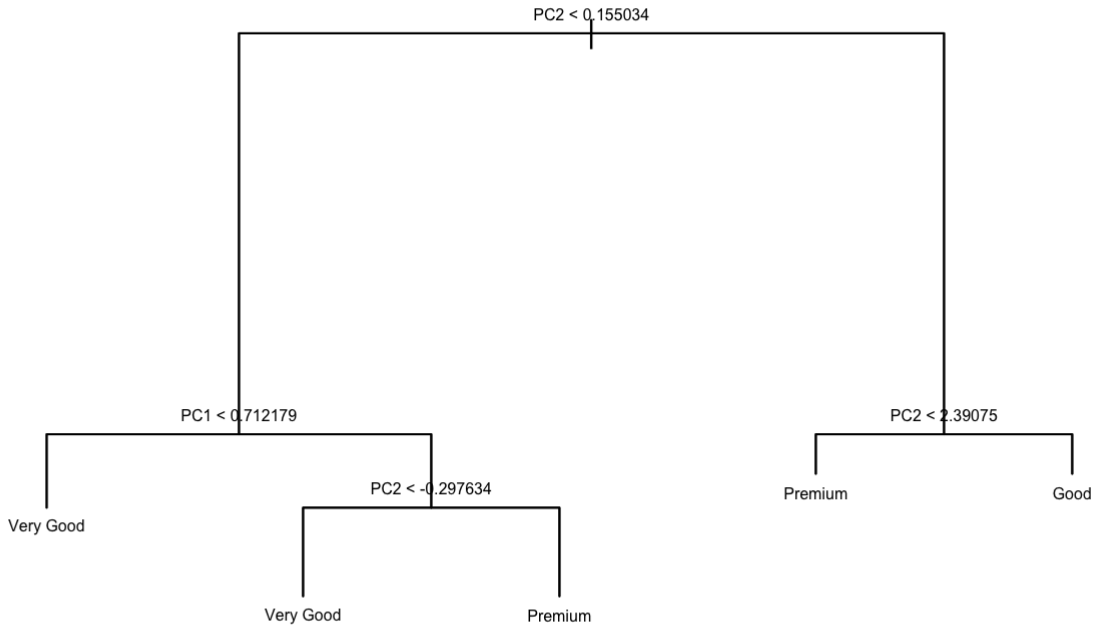
Testing to see if our two PCA components can accurately predict cut from the test data

```
train_df <- data.frame(train_pc$PC1, train_pc$PC2, train$cut)
test_df <- data.frame(test_pc$PC1, test_pc$PC2, test$cut)
set.seed(1234)
pred <- knn(train=train_df[,1:2], test=test_df[,1:2], cl=train_df[,3], k=3)
mean(pred==test$cut)
```

```
## [1] 0.5518168
```

Accuracy is much lower than if we ran it with all our predictors.

```
colnames(train_df) <- c("PC1", "PC2", "Cut")
colnames(test_df) <- c("PC1", "PC2", "Cut")
set.seed(1234)
tree1 <- tree(Cut~., data=train_df)
plot(tree1)
text(tree1, cex=0.5, pretty=0)
```




```
pred <- predict(tree1, newdata=test_df, type="class")
mean(pred==test$cut)
```

```
## [1] 0.6112347
```

With the decision tree we get a little higher accuracy

LDA

```
lda1 <- lda(Cut~., data=train_df)
coef(lda1)
```

```
##            LD1        LD2
## PC1 -0.2649236  0.5162082
## PC2 -1.1491105 -0.3985027
```

```
lda1$means
```

```
##                   PC1        PC2
## Fair       0.95527337  0.6021259
## Good       0.31226850  0.5097401
## Very Good -0.45542449 -0.6145795
## Premium    0.41484896  0.5216808
## Ideal      0.08356481  0.2099710
```

```
lda_pred <- predict(lda1, newdata=train_df, type="class")
mean(lda_pred$class==test$cut)
```

```
## [1] 0.3498563
```

Running Linear Regression on the reduced data

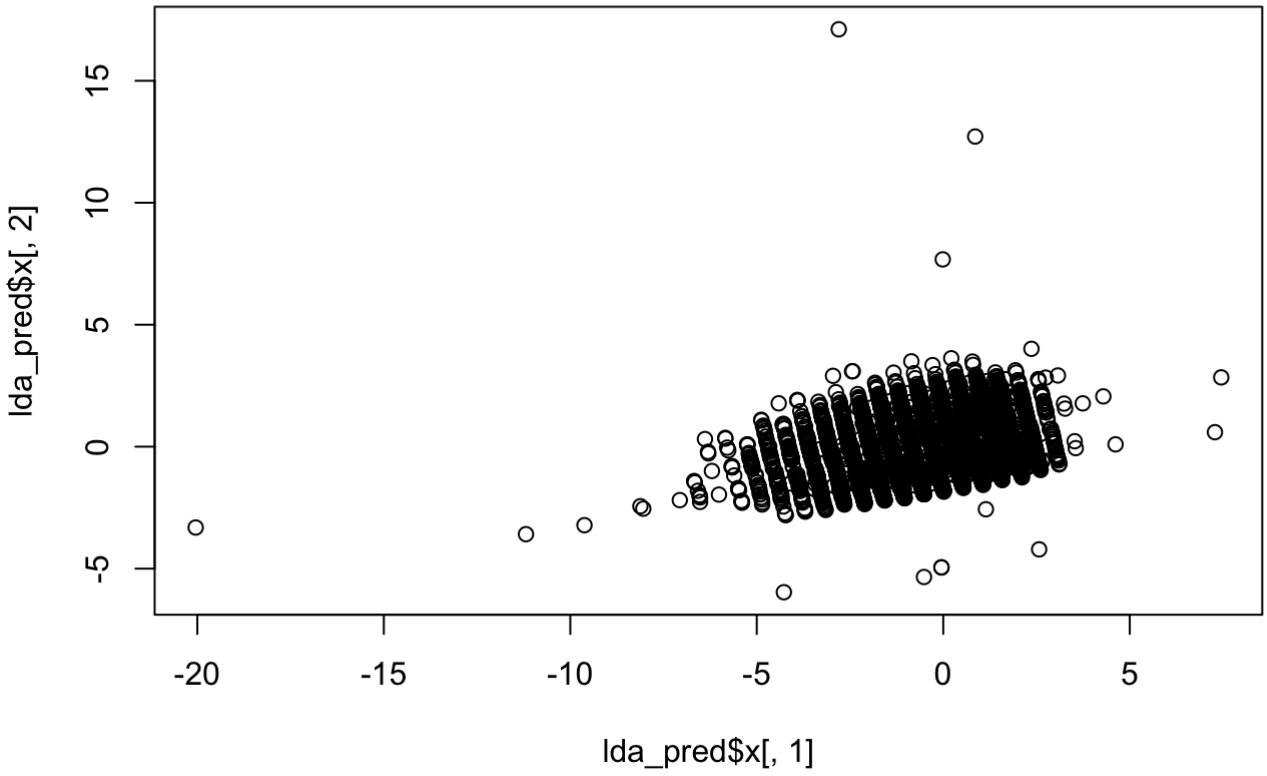
```
glm1 <- glm(Cut~., data=train_df, family=binomial)
summary(glm1)
```

```
##
## Call:
## glm(formula = Cut ~ ., family = binomial, data = train_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5543   0.1581   0.2049   0.2683   2.3558
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.76173    0.03481  108.06   <2e-16 ***
## PC1         -0.29603    0.01596  -18.55   <2e-16 ***
## PC2         -0.52764    0.02454  -21.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11513  on 43151  degrees of freedom
## Residual deviance: 10728  on 43149  degrees of freedom
## AIC: 10734
##
## Number of Fisher Scoring iterations: 7
```

A significant amount of accuracy was lost with dimensional reduction. Usually LDA gives a higher accuracy, so this might there might be unique about the data that really hurts data reduction.

Ploting LDA

```
plot(lda_pred$x[,1], lda_pred$x[,2], pch=c(23,21,22)[unclass(lda_pred$cut)], bg=c("re
d","green","blue")[unclass(test_pc$cut)])
```


lda_pred$x[, 2]
15
10
5
0
-5
-20
-15
-10
-5
0
5
lda_pred$x[, 1]