Dimensionality

Spencer Gray

2022-10-08

Data Setup

Cleaning up our data, removing all NAs and setting to 0 to assist with our kNN model in the future.

```
library(caret)
```

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

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

```
aus <- read.csv("weatherAUS.csv")

aus <- subset(aus,RainTomorrow  != "NA")

for(i in c(3, 4, 5, 9, 12, 13, 14, 15, 16, 17, 20, 21))
{
  aus[is.na(aus[,i]), i] <- mean(aus[,i], na.rm = TRUE)
}


dim(aus)
```

```
## [1] 142193     23
```

```
head(aus)
```

```
##         Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir
## 1 2008-12-01   Albury    13.4    22.9      0.6          NA       NA           W
## 2 2008-12-02   Albury     7.4    25.1      0.0          NA       NA         WNW
## 3 2008-12-03   Albury    12.9    25.7      0.0          NA       NA         WSW
## 4 2008-12-04   Albury     9.2    28.0      0.0          NA       NA          NE
## 5 2008-12-05   Albury    17.5    32.3      1.0          NA       NA           W
## 6 2008-12-06   Albury    14.6    29.7      0.2          NA       NA         WNW
##   WindGustSpeed WindDir9am WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am
## 1            44          W        WNW           20           24          71
## 2            44        NNW        WSW            4           22          44
## 3            46          W        WSW           19           26          38
## 4            24         SE          E           11            9          45
## 5            41        ENE         NW            7           20          82
## 6            56          W          W           19           24          55
##   Humidity3pm Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm
## 1          22      1007.7      1007.1        8       NA    16.9    21.8
## 2          25      1010.6      1007.8       NA       NA    17.2    24.3
## 3          30      1007.6      1008.7       NA        2    21.0    23.2
## 4          16      1017.6      1012.8       NA       NA    18.1    26.5
## 5          33      1010.8      1006.0        7        8    17.8    29.7
## 6          23      1009.2      1005.4       NA       NA    20.6    28.9
##   RainToday RainTomorrow
## 1        No           No
## 2        No           No
## 3        No           No
## 4        No           No
## 5        No           No
## 6        No           No
```

```
i <- sample(1:nrow(aus), 0.8 * nrow(aus), replace = FALSE)

train <- aus[i,]
test <- aus[-i,]
```

Data Representation

Selecting relatively stable and numerically recorded variables (quantitative) to use our PCA model on. Predicting rain tomorrow in Column 23. MinTemp -> column 3 MaxTemp -> column 4 Rainfall -> column 5 WindGustSpeed -> column 9 WinSpeed9am -> column 12 WinSpeed3pm -> column 13 Humidity9am -> column 14 Humidity3pm -> column 15 Pressure9am -> column 16 Pressure3pm -> column 17 Temp9am -> column 20 Temp3pm -> column 21

```
set.seed(1234)
pcaModel <- preProcess(train[,c(3, 4, 5, 9, 12, 13, 14, 15, 16, 17, 20, 21, 23)], method = c("center", "scale", "pca"))
pcaModel
```

```
## Created from 113754 samples and 13 variables
## 
## Pre-processing:
##   - centered (12)
##   - ignored (1)
##   - principal component signal extraction (12)
##   - scaled (12)
## 
## PCA needed 7 components to capture 95 percent of the variance
```

PCA Model Setup

```
trainPCA <- predict(pcaModel, train[, c(3, 4, 5, 9, 12, 13, 14, 15, 16, 17, 20, 21, 23)])
testPCA <- predict(pcaModel,  test[, c(3, 4, 5, 9, 12, 13, 14, 15, 16, 17, 20, 21, 23)])
```

PCA Model Accuracy

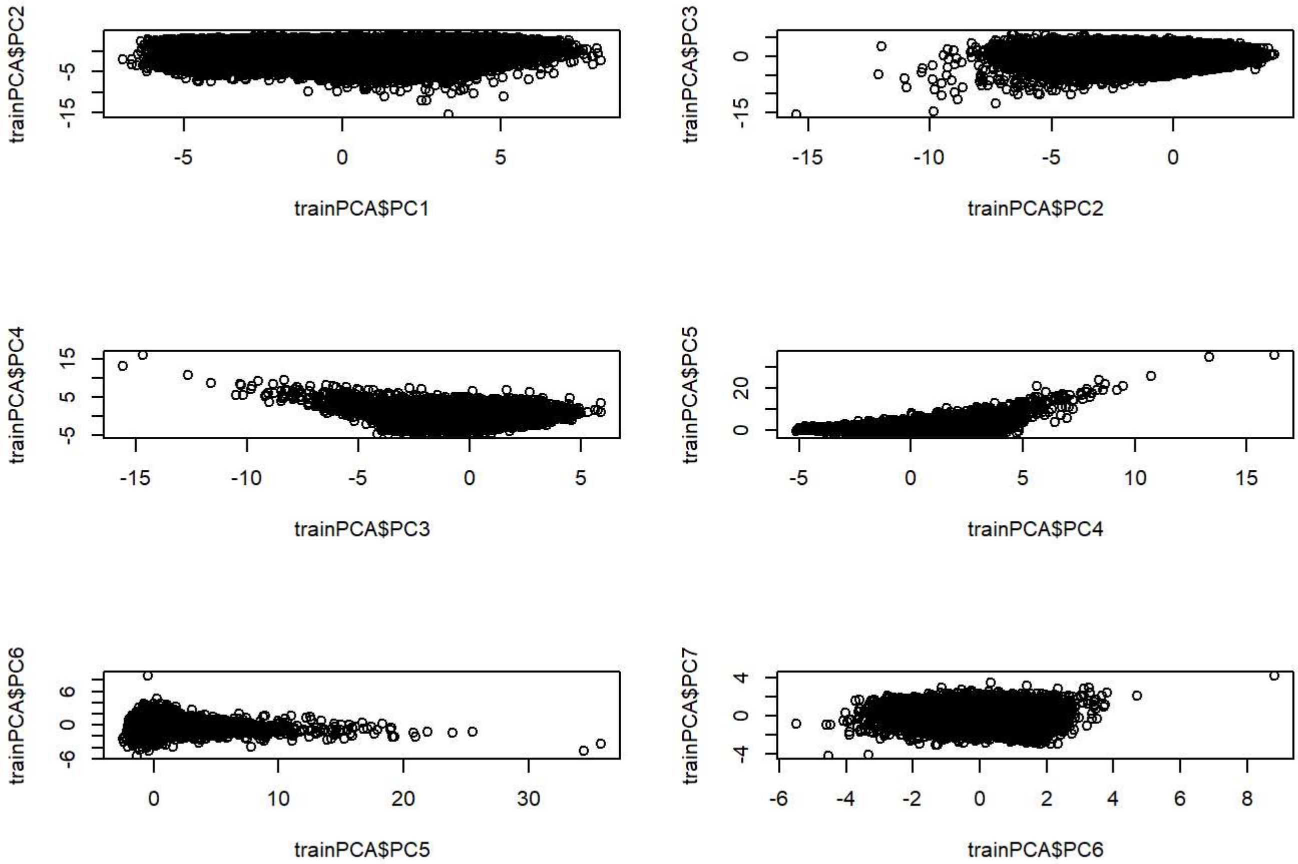
Graphing data of different principal components against one another. Training a knn model with PCA predict rain tomorrow based on variants of specific weather characteristics including: temperature, humidity, and wind.

```
library(class)
set.seed(1234)

trainDF <- data.frame(trainPCA$PC1, trainPCA$PC2, trainPCA$PC3, trainPCA$PC4, trainPCA$PC5, trainPCA$PC6, trainPCA$PC7, train$RainTomorrow)
testDF <- data.frame(testPCA$PC1, testPCA$PC2, testPCA$PC3, testPCA$PC4, testPCA$PC5, testPCA$PC6, testPCA$PC7, test$RainTomorrow)



par(mfrow=c(3,2))
plot(trainPCA$PC1, trainPCA$PC2)
plot(trainPCA$PC2, trainPCA$PC3)
plot(trainPCA$PC3, trainPCA$PC4)
plot(trainPCA$PC4, trainPCA$PC5)
plot(trainPCA$PC5, trainPCA$PC6)
plot(trainPCA$PC6, trainPCA$PC7)
```



```
start_time <- Sys.time()
pred_reduced <- knn(trainDF[,1:7], testDF[,1:7], trainDF[,8], k = 6)

mean(pred_reduced == test$RainTomorrow)
```

```
## [1] 0.8233763
```

Regular Model Accuracy

Training a kNN model based on all the previous parameters but unmodified.

```
library(class)
set.seed(1234)

trainDF_real <- data.frame(train$MinTemp, train$MaxTemp, train$WindGustSpeed, train$WindSpeed9am, train$WindSpeed3pm, train$Humidity9am, train$Humidity3pm, train$Pressure9am, train$Pressure3pm, train$Temp9am, train$Temp3pm, train$Rainfall, train$RainTomorrow)

testDF_real <- data.frame(test$MinTemp, test$MaxTemp, test$WindGustSpeed, test$WindSpeed9am, test$WindSpeed3pm, test$Humidity9am, test$Humidity3pm, test$Pressure9am, test$Pressure3pm, test$Temp9am, test$Temp3pm, test$Rainfall, test$RainTomorrow)


pred <- knn(trainDF_real[,1:12], testDF_real[,1:12], trainDF_real[,13], k = 10)

mean(pred==test$RainTomorrow)
```

```
## [1] 0.8407469
```

LDA Model Setup and Accuracy

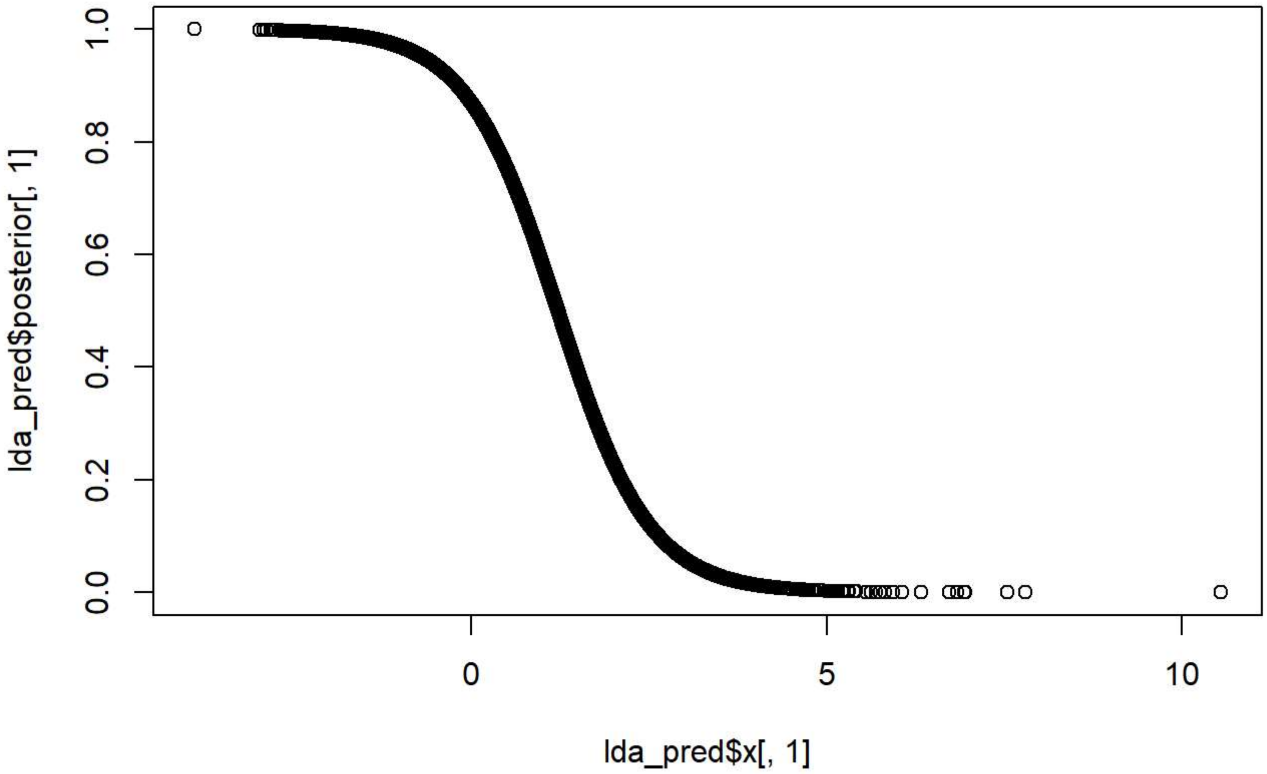
```
library(MASS)
ldaModel <- lda(RainTomorrow~MinTemp + MaxTemp + WindGustSpeed + WindSpeed9am + WindSpeed3pm + Humidity9am + Humidity3pm + Pressure9am + Pressure3pm + Temp9am + Temp3pm + Rainfall, data = train)
ldaModel$means
```

```
##      MinTemp  MaxTemp WindGustSpeed WindSpeed9am WindSpeed3pm Humidity9am
## No  11.88450 23.83419      38.37822     13.56496     18.24297    66.27050
## Yes 13.19607 21.12888      45.44631     15.48172     20.02059    77.84037
##     Humidity3pm Pressure9am Pressure3pm  Temp9am  Temp3pm Rainfall
## No     46.62905    1018.522    1016.046 17.06625 22.38704 1.270456
## Yes    68.31843    1014.713    1012.594 16.69033 19.26103 6.113548
```

```
lda_pred <- predict(ldaModel, newdata=test, type="class")
mean(lda_pred$class==test$RainTomorrow)
```

```
## [1] 0.8395513
```

```
plot(lda_pred$x[,1], lda_pred$posterior[,1])
```


1.0
0.8
0.6
0.4
0.2
0.0
lda_pred$posterior[, 1]
0
5
10
lda_pred$x[, 1]