R Notebook

Name: Gabriel Bentley

Date: 10/06/22

Dataset: Gas Turbine Metric Measurements

https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set

Data set information

The dataset contains 14795 instances of 11 sensor measures aggregated over one hour (by means of average or sum) from a gas turbine located in Turkey's north western region for the purpose of studying flue gas emissions, namely CO and NOx (NO + NO2). This data is collected in another data range (01.01.2011 - 31.12.2011) and (01.01.2015 - 31.12.2015), includes gas turbine parameters (such as Turbine Inlet Temperature and Compressor Discharge pressure) in addition to the ambient variables.

Attribute information

The explanations of sensor measurements and their brief statistics are given below.

Variable (Abbr.) Unit Min Max Mean Ambient temperature (AT) C â€“6.23 37.10 17.71 Ambient pressure (AP) mbar 985.85 1036.56 1013.07 Ambient humidity (AH) (%) 24.08 100.20 77.87 Air filter difference pressure (AFDP) mbar 2.09 7.61 3.93 Gas turbine exhaust pressure (GTEP) mbar 17.70 40.72 25.56 Turbine inlet temperature (TIT) C 1000.85 1100.89 1081.43 Turbine after temperature (TAT) C 511.04 550.61 546.16 Compressor discharge pressure (CDP) mbar 9.85 15.16 12.06 Turbine energy yield (TEY) MWH 100.02 179.50 133.51 Carbon monoxide (CO) mg/m3 0.00 44.10 2.37 Nitrogen oxides (NOx) mg/m3 25.90 119.91 65.29

Read in and seperate the data set into train and test

```
df <- read.csv("gt_2011.csv")

df$year <- rep(c("2011"), each = 7411)

colSums(is.na(df))
```

```
##   AT   AP   AH AFDP GTEP  TIT  TAT  TEY  CDP   CO  NOX year
##    0    0    0    0    0    0    0    0    0    0    0    0
```

```
df2 <- read.csv("gt_2015.csv")
df2$year <- rep(c("2015"), each = 7384)

colSums(is.na(df2))
```

```
##   AT   AP   AH AFDP GTEP  TIT  TAT  TEY  CDP   CO  NOX year
##    0    0    0    0    0    0    0    0    0    0    0    0
```

```
df3 <- rbind(df, df2)

df3$year <- factor(df3$year)
```

```
set.seed(1017)
i <- sample(1:nrow(df3), nrow(df3)*0.80, replace=FALSE)
train <- df3[i,]
test <- df3[-i,]
```

Explore the data set

```
str(train)
```

```
## 'data.frame':    11836 obs. of  12 variables:
##  $ AT  : num  14.9 21 23.3 27.9 18.4 ...
##  $ AP  : num  1011 1010 1016 1011 1004 ...
##  $ AH  : num  86.8 72 84.7 54.8 82.5 ...
##  $ AFDP: num  4.47 4.33 3.4 4 3.12 ...
##  $ GTEP: num  31.6 31.1 21.4 26.1 21.8 ...
##  $ TIT : num  1100 1100 1068 1094 1062 ...
##  $ TAT : num  533 541 550 551 550 ...
##  $ TEY : num  157 153 117 134 120 ...
##  $ CDP : num  13.7 13.4 11.1 12.4 11 ...
##  $ CO  : num  0.774 1.784 1.52 0.107 2.941 ...
##  $ NOX : num  60.5 50.5 52.3 64.5 51 ...
##  $ year: Factor w/ 2 levels "2011","2015": 1 2 1 1 2 2 2 1 1 2 ...
```

```
summary(train)
```

```
##        AT                AP                AH               AFDP
##  Min.   :-6.235   Min.   : 989.4   Min.   : 24.09   Min.   :2.369
##  1st Qu.:10.970   1st Qu.:1009.8   1st Qu.: 63.94   1st Qu.:3.289
##  Median :16.918   Median :1013.8   Median : 76.17   Median :3.856
##  Mean   :17.145   Mean   :1014.4   Mean   : 73.97   Mean   :3.843
##  3rd Qu.:23.506   3rd Qu.:1018.3   3rd Qu.: 85.20   3rd Qu.:4.322
##  Max.   :37.103   Max.   :1036.6   Max.   :100.17   Max.   :7.319
##       GTEP            TIT             TAT              TEY
##  Min.   :17.70   Min.   :1001   Min.   :512.5   Min.   :100.0
##  1st Qu.:23.20   1st Qu.:1073   1st Qu.:543.1   1st Qu.:127.6
##  Median :25.04   Median :1086   Median :549.8   Median :133.8
##  Mean   :25.87   Mean   :1082   Mean   :545.6   Mean   :134.8
##  3rd Qu.:29.96   3rd Qu.:1100   3rd Qu.:550.0   3rd Qu.:147.4
##  Max.   :39.37   Max.   :1101   Max.   :550.6   Max.   :179.5
##       CDP               CO                NOX           year
##  Min.   : 9.871   Min.   : 0.00039   Min.   : 25.91   2011:5910
##  1st Qu.:11.532   1st Qu.: 1.09575   1st Qu.: 55.22   2015:5926
##  Median :11.977   Median : 1.79110   Median : 61.77
##  Mean   :12.146   Mean   : 2.37158   Mean   : 63.81
##  3rd Qu.:13.172   3rd Qu.: 2.98528   3rd Qu.: 70.47
##  Max.   :15.159   Max.   :43.62200   Max.   :119.68
```

```
head(train)
```

```
##           AT     AP     AH   AFDP   GTEP    TIT    TAT    TEY    CDP      CO
## 2338  14.859 1010.6 86.796 4.4678 31.576 1100.0 533.21 157.47 13.727 0.77443
## 11208 20.993 1010.3 71.966 4.3313 31.066 1099.9 541.23 152.56 13.447 1.78360
## 5358  23.331 1016.2 84.739 3.4000 21.358 1067.8 549.85 116.56 11.092 1.51980
## 5287  27.939 1010.9 54.820 4.0037 26.080 1093.8 550.57 133.73 12.377 0.10666
## 10133 18.357 1004.4 82.508 3.1160 21.828 1062.2 550.21 119.84 11.015 2.94110
```

```
## 13160 18.038 1012.3 75.757 3.8826 24.577 1080.5 549.70 132.11 11.918 3.26400
##          NOX year
## 2338  60.476 2011
## 11208 50.520 2015
## 5358  52.319 2011
## 5287  64.510 2011
## 10133 51.000 2015
## 13160 48.010 2015
```

```
names(train)
```

```
##  [1] "AT"   "AP"   "AH"   "AFDP" "GTEP" "TIT"  "TAT"  "TEY"  "CDP"  "CO"
## [11] "NOX"  "year"
```

```
colSums(is.na(train))
```

```
##   AT   AP   AH AFDP GTEP  TIT  TAT  TEY  CDP   CO  NOX year
##    0    0    0    0    0    0    0    0    0    0    0    0
```

Create PCA df

```
library(caret)
```

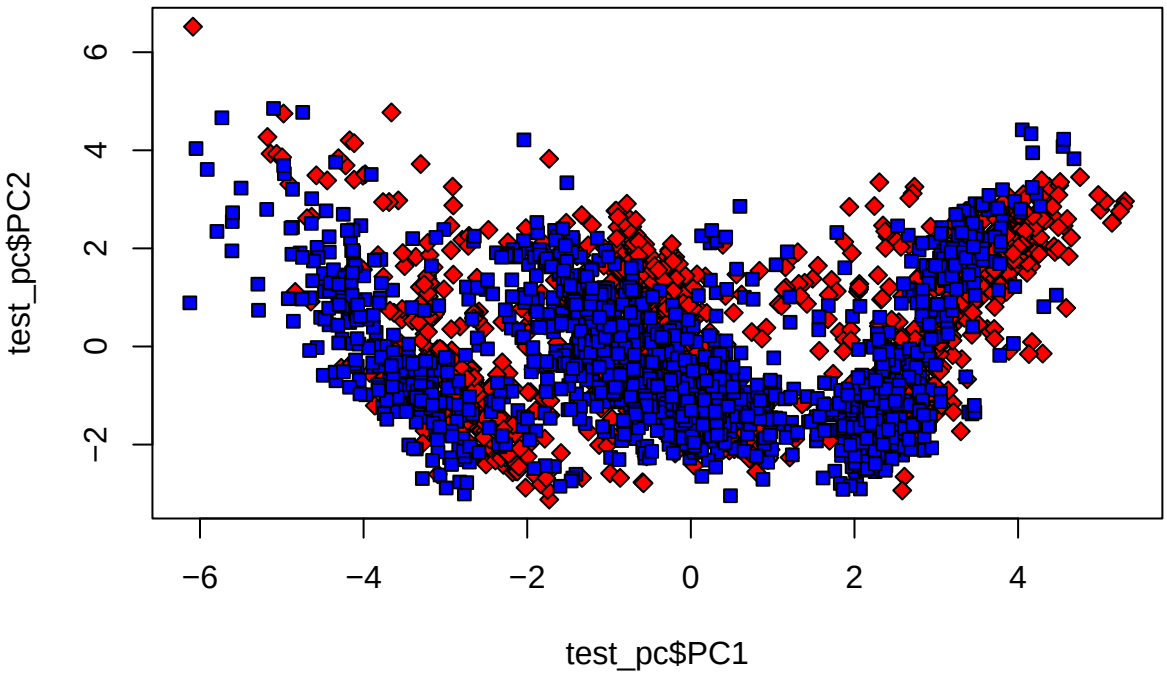
```
## Loading required package: ggplot2
```

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

```
pca_out <- preProcess(train[,1:11], method=c("center","scale","pca"))
pca_out
```

```
## Created from 11836 samples and 11 variables
##
## Pre-processing:
##   - centered (11)
##   - ignored (0)
##   - principal component signal extraction (11)
##   - scaled (11)
##
## PCA needed 6 components to capture 95 percent of the variance
```

```
train_pc <- predict(pca_out, train[,])
test_pc <- predict(pca_out , test[,])
plot(test_pc$PC1, test_pc$PC2, pch=c(23,22)[unclass(test_pc$year)], bg=c("red","blue")[unclass(test$yea
```



PCA Logistic Regression

```
glm1 <- glm(year~., data=train_pc, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm1)
```

```
##
## Call:
## glm(formula = year ~ ., family = "binomial", data = train_pc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.3643  -0.2319   0.0004   0.1626   4.5119
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20254    0.04752   4.262 2.03e-05 ***
## PC1         -0.13002    0.01635  -7.953 1.82e-15 ***
## PC2         -0.64211    0.02939 -21.848  < 2e-16 ***
## PC3          2.85620    0.06431  44.414  < 2e-16 ***
## PC4         -0.08074    0.05565  -1.451    0.147
## PC5          2.14436    0.06733  31.849  < 2e-16 ***
## PC6          5.99326    0.13810  43.399  < 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: 16408.2  on 11835  degrees of freedom
## Residual deviance:  4605.2  on 11829  degrees of freedom
```

```
## AIC: 4619.2
##
## Number of Fisher Scoring iterations: 7
```

```
probsLR <- predict(glm1, newdata=test_pc, type="response")
predLR <- ifelse(probsLR>0.5, 2015, 2011)
accLR <- mean(predLR == test_pc$year)

cat("accuracy: ", accLR)
```

```
## accuracy:  0.929706
```

```
table(predLR, test_pc$year)
```

```
##
## predLR 2011 2015
##   2011 1401  108
##   2015  100 1350
```

```
confusionMatrix(as.factor(predLR), reference = test_pc$year)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 2011 2015
##       2011 1401  108
##       2015  100 1350
##
##                Accuracy : 0.9297
##                  95% CI : (0.9199, 0.9387)
##     No Information Rate : 0.5073
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8594
##
##  Mcnemar's Test P-Value : 0.6274
##
##             Sensitivity : 0.9334
##             Specificity : 0.9259
##          Pos Pred Value : 0.9284
##          Neg Pred Value : 0.9310
##              Prevalence : 0.5073
##          Detection Rate : 0.4735
##    Detection Prevalence : 0.5100
##       Balanced Accuracy : 0.9297
##
##        'Positive' Class : 2011
##
```

Create LDA

```
library(MASS)
lda1 <- lda(year~., data=train)
lda1$means
```

```
##            AT       AP       AH     AFDP     GTEP      TIT      TAT      TEY
```

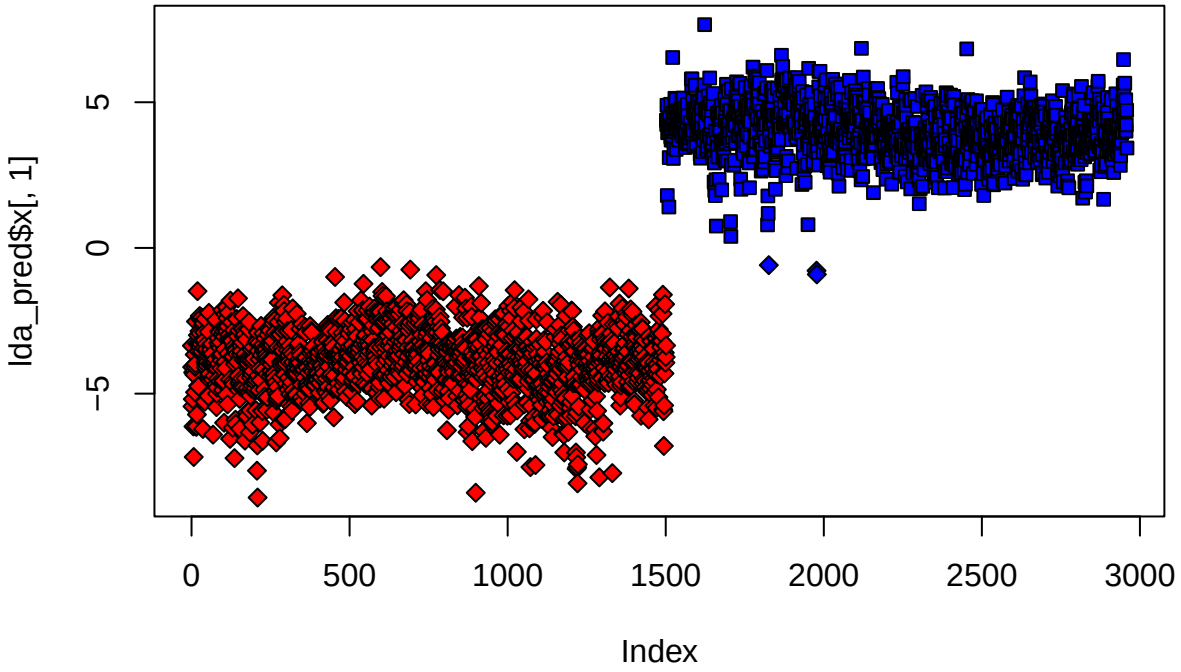
```
## 2011 17.06376 1014.227 79.30615 4.089230 25.62738 1084.655 544.5747 135.6416
## 2015 17.22554 1014.505 68.65363 3.597972 26.10829 1078.856 546.6143 133.9205
##           CDP       CO      NOX
## 2011 12.19852 1.587241 67.64966
## 2015 12.09338 3.153810 59.97648
```

LAC Classification

```
lda_pred <- predict(lda1, newdata=test, type="class")

LAC_acc <- mean(lda_pred$class==test$year)

plot(lda_pred$x[,1], pch=c(23,22)[unclass(lda_pred$class)], bg=c("red","blue")[unclass(test_pc$year)])
```




```
cat("\nAccuracy: ", LAC_acc)
```

```
##
## Accuracy:  0.9989861
```

Analysis

When using the PCA method of dimensional reduction for the data set there was a loss of accuracy for logistic regression when compared to logistic regression without PCA. PCA logistic regression had an accuracy of only 92% when predicting the year of the data point while logistic regression without PCA had an accuracy of around 98%.

When using LDA method of dimensional reduction for the data set there was an increase in total accuracy for classification. The classification done when using LDA had an accuracy of 99% higher than any of the other classification methods used previously.