# Analysis of California Grid Emergencies

Michaela Bodie, Chad Kite

San Francisco State University

MATH 448 - Introduction to Statistical Learning and Data Mining

**CONTENTS**

## 1. Abstract

The California Independent System Operator (CAISO) grid is an intricate yet forceful electricity transmission system that is the foundation of the electrical infrastructure in California. The CAISO grid is one of the most complex and advanced grid systems in the world, and plays a significant role in guaranteeing that clients across California receive efficient transmission of electricity. To continue being a catalyst for efficient energy, California strives to achieve a 60% renewable generation portfolio by 2030. However, no matter how consistent the electric frequency of the grid is, power shortages and shut downs still occur. Understanding the connection between different methods of power generation and causation of grid emergencies, calls for an analysis of the energy produced by each specific resource type: Renewables, Nuclear, Thermal, Hydro, and Imports.

The CAISO website maintains hourly records of power generated by each resource category. The data was captured in daily updates that cover the time period between June 1, 2018 to November 30, 2021. Each of the updates are in text format and cover the electricity output over the 24 hour period of the day recorded. In addition, there are tables that contain all of the grid emergencies recorded from 1998 to 2022. Each record contains the date, time, reason for the message, region, and duration of the impacts to the grid. A grid emergency occurring was classified as TRUE, while no grid emergency was classified as FALSE.

This paper delves into the analysis of power generation revolving around various energy sources and trends, moreover discusses the various statistical analysis techniques utilized to analyze the power generation dynamics.

## 2. Introduction

There is no clear understanding of how changes in power generation relate to shortages in power availability. We want to analyze which renewable resources play a significant role in the cause of a power outage, as well as use this information to be able to determine when a power outage will occur. Overall, we will acknowledge the significance of analyzing the differing trends in power generation. The resources we will be analyzing include Renewables, Nuclear, Thermal, Imports, and Hydro. While conducting our analysis, we will also take into account the various segments of the day that utilize different resources during each of these times. Our objective for this project is inference, as we are exploring what proportion of grid emergencies are correlated with each resource.

## 3. The Data

The California Independent System Operator (CISO) maintains records of the amount of power generated by resource category on an hourly basis. This data is captured in the form of daily updates that are available on the CISO website for the period of June 1, 2018 to November 30, 2021, except for a 12 day gap from September 23, 2019 to October 3, 2019. The updates are in text format and each covers a 24 hour period. Combining the files results in a total of 28,248 records of power generated by resource category during the specified hour and day.

CISO has also published a report on its website that contains tables of all grid emergencies from 1998 to 2022. The report is in pdf format. For the period from June 1, 2018 to November 30, 2021, there were 118 alerts, warnings and emergencies with sufficient information to include in this analysis. Each record contains the date, time, and reason for the message, as well as the region and duration of the anticipated impacts to the grid.

## 4. Model Selection

### a. Unsupervised Methods

Unsupervised methods were utilized to provide insight into the relationships between the power produced by each resource type.

#### i. Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique used for data dimensionality reduction and feature extraction. It aims to retain as much information and variability within the dataset as possible while reducing the number of variables. PCA achieves this by identifying the directions, known as principal components, along which the data varies the most. In our analysis, the summary indexes were computed as follows:

| | PC1 | PC2 | PC3 | PC4 | PC5 |
|---|---|---|---|---|---|
| RENEWABLES | 0.76 | 0.50 | 0.39 | 0.16 | -0.03 |
| NUCLEAR | 0.01 | -0.01 | 0.13 | -0.14 | 0.98 |
| THERMAL | -0.51 | 0.85 | -0.10 | 0.04 | 0.04 |
| IMPORTS | -0.41 | -0.16 | 0.82 | 0.36 | -0.06 |
| HYDRO | -0.05 | 0.07 | 0.37 | -0.91 | -0.18 |

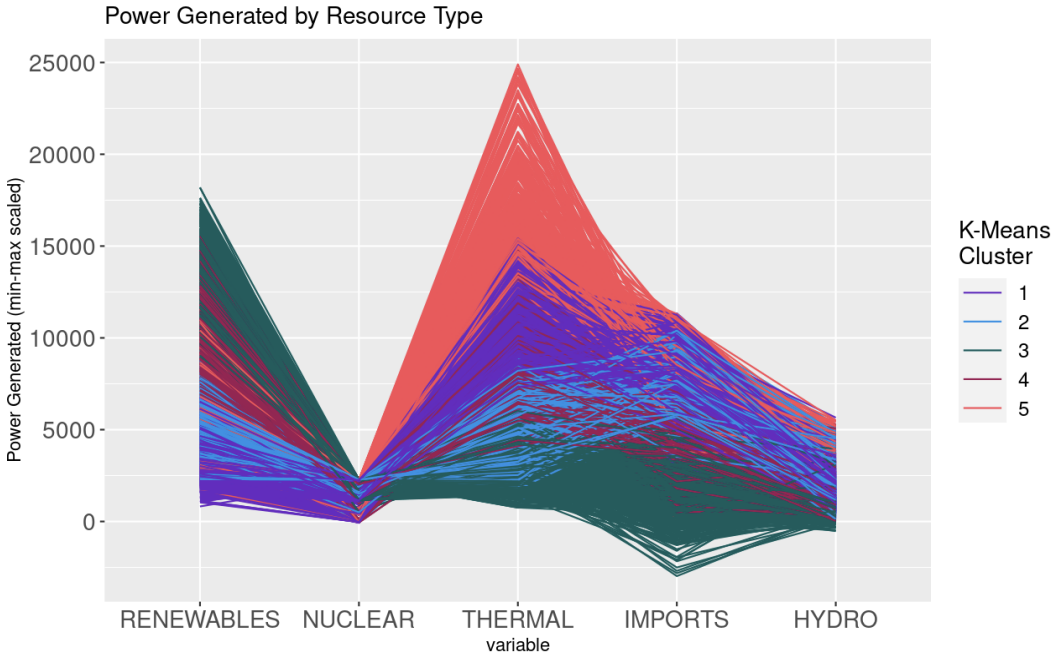
The first principal component, PC1 is the combination of variables that best approximates the power generation from each resource. From there, PC2 is calculated to reflect the second largest source of variation in the data with regard to PC1. The remaining principal components are calculated as so.

| | PC1 | PC2 | PC3 | PC4 | PC5 |
|---|---|---|---|---|---|
| Standard deviation | 5118.75 | 3236.67 | 1518.31 | 1092.05 | 511.59 |
| Proportion of Variance | 0.65 | 0.26 | 0.06 | 0.03 | 0.01 |
| Cumulative Proportion | 0.65 | 0.91 | 0.96 | 0.99 | 1.00 |

As seen in the summary PCA statistics in the image above, PC1 accounts for 65% of the variation within the dataset.

### ii. K-Means Clustering

K-means clustering is used to partition a dataset into distinct groups or clusters based on their similarities. The goal of K-means is to divide the data points into K clusters, where K is a defined parameter. In our analysis, the data was divided into K = 5 clusters:
Low Renewables/Medium Thermal/Medium Hydro, Medium-Low Renewables/Medium-Low Thermal/High Imports, High Renewables/Low Thermal/Low Imports, Medium Renewables/Medium Thermal/Medium Imports, and High Thermal/High Imports/High Hydro.




| | NoAlert | Alert | Pct |
|---|---|---|---|
| 1 | 9301 | 196 | 2.1% |
| 2 | 7191 | 44 | 0.6% |
| 3 | 5146 | 58 | 1.1% |
| 4 | 4495 | 134 | 2.9% |
| 5 | 1368 | 315 | 18.7% |

The corresponding K-Means plot and summary statistics are shown in the images above. Using K-Means Clustering, we found 5 distinguishable clusters, all of which maintain their integrity when plotted using the first 3 primary components found by PCA. Of those 5 clusters, 3 correspond to easily identifiable levels for particular resource types. It is observed that the Cluster 5 (High Thermal/High Imports/High Hydro) is very significant, generating the highest amount of power among the other clusters, while also producing the highest percentage of alerts compared to the other clusters. However, the number of grid emergencies that occurred during this time period was not evenly distributed within the groups. Cluster 5, all the while producing those significant records, accounts for 315 of the 747 grid emergencies during this time period (42.2%).

### b. Supervised Methods

Supervised methods were utilized to analyze the strength of the connection between the predictors and the outcome.

#### i. Linear Discriminant Analysis

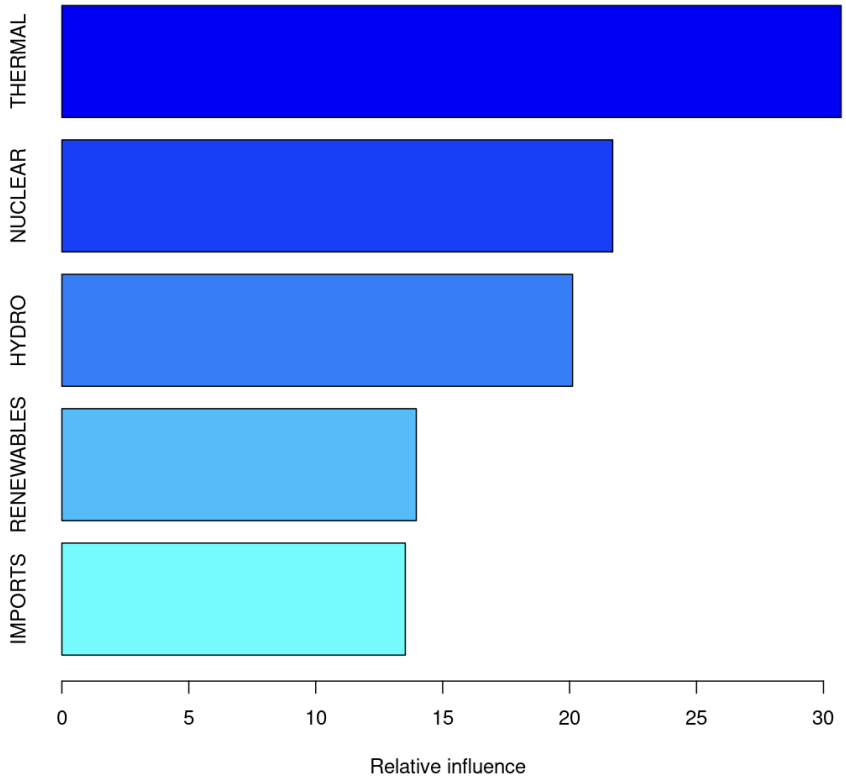
Linear Discriminant Analysis (LDA) is applied to a set of predictor variables when classifying the response variable into two or more classes. It is intended to find a linear combination of features that maximally separates different classes in a dataset. In our analysis, we have the 5 categories of resources: Renewables, Nuclear, Thermal, Hydro, and Imports. The LDA model produced a model with 96.885% accuracy, however struggled to accurately identify grid emergencies, with a sensitivity of 34.722%. The specificity of the model, in other words the accuracy of the model identifying no grid emergencies was 98.511%. In addition to a simple LDA, we ran a K-Fold Cross Validation with LDA, K = 5. The results were similar, with an accuracy of 96.573%, sensitivity of 32.798%, and specificity of 98.306%.

#### ii. K-Nearest Neighbors

K-Nearest Neighbors is used for both classification and regression tasks, making predictions based on the similarities between the input data and its neighboring data points in the feature space. For our analysis we used K = 4, indicating the number of nearest neighbors considered for each point prediction in our dataset. This model produced an accuracy of 97.912%, with a sensitivity of 14.993%, and specificity of 99.113%.

#### iii. Random Forest / Boost

Boosted Random Forests combines the strengths of both Random Forest and boosting algorithms. The ability of the Random Forest handling the dataset and capturing interactions between variables is combined with the boosting's capability to improve model performance to output a model's accuracy through training and weighting of observations. Random Forests remove the correlation between the trees generated from the bootstrapped samples in the bagging method by considering only a specific number of predictors at each split. This in turn discards the significant influence of stronger predictors that may hide the influence of other moderately strong predictors.




Using a Boosted Random Forest in our analysis produced the best model in terms of sensitivity at 94.511%, but produced a lower overall accuracy of 83.188% and specificity of 82.933%. As seen in the summary plot above, the most important resource type in this model is Thermal.

| SplitVar | SplitCodePred | LeftNode | RightNode | MissingNode | ErrorReduction | Weight | Prediction |
|---|---|---|---|---|---|---|---|
| 2 | 18606.50 | 1 | 11 | 15 | 34.43 | 12711.00 | 0.00 |
| 2 | 11020.50 | 2 | 3 | 10 | 4.26 | 12510.00 | -0.02 |
| -1 | -0.05 | -1 | -1 | -1 | 0.00 | 10533.00 | -0.05 |
| 4 | 1257.50 | 4 | 8 | 9 | 7.26 | 1977.00 | 0.14 |
| 1 | 2237.00 | 5 | 6 | 7 | 7.19 | 155.00 | 0.95 |
| -1 | 0.14 | -1 | -1 | -1 | 0.00 | 80.00 | 0.14 |
| -1 | 1.81 | -1 | -1 | -1 | 0.00 | 75.00 | 1.81 |
| -1 | 0.95 | -1 | -1 | -1 | 0.00 | 155.00 | 0.95 |
| -1 | 0.07 | -1 | -1 | -1 | 0.00 | 1822.00 | 0.07 |
| -1 | 0.14 | -1 | -1 | -1 | 0.00 | 1977.00 | 0.14 |
| -1 | -0.02 | -1 | -1 | -1 | 0.00 | 12510.00 | -0.02 |
| 2 | 21315.50 | 12 | 13 | 14 | 8.76 | 201.00 | 1.59 |
| -1 | 1.06 | -1 | -1 | -1 | 0.00 | 140.00 | 1.06 |
| -1 | 2.81 | -1 | -1 | -1 | 0.00 | 61.00 | 2.81 |
| -1 | 1.59 | -1 | -1 | -1 | 0.00 | 201.00 | 1.59 |
| -1 | 0.00 | -1 | -1 | -1 | 0.00 | 12711.00 | 0.00 |

The image above displays the Decision Nodes obtained from our analysis. SplitVar denotes the attribute that the method is looking at to decide which branch to follow. If the corresponding value in that field is less than the value in SplitCodePred, it will go to the row number listed under LeftNode, and if it is greater it will go to the row number listed under RightNode. -1 indicates a terminal node, and Prediction represents the predicted probability.

## 5. Conclusion & Discussion

Various supervised and unsupervised regression methods were applied to the CAISO renewables/ grid emergencies data sets in order to analyze the significance that renewable resources play in power generation as well as causing a shortage of it. The results of our analysis lead to the conclusion that predicting whether there will be a grid emergency for a given point in time would be further enhanced by more information than is contained in the records of hourly power generation by resource type found on the CAISO website. Given that Thermal energy production is most associated with grid emergencies, it is likely that power generation resourcing decisions are driven by anticipated demand, which is likely the driver of the majority of grid emergencies. Focusing only on the power production side, organization and planning of future grid resourcing decisions should focus on combining responsive  production that can be scaled up or tapped when demand exceeds expectations while simultaneously ensuring that sufficient resources exist to prevent loss of power in any situation.

When considering the prospect of further study on this material, projections of anticipated demand and records of actual usage would likely improve the ability to model grid emergencies. With this concept of improved model grid emergencies, we would be able to model grid emergency levels. Another important factor in power availability is the ability for power infrastructure systems to share excess power when demand is low to areas where demand is high. Analyzing how and when power imports and exports occur in California could provide more insight into the occurrence and mitigation of grid emergencies. Analyzing this specific type of data through an anomaly detection regression model should be considered as well in future studies. Since the grid emergencies do not arise as often as non emergencies, it would better suit the dataset to use a model that analyzes the significance of these anomalies and when they happen.

## APPENDIX - R CODE

```r
library(dplyr)

gvo <- readRDS('C:/Users/Charl/Documents/College/2 - Data Mining/project/proc_data/gvo.rds')
str(gvo)
colnames(gvo) = c('Hour', 'RENEWABLES', 'NUCLEAR', 'THERMAL', 'IMPORTS', 'HYDRO',
                  'date', 'dt', 'alert', 'alert_level', 'alert_from', 'alert_to')

# of records in data
nrow(gvo)

# # of records with alert
length(gvo$alert[which(gvo$alert==TRUE)])

# % of records with alerts
length(gvo$alert[gvo$alert==TRUE])/nrow(gvo)

summary(gvo[,2:6])
pairs(gvo[,c(1:6,9,10)])

gen_w_emerg = gvo[,c(1:7,9,10)]
str(gen_w_emerg)

#saveRDS(gvo, file='C:/Users/Charl/Documents/College/2 - Data Mining/project/proc_data/gvo.rds')

# Fancy correlation plot of variables
x11()
library(ggplot2)
library(GGally)
ggpairs(gvo[,c(1:6,9,10)])

x11()
# heatmap() for hourly production by resource
agg_by_hour <- as.matrix(aggregate( . ~ Hour, data=gvo[,c(1:6)], FUN = sum))
agg_by_hour
heatmap(agg_by_hour[,2:6], scale = 'row',
        Rowv = NA, Colv = NA, revC = TRUE, margin=c(11,1),
        main="   Heatmap of power generation by resource by hour")
```

```r
# heatmap of alerts and alert status by hour
agg_alerts <- as.matrix(aggregate(alert ~ Hour, data=gvo, FUN = sum))
agg_alert_status <- as.matrix(aggregate(alert_status ~ Hour, data = gvo, FUN = sum))
agg_alerts
agg_alert_status
agg_alert_merged <- as.matrix(merge(agg_alerts, agg_alert_status))
colnames(agg_alert_merged) <- c("Hour", "Alert", "Status")
heatmap(agg_alert_merged[,c(2,3)], scale='column',
        Rowv = NA, Colv = NA, revC = TRUE, margin=c(9,4),
        main="   Heatmap of alerts and alert status by hour")

# stacked area chart of power generation
library(ggplot2)
library(dplyr)
library(tidyverse)
agg_by_hour_long <- data.frame(agg_by_hour) %>%
  pivot_longer(cols=-1)
colnames(agg_by_hour_long) <- c("Hour", "Resource", "value")
options(scipen=12, digits = 4)
ggplot(agg_by_hour_long, aes(x=Hour, y=value, fill=Resource)) +
  geom_area() +
  ylab("Power Generated (MW)")

# reset graphical device when it stops working
#dev.off()

#### Remember that actions taken between issuance of AWE and actual AWE may
# change generation levels (more imports because of anticipated shortages
# elsewhere, for instance), so even good models need to be analyzed to
# determine actual relationship between variables

attach(gvo)
summary(lm(RENEWABLES ~ Hour))
summary(lm(NUCLEAR ~ Hour))
summary(lm(THERMAL ~ Hour))
summary(lm(IMPORTS ~ Hour))
summary(lm(HYDRO ~ Hour))
```

```r
#### function to get sensitivity and specificity from a set of predicted and actual values
sens_spec <- function(y.pred, y.actual) {
  conf.table <- table(y.pred, y.actual)
  sens <- conf.table[2,2] / sum(conf.table[,2])
  spec <- conf.table[1,1] / sum(conf.table[,1])
  return(c(sens, spec))
}

#### PCA
gvo.pca <- prcomp(gvo[,2:6], center=TRUE)
gvo.pca.sum <- summary(gvo.pca)
gvo.pca.det <- gvo.pca
gvo.pca.sum
biplot(gvo.pca)
plot(gvo.pca)

PCs <- data.frame(gvo.pca$x, gvo[,c(2:6,9)])
head(PCs)


# 3D viz of PC1, PC2, and PC3
library(rgl)

mycolors <- c('royalblue1', 'red')
gvo.pcs$color <- mycolors[ as.numeric(gvo.pcs$gvo.alert) ]

# Plot
plot3d(
  x=gvo.pcs$PC1[-train] , y=gvo.pcs$PC2[-train], z=gvo.pcs$PC3[-train],
  col = gvo.pcs$color[-train],
  type = 's',
  radius = 200,
  xlab="PC 1", ylab="PC 2", zlab="PC 3")
```

```r
#### K-means clustering
set.seed(7)
gvo.kmeans <- kmeans(gvo[,2:6], 5, nstart=40)
gvo.kmeans$cluster

gvo.pcs['cluster'] <- gvo.kmeans$cluster

mycolors <- c('royalblue1', 'red', 'green', 'black', 'yellow')
gvo.pcs$color <- mycolors[ as.numeric(gvo.pcs$cluster) ]

viewSet <- sample(dim(gvo)[1], dim(gvo)[1] * .2)

# Color K-means clusters in 3d plot of PC1, PC2, PC3
plot3d(
  x=gvo.pcs$PC1[viewSet] , y=gvo.pcs$PC2[viewSet], z=gvo.pcs$PC3[viewSet],
  col = gvo.pcs$color[viewSet],
  type = 's',
  radius = 200,
  xlab="PC 1", ylab="PC 2", zlab="PC 3")


gvo_clustered = data.frame(gvo[,2:6], as.factor(gvo.kmeans$cluster), gvo$alert)
sum(gvo_clustered$gvo.alert[which(gvo_clustered$as.factor.gvo.kmeans.cluster.==3)])
sum(gvo_clustered$gvo.alert[which(gvo_clustered$as.factor.gvo.kmeans.cluster.==4)])
```

```
library(GGally)
library(hrbrthemes) # provides themes for axis and plot

x11()
# Parallel coordinate plot to determine how clusters relate to predictor variables
ggparcoord(gvo_clustered[viewSet,],
           columns=1:5, groupColumn=6, scale="globalminmax") +
  xlab("Resource Type") +
  ylab("Power Generated (min-max scaled)") +
  ggtitle("Power Generated by Resource Type") +
  guides(color=guide_legend(title="K-Means\nCluster")) +
  theme(plot.title = element_text(size=12),
        legend.key.size = unit(1, 'cm'), #change legend key size
        legend.title = element_text(size=12), #change legend title font size
        legend.text = element_text(size=10)) #change legend text font size)

#### QDA with raw data
gvo.qda <- qda(alert ~ RENEWABLES + NUCLEAR + IMPORTS + THERMAL + HYDRO, data = gvo,
               subset = train)

gvo.qda.class <- predict(gvo.qda , gvo[-train,])$class
as.data.frame.matrix(table(gvo.qda.class, gvo$alert[-train]))
mean(gvo.qda.class == gvo$alert[-train])
sens_spec(gvo.qda.class, gvo$alert[-train])


gvo.pcs.qda <- qda(gvo.alert ~ PC1 + PC2 + PC3, data=gvo.pcs, subset = train)
gvo.pcs.qda.class <- predict(gvo.pcs.qda, gvo.pcs[-train,])$class
table(gvo.pcs.qda.class, gvo.pcs$gvo.alert[-train])
mean(gvo.pcs.qda.class == gvo.pcs$gvo.alert[-train])
```

```
#### K-Nearest Neighbors
set.seed(11)
train <- sample(dim(gvo)[1], dim(gvo)[1] * .9)

library(class)
train.X <- cbind(gvo[,2], gvo[,3], gvo[,4], gvo[,5], gvo[,6])[train,]
test.X <- cbind(gvo[,2], gvo[,3], gvo[,4], gvo[,5], gvo[,6])[-train,]
train.Y <- gvo$alert[train]
test.Y <- gvo$alert[-train]

knn.k.res <- c(rep(0, 10))
for (i in 1:10) {
  knn.pred <- knn(train.X, test.X, train.Y, k=i)
  gvo.knn.table <- table(knn.pred, test.Y)
  #gvo.knn.table
  knn.k.res[i] <- (gvo.knn.table[1,1] + gvo.knn.table[2,2]) / sum(gvo.knn.table)
}
max(knn.k.res)
which(knn.k.res==max(knn.k.res))
knn.pred <- knn(train.X, test.X, train.Y, k=5)
gvo.knn.table <- table(knn.pred, test.Y)
gvo.knn.table
sens_spec(knn.pred,test.Y)

#### Split train/test
train <- sample(dim(gvo)[1], dim(gvo)[1] * .8)
```

```r
# K-fold CV with any method
for (i in 1:8){
  train_data <- gvo[-((i*3531-3530):(i*3531)), ]
  test_data <- gvo[((i*3531-3530):(i*3531)), ]
  mod <- lda(alert ~ RENEWABLES + NUCLEAR + THERMAL + IMPORTS + HYDRO, data = train_data)
  mod_pred <- predict(mod, test_data)$class
  conf.table <- table(mod_pred, test_data$alert)
  for (j in 1:dim(conf.table)[2]){ mav.df[,j] = mav.df[,j] + conf.table[,j]}
  mae <- c(mae, mean(abs(as.integer(mod_pred == 1) - as.integer(test_data$alert == 1))))
}
mean(mae)
mav.df
(mav.df[1,1] + mav.df[2,2]) / sum(mav.df)
mav.df[2,2] / sum(mav.df[,2])


#### Boosting
library(gbm)
set.seed (7)
gvo.boost <- gbm::gbm(alert ~ RENEWABLES + NUCLEAR + THERMAL + IMPORTS + HYDRO,
                      data = gvo[train , ], distribution = "bernoulli",
                      n.trees = 5000, interaction.depth = 6, shrinkage=.1)
summary(gvo.boost)

yhat.boost <- predict(gvo.boost ,
                      newdata = gvo[-train , ], n.trees = 5000, type="response")
mean ((as.integer(yhat.boost) - as.integer(gvo.test.y))^2)
yhat.boost <- ifelse(yhat.boost>.5, TRUE, FALSE)
sens_spec(yhat.boost, gvo.test.y)
table(yhat.boost, gvo.test.y)

summary(gvo.boost, order=TRUE)
pretty.gbm.tree(gvo.boost, i.tree=6)
```

```r

library(anytime)

directory <- "C:/Users/Charl/Documents/College/2 - Data Mining/project/proc_data/"
fn <- "2018-2022_grid_emerg_data.csv"

emerg_data <- read.csv(paste0(directory, fn))
str(emerg_data)

emerg_data <- emerg_data[!is.na(emerg_data$Date),]
emerg_data$Date <- anydate(emerg_data$Date)
emerg_data <- emerg_data[which(emerg_data$Date < as.Date('2021/11/30')),]
max(emerg_data$Date)

emerg_data <- emerg_data[which(emerg_data$AWE.Event != ""),]

emerg_data$from <- NA
emerg_data$to <- NA

emerg_data$Time.Frame <- gsub("to", "through", emerg_data$Time.Frame)
emerg_data$Time.Frame <- gsub("\n", "", emerg_data$Time.Frame)
emerg_data$Time.Frame <- gsub("at", "", emerg_data$Time.Frame)
emerg_data$Time.Frame <- gsub("  ", " ", emerg_data$Time.Frame)
emerg_data$Time.Frame[19] = "10/1/2020 15:00 through 10/1/2020 22:00"

for (i in 1:length(emerg_data$Time.Frame)) {
  if (nchar(emerg_data$Time.Frame[i]) > 14) {
    tf_parse <- strsplit(emerg_data$Time.Frame[i], "through")[[1]]
    emerg_data$from[i] <- as.character(anytime(tf_parse[1]))
    if (nchar(tf_parse[2]) > 20) {
      emerg_data$to[i] <- as.character(anytime(strsplit(tf_parse[2], "[()]")[[1]][1]))
    } else {
      emerg_data$to[i] <- as.character(anytime(tf_parse[2]))
    }
  } else {
    tf_parse <- strsplit(emerg_data$Time.Frame[i], "- ")[[1]]
    emerg_data$from[i] <- as.character(anytime(paste(emerg_data$Date[i], tf_parse[1])))
    emerg_data$to[i] <- as.character(anytime(paste(emerg_data$Date[i], tf_parse[2])))
  }
```

```
}

emerg_data$from <- as.POSIXct(emerg_data$from, format = "%Y-%m-%d %H:%M:%S")
emerg_data$to <- as.POSIXct(emerg_data$to, format = "%Y-%m-%d %H:%M:%S")
str(emerg_data)

# create combined DF (generation versus outages)
gvo <- t2_all

library(hms)
gvo$dt <- paste(gvo$date, hms(hour = as.numeric(gvo$Hour)))
gvo$dt[1:10]
gvo$dt <- as.POSIXct(gvo$dt, format = "%m/%d/%y %H:%M:%S")
str(gvo)

# identify date time groups that didn't convert correctly
gvo[is.na(gvo$dt),]

# convert observation 6770
gvo$dt[6770] = as.POSIXct(paste("2019/03/19", hms(hour=as.numeric(gvo$Hour[6770]))),
                          format="%Y/%m/%d %H:%M:%S")
gvo[6770,]

# convert observation 15242
gvo$dt[15242] = as.POSIXct(paste("2020/03/08", hms(hour=as.numeric(gvo$Hour[15242]))),
                           format="%Y/%m/%d %H:%M:%S")
gvo[15242,]
### DST is the problem. +7200 should be 2am, but it skips to 3am
### which begs the question, why does this day have 24 hours worth of data?
corrdate <- as.POSIXct("2020/03/08")
gvo$dt[15242] <- as.POSIXct(corrdate + 7199)
gvo$dt[15242]

# convert observation 24146
gvo$dt[24146] = as.POSIXct(paste("2021/03/14", hms(hour=as.numeric(gvo$Hour[24146]))),
                           format="%Y/%m/%d %H:%M:%S")
```

```
gvo[24146,]
### DST is the problem. +7200 should be 2am, but it skips to 3am
### which begs the question, why does this day have 24 hours worth of data?
corrdate <- as.POSIXct("2021/03/14")
gvo$dt[24146] <- as.POSIXct(corrdate + 7199)
gvo$dt[24146]

# create new columns for merging A/W/E data with hourly power generation data
gvo$alert <- FALSE # True if an A/W/E was in effect during that hour, o/W False
gvo$alert_status <- 0 # RMO = 1, Flex Alert = 2, Alert = 3, Warning = 4, Stage 2 = 5,
gvo$alert_from <- NA # DTG A/W/E started
gvo$alert_to <- NA # DTG A/W/E ended

# Remove A/W/E events that were explicitly unrelated to power generation
levels(as.factor(emerg_data$AWE.Event))
emerg_data <- emerg_data[emerg_data$AWE.Event != "1-hour \nNotification",]
emerg_data <- emerg_data[emerg_data$AWE.Event != "Transmission \nEmergency",]

# Iterate through emerg_data and gvo to fill in gvo alert related fields
for (j in 1:length(emerg_data$from)) {
  for (k in 1:length(gvo$dt)) {
    if (gvo$dt[k] >= emerg_data$from[j] & gvo$dt[k] < emerg_data$to[j]) {
      gvo$alert[k] = TRUE
      gvo$alert_from[k] = as.character(as.POSIXct(emerg_data$from[j],
                                      format = "%Y-%m-%d %H:%M:%S",
                                      origin = "1970-01-01"))
      gvo$alert_to[k] = as.character(as.POSIXct(emerg_data$to[j],
                                    format = "%Y-%m-%d %H:%M:%S",
                                    origin = "1970-01-01"))
      if (emerg_data$AWE.Event[j] == "RMO") {gvo$alert_status[k] = 1}
      else if (emerg_data$AWE.Event[j] == "Flex Alert") {gvo$alert_status[k] = 2}
      else if (emerg_data$AWE.Event[j] == "Alert") {gvo$alert_status[k] = 3}
      else if (emerg_data$AWE.Event[j] == "Warning") {gvo$alert_status[k] = 4}
      else if (emerg_data$AWE.Event[j] == "Stage 2") {gvo$alert_status[k] = 5}
      else if (emerg_data$AWE.Event[j] == "Stage 3") {gvo$alert_status[k] = 6}
    }
  }
}
```

```python
# -*- coding: utf-8 -*-

import camelot
import pandas as pd
from os import listdir
from pathlib import Path

# set directory to retrieve grid emergency history report pdf
directory = "C:/Users/Charl/Documents/College/2 - Data Mining/project"
fn = "/Grid-Emergencies-History-Report-1998-Present.pdf"

# use camelot to extract tables from pdf by page # range (need p_b=True for 45-127)
tables = camelot.read_pdf(directory + fn, pages='45-127', process_background=True)

# check 1st table to ensure data is read correctly
tables[0].df

# set directory to save tables as csv files
directory = "C:/Users/Charl/Documents/College/2 - Data Mining/project/raw_data/GEHD"
tables.export(directory + '/Grid_Emergencies_History_Data.csv', f='csv')

# create list to save path/to/filename for each csv file
file_list = []

# iterate through directory saving each path/to/filename
for filename in listdir(directory):
    file_list.append(directory + '/' + filename)

# concatenate data from each pdf into a single pandas df
df = pd.concat(map(pd.read_csv, file_list),ignore_index=True)
```

```python
# drop all data that is not in the relevant columns
grid_emerg_data1 = df.iloc[:,0:6]
grid_emerg_data2 = df.iloc[:,6:12]

# rename columns
grid_emerg_data1.columns = ['Date', 'Day', 'Region', 'Time Frame', 'AWE Event', 'Reason']
grid_emerg_data2.columns = ['Date', 'Day', 'Region', 'Time Frame', 'AWE Event', 'Reason']

# combine dfs
frames = [grid_emerg_data1, grid_emerg_data2]
grid_emerg_data = pd.concat(frames)
grid_emerg_data.dropna(how='all', inplace=True)

# drop rows that contain all NAs
grid_emerg_data.dropna(how='all', inplace=True)

# check data type of each column
grid_emerg_data.info()

# change Date column to date_time data type some dates are ranges, so does not work as is
# grid_emerg_data['Date'] = pd.to_datetime(grid_emerg_data['Date'])

# save data to csv file
directory = "C:/Users/Charl/Documents/College/2 - Data Mining/project/proc_data"
filepath = Path(directory + "/2018-2022_grid_emerg_data.csv")
grid_emerg_data.to_csv(filepath, index=False)
```

```r

### download.file needs matching lists of source and destination files

# set recurring part of file as string
fn <- '_DailyRenewablesWatch.txt'

# set base of source url
url <- "http://content.caiso.com/green/renewrpt/"

# set directory for saving downloaded files
directory <- "C:/Users/Charl/Documents/College/2 - Data Mining/project/raw_data/DRW/"

loop_pts <- c("2018/06/01", "2018/12/01", "2019/06/01", "2019/12/01",
              "2020/06/01", "2020/12/01", "2021/06/01", "2020/11/30")

for (i in 2:length(loop_pts)) {

  # create a sequence of dates to use as part of filename
  dates <- seq(as.Date(loop_pts[i-1]), as.Date(loop_pts[i]), "days")

  # create list of filenames for month of November
  files <- paste0(gsub('-', '', dates), fn)

  # create list of urls for November files
  urls <- paste0(url, files)

  # create list of destinations for downloaded files
  paths <- paste0(directory, files)

  # download and save files
  download.file(urls, paths)

  Sys.sleep(5)

}

```