Generalized Additive Models & High-dimensional methods

Rakhshanda Jabeen

11/12/2019

Contents

1. GAM and GLM Models To Examine The Mortality Rates

In this task we are examining weekly data on the mortality and the number of laboratory-confirmed cases of influenza in Sweden.

1.1 Visual Inspection of Mortality and Influenza

In following time-series plot, it can be easily obserbed that mortatlity rate increases as influenza cases increases. One can say that influenza cases and mortlity rate have positive correlation between eachother.

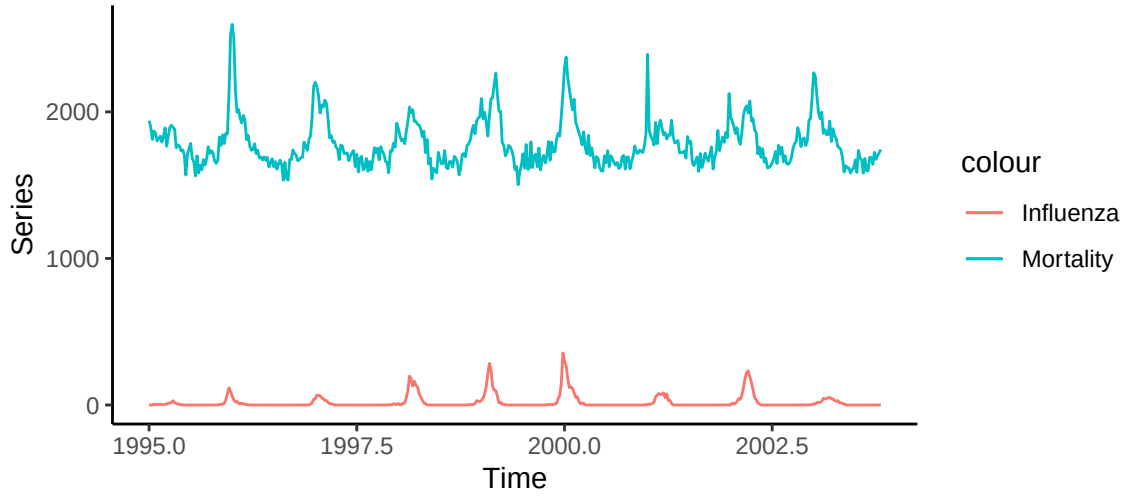


Figure 1: Time Series Plot

1.2 GAM model

In this task, we have modeled Mortality as a lineay function of feature Year and a spline function of week. The underlying **probabilistic model** is as follows:

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week)
##
## Estimated degrees of freedom:
## 3.43  total = 5.43
##
## GCV score: 9783.401
```

$$\text{Mortality} = \beta_0 + \beta_1 * s(\text{week}) + \beta_2 * \text{year} + \epsilon$$
$$\epsilon = \mathcal{N}(0, \sigma)$$

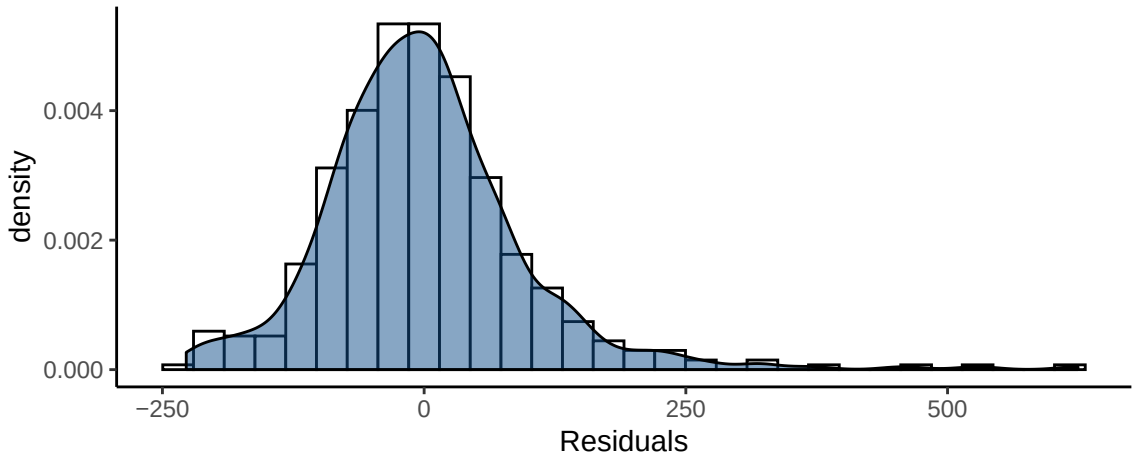


Figure 2: Residuals

1.2.1. Histogram of Residuals

It is evident from Figure 2. that residuals of GAM model are normally distributed with $\mu = 0$.

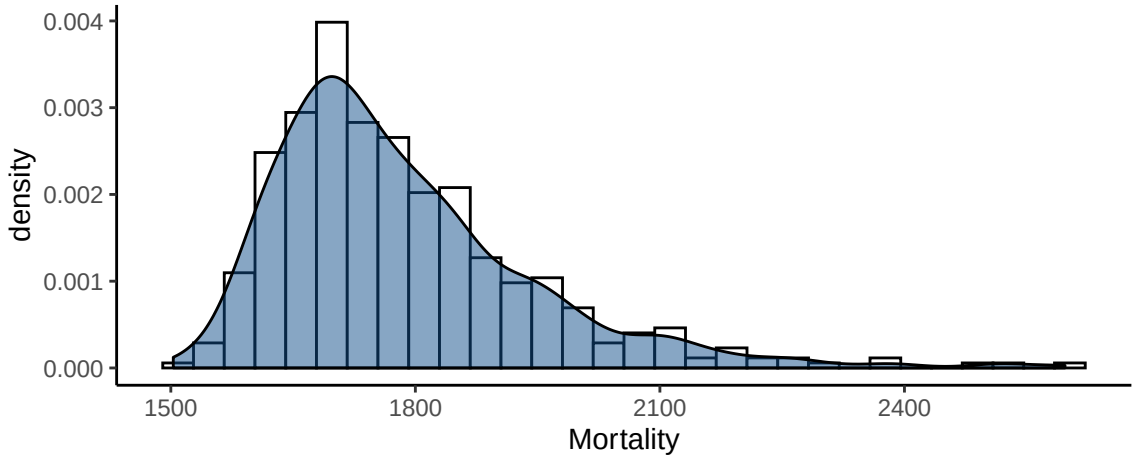


Figure 3: Density Curve of Mortality

In Figure 3. it can be observed that Mortality density curve has a bell shaped curve and is a little left skwed due to some outliers so we can say that Mortality $\sim \mathcal{N}(\mu, \sigma)$.

1.3 Analysis Of GAM Model

In Figure 4. we can observe that mortality data is representing same trend every year excep a few outliers in the intial year. Mortality curve has a higher peak at the start of each year.

It is evident that the predicted curve is showing the same trend as original curve. Thus one can say that GAM model is a good approximation of data but it is unable to capture all the high peaks of mortality curve.

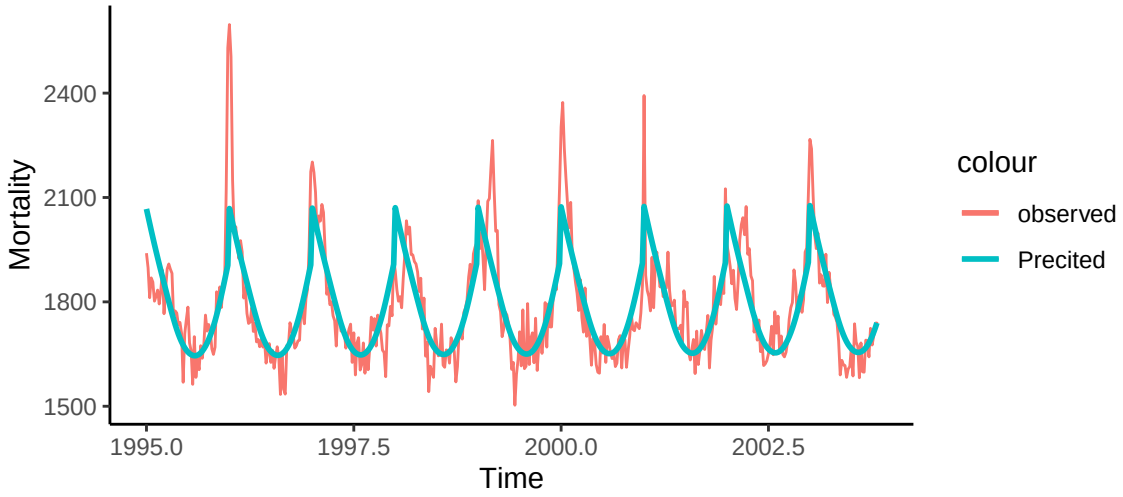


Figure 4: GAM Model

Figure 5. representes the dependence of mortality rate on spline function of week. It is evident that in initial weeks of the year we have more influenza cases as compare to the middle of the year. This is because of the different whether conditions. Thus we can infer that people suufer more from influenza in winter.

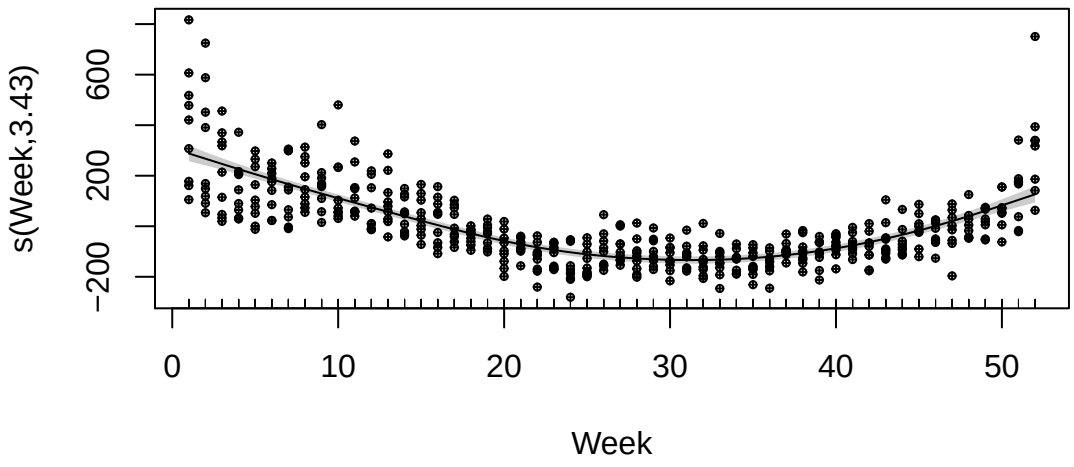


Figure 5: spline Component

1.4. Penalty Factor of The Spline Function in GAM

In this task we are examining how the penalty factor of the spline function in the GAM model influences the estimated deviance and degree of freedom of the model.

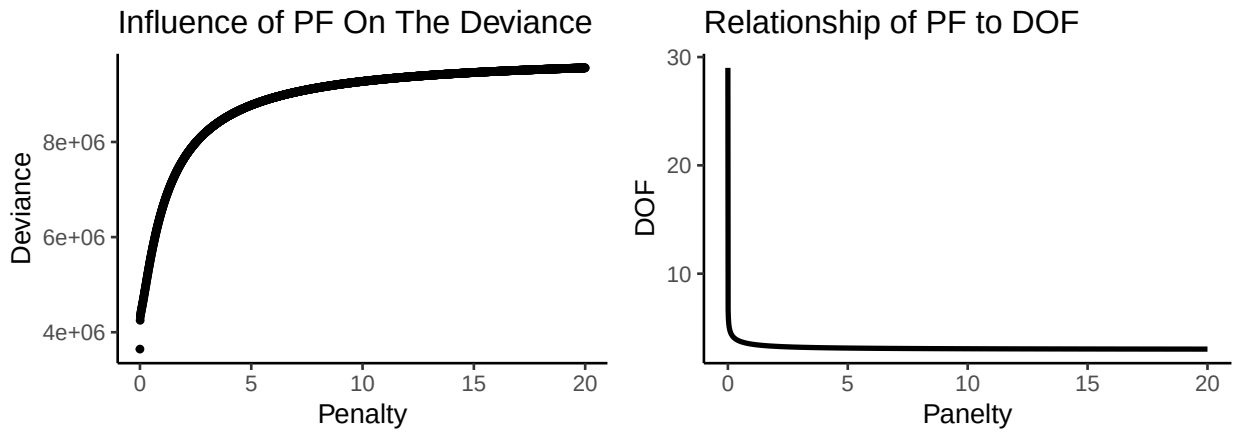


Figure 6: Deviance

Figure 6.indicates that an increse in the penalty factor of the spline function causes an increases in the estimated deviance of the model. Whilst an increse in the the penalty factor causes a decrease in the degrees of freedom.

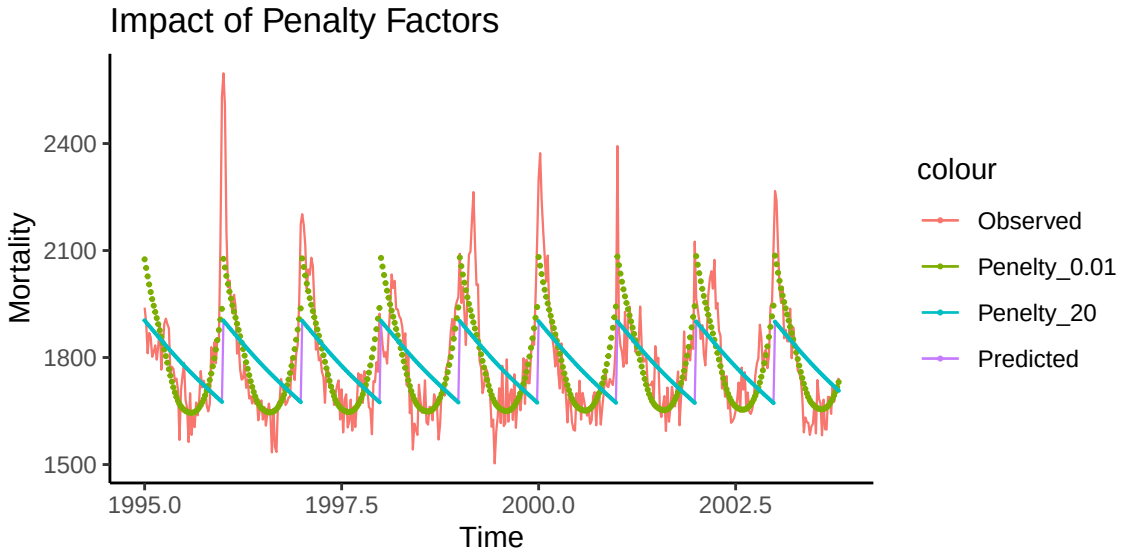


Figure 7: Penalty Factors

It is evident from Figure 7. that a high value of penalty factor for spline function yields a vey simple model which is unable to predict the high peaks of the original curve. Whilst with a vey small value of panelty factor predicted curve is trying to capture all the points. Thus we can infer that a very small value of penalty factor can overfit the data. We can choose optimal penalty factor by cross validation.

1.5. Influenza and GAM Residuals

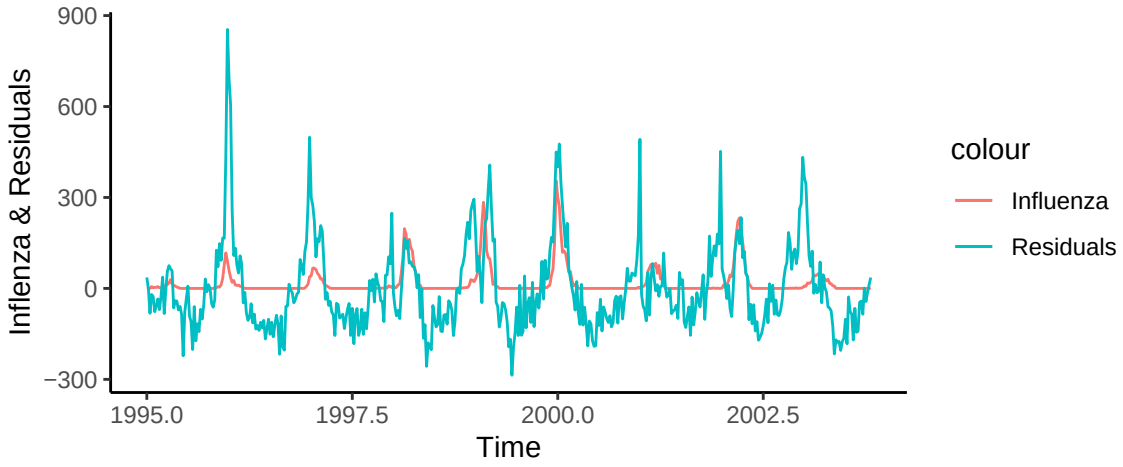


Figure 8: Comaprison of Influenze curve with Residulas of Model

It is evident from Figure 8. that number of influenza cases and residuals of GAM model are positively correlated to each other. A peak in influenza cases indicates a peak in residuals of the model.

1.6. Modelling as an Additive Function Of Spline Functions Of Year, Week & Influenza cases

In this task we are fitting a GAM model in which mortality is modelled as an additive function of the spline functions of year, week, and the number of confirmed cases of influenza.

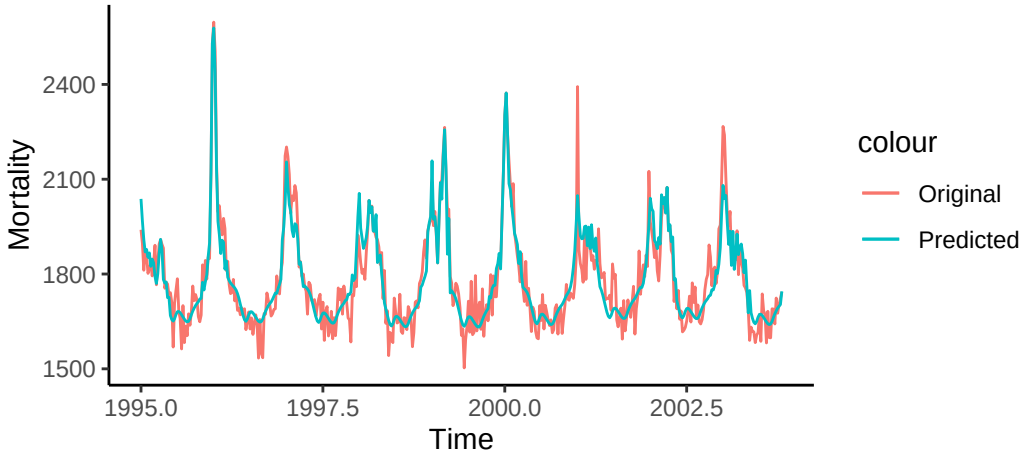


Figure 9: GAM Model

The additive model is predicticting the mortality rate more effectively. As we can see in Figure 9. that predicted curve is capturing all the high and low peaks of original data.

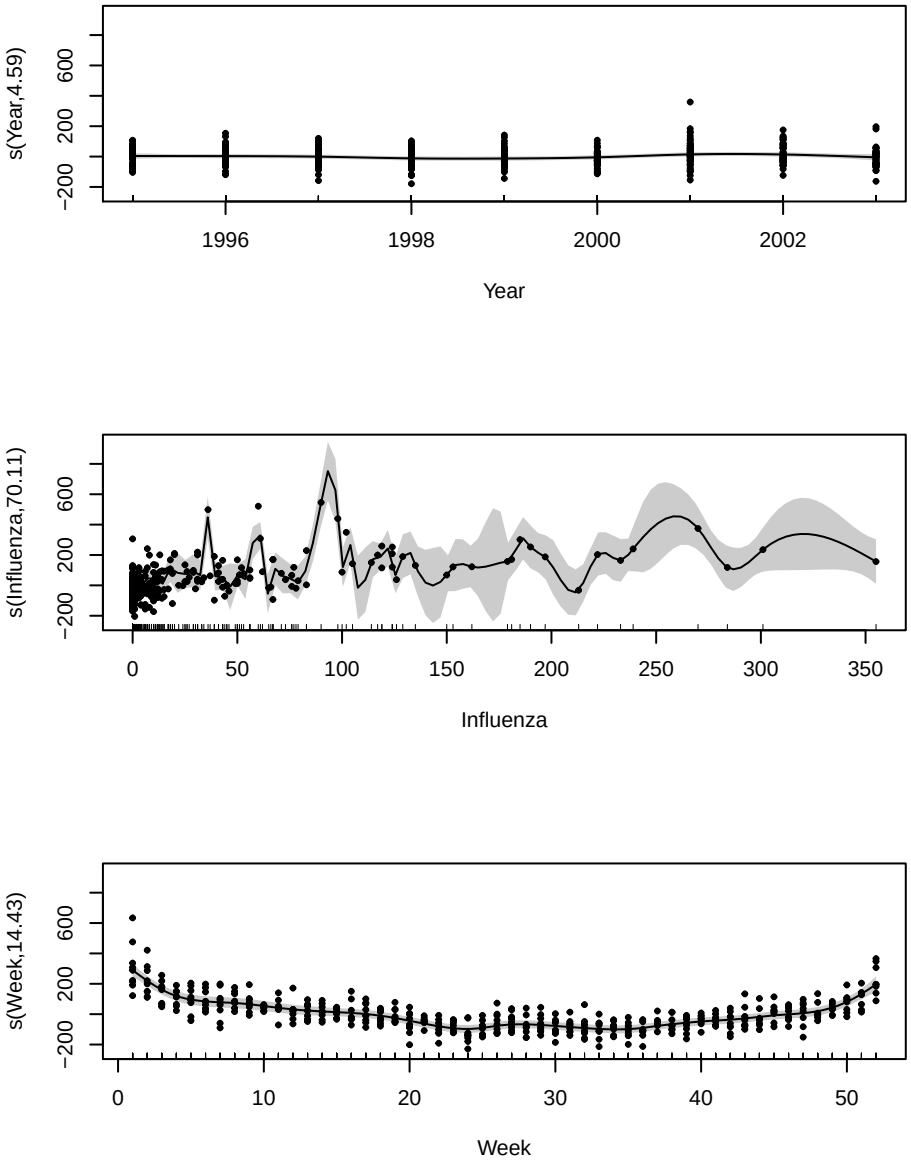


Figure 10: Spline Components

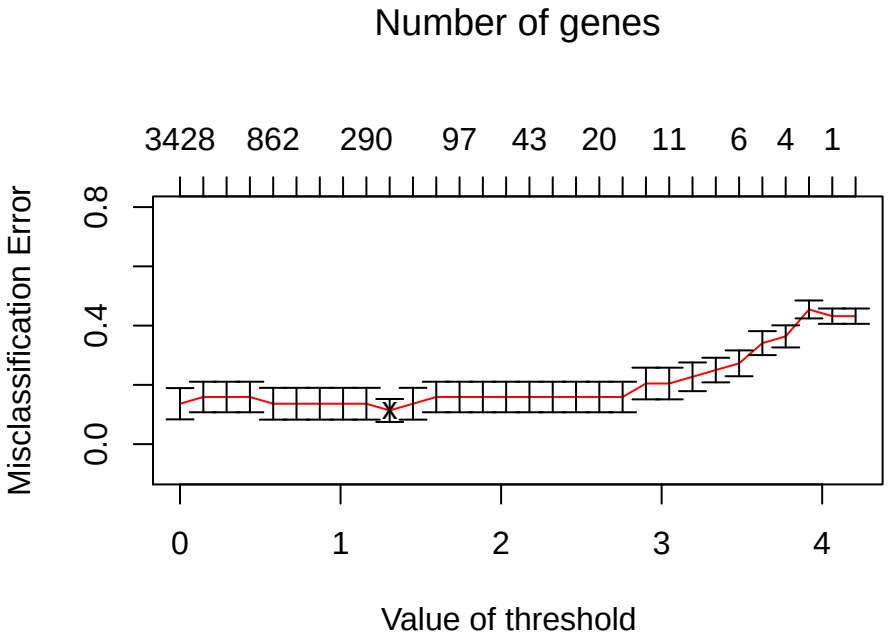
In Figure 9. we can observe that feature of influenza is most influential on the moartality rate whilst year feature hast the least impact on mortality rate.

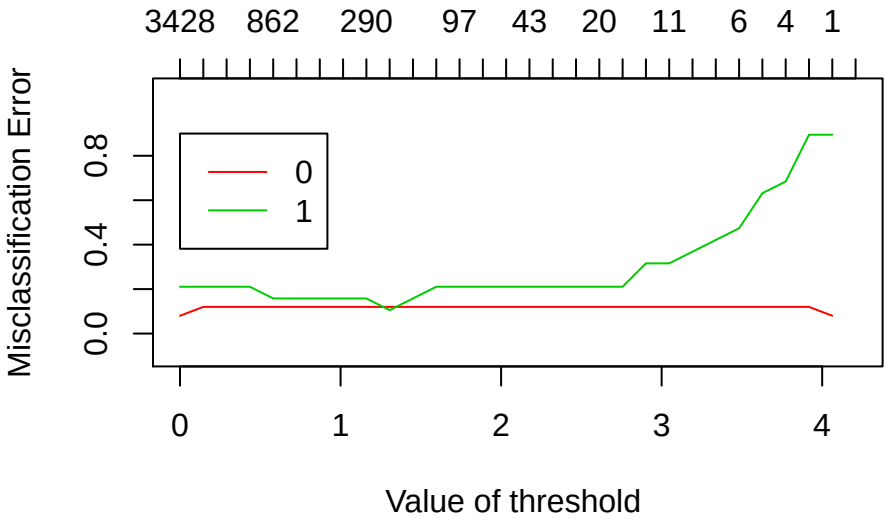
2. High-Dimensional Methods

Our data contains information about 64 e-mails which were manually collected from DBWorld mailing list. The data is consist of 64 observation and 4703 features which indicates that it is a wide data.

2.1. Nearest Srunken Centroid Classification

In this task we divide the data into training and test sets (70/30) without scaling. Then we perform nearest shrunken centroid classification on training data and choose threshold by cross-validation.





The value of threshold after cross validation is 1.3059335.

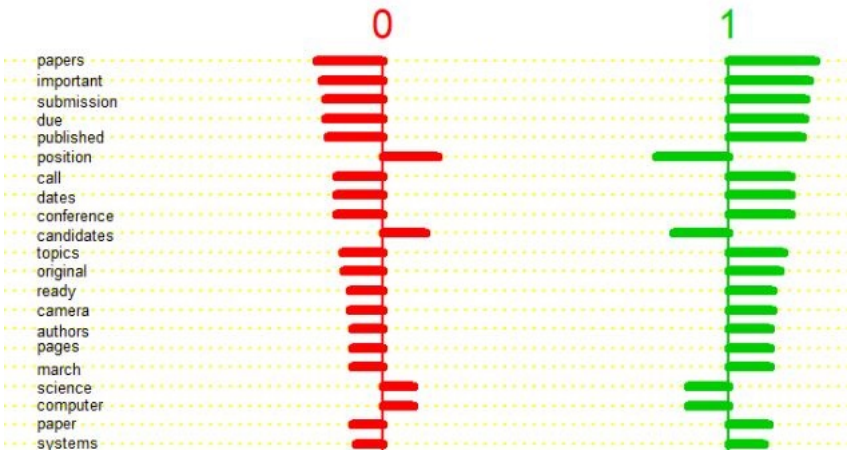


Figure 11: Centroid plot

Centroid plot represents the shrunken class centroids for each class, for genes surviving the threshold for at least one class. In this task our data is calssified into classes whether the e-mail is a conference or not conference.
In Figue 11. we can see that the words papers, important, submission, due and published have huge distance from the mean to the right and hence these words have a strong decisive power whether an email is conference or not. Thus if these words are appearing in an email there is a very high probability that email is conference.

On the other hand, word position and candidate are away from the mean to the left indicating that if these words are appearing in an e-mail then there is a high probability that e-mail is not conference.

2.2.1. Features Selected By NSC

When we performed nearest shrunken centroid classification on the training data. it selects 231 features. Following table represents the top 10 selected features by the method.

Table 1: Top selected Features

id	name	0-score	1-score
3036	papers	-0.3814	0.5019
2049	important	-0.3519	0.4631
4060	submission	-0.3368	0.4431
1262	due	-0.3301	0.4344
3364	published	-0.3223	0.4241
3187	position	0.318	-0.4184
596	call	-0.2717	0.3575
869	conference	-0.2698	0.355
1045	dates	-0.2698	0.355
607	candidates	0.2468	-0.3247

Table 2: Confusion Matrix Of NSC

	0	1
0	10	0
1	2	8

Thus misclassification rate of the test is 10%.

2.

2.2.1. Elastic Net With Binomial Response

In this task we are fitting an elastic net model with the binomial response and $\alpha = 0.5$ in which penalty is selected by cross-validation to our test data.

Elastic net is the combination of both ridge and lasso regression.We can say thay if we want to implement the functionality of both methods equally then we use choose $\alpha = 0.5$.

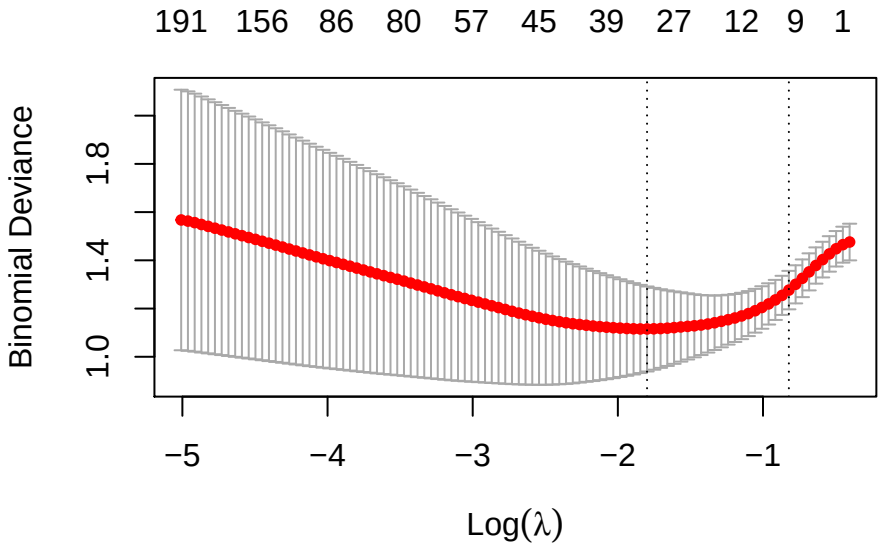


Figure 12: Elastic Net Model

The optimal value of λ which yields minimum error in cross-validation is 0.1655087.

Table 3: Confusion Matrix Of EN

	0	1
0	10	0
1	2	8

Thus misclassification rate of the test is 10%.

2.2.2. Support Vector Machine (SVM)

In this task we are fitting an SVM model with "vanilladot" kernel to our test data.

```
##  Setting default kernel parameters
```

Table 4: Confusion Matrix Of SVM

	0	1
0	10	1
1	0	9

Thus misclassification rate of the test is 5%.

2.2.3. Comparison of NSC, Elastic Net & SVM models

The following table represents the eroor rates and number of features usd by each method for classification.

Table 5: Confusion Matrix

Model	Error.Rates	Features
NSC	10	231
EN	10	33
SVM	5	43

Table 5. represents the comparison of all the models. It is evident that misclassification rate of SVM model is least as comoare to other two methods and it uses 43 to achieve that. Thus we will prefer SVM model amongst all of the rest.
Whilst nearest shruken cetroids and elastic net models have the same missclassification rate but elastic net is using 33 features to achieve that so we can say that elastic net works better between te two of them for this particular data set.

2.3. Benjamini-Hochberg Method

In this task we are implementing Benjamini-Hochberg methods on the origial data. The *Benjamini-Hochberg* method is a powerful tool to decrease the False Discovery Rate. We are following these steps:

- Setting individual p-values in ascending order.
- Assigning ranks to the p-values. The smallest has a rank of 1, the second smallest has a rank of 2 ans so on.
- Calculate each individual p-value's Benjamini-Hochberg critical value, using the formula $(\frac{i}{m})*Q$, where:

$$i = \text{individual p-value rank}$$
$$m = \text{totl number of tests}$$
$$Q = \text{ the false discovery rate}$$

- Compare original p-values to the BH-critical values and then find the largest p-value which is smaller than the critical value.

Pllyinh BH-method on our data it selects features. The selected top 10 features are shown in table 6.

Table 6: Top Selescted Features

	Features	p.value	BH_value
3036	papers	0.0e+00	0.0000005
4060	submission	0.0e+00	0.0000019
3187	position	0.0e+00	0.0000129
3364	published	2.0e-07	0.0002157
2049	important	3.0e-07	0.0002860
596	call	4.0e-07	0.0003122
869	conference	5.0e-07	0.0003420
607	candidates	9.0e-07	0.0005062
1045	dates	1.4e-06	0.0006576
3035	paper	1.4e-06	0.0006576

Appendix

```
setwd("E:/Machine Learning/lab 02 block 02")
library(knitr)
library(kableExtra)
library(ggpubr)
library(readr)
library(readxl)
library(pamr)
library(ggplot2)
library(mgcv)
library(e1071)
library(glmnet)
library(kernlab)
library(dplyr)
library(caret)
colorize <- function(x, color) {
  if (knitr::is_latex_output()) {
    sprintf("\\textcolor{%s}{%s}", color, x)
  } else if (knitr::is_html_output()) {
    sprintf("<span style='color: %s;'>%s</span>", color,
      x)
  } else x
}
RNGversion('3.5.1')
data <- read_excel("Influenza.xlsx")
ggplot(data) + geom_line(aes(x=Time,y = Mortality,col="Mortality"), size = 0.5) +
  geom_line(aes(x=Time, y = Influenza ,col="Influenza"), size = 0.5)+ ylab("Series")+theme_classic()
gam_model = gam(formula = Mortality~Year+s(Week), data = data,  sp = data$Week, method = "GCV.Cp")
fit = predict(gam_model, data)
gam_model
ggplot(data=data.frame(Residuals = gam_model$residuals), aes(x=Residuals)) +  geom_histogram(aes(y=..de
  geom_density(alpha=.5, fill="dodgerblue4")+theme_classic()
ggplot(data, aes(x=Mortality)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.5, fill="dodgerblue4")+theme_classic()
fitted_df = cbind( "Predicted" = fit, data)
ggplot(fitted_df) + theme_classic()+
  geom_line(aes(x=Time, y = Mortality,col="observed"), size = 0.5)+
  geom_line(aes(x=Time, y = fit, col="Precited"),  size = 1)
plot(gam_model, pch = 10, cex = 0.5,scheme = 1, residuals = T)
Panelty = seq(0,20, by = 0.01)
Deviance = 0
Residuals = 0
DOF = 0
panelty_df = as.data.frame(data)
count = 1
for (i in Panelty) {
  gam_model = gam(formula = Mortality~Year+s(Week,k=length(unique(data$Week)),sp=i),
                  data = data,  sp = data$Week, method = "GCV.Cp")
  fit = predict(gam_model,newdata = data)
  panelty_df[paste0("Panelty_",i)] = fit
  Deviance[count] = deviance(gam_model)
```

```
  Residuals[count] = residuals(gam_model)
  DOF[count] = sum(gam_model$edf)
  count = count + 1
}
dv_df = data.frame(Panelty, Deviance)
deg_df = data.frame(Panelty, DOF)
p = ggplot(dv_df)+geom_point(aes(x = Panelty, y = Deviance), size = 1)+
  theme_classic()+xlab('Penalty')+ggtitle('Influence of PF On The Deviance')
q = ggplot(deg_df)+geom_line(aes(x = Panelty, y = DOF), size = 1)+
  theme_classic()+ggtitle('Relationship of PF to DOF')
ggarrange(p,q, nrow = 1, label.x = 'Penalty')
ggplot(panelty_df) + geom_line(aes(x=Time, y = Mortality,col="Observed"), size = 0.4)+
  geom_line(aes(x=Time, y = fit, col="Predicted"), size=0.4)+
  geom_point(aes(x=Time, y = Panelty_0.01, col="Penelty_0.01"), size=0.4)+
   geom_point(aes(x=Time, y = Panelty_20, col="Penelty_20"), size = 0.1)+
  theme_classic()+ ggtitle('Impact of Penalty Factors')
res_df = cbind(data, "Residuals" = residuals(gam_model))
ggplot(res_df)+ geom_line(aes(x = Time, y = Influenza, col = "Influenza"))+
  geom_line(aes(x = Time, y = Residuals, col = "Residuals"))+ ylab("Inflenza & Residuals")+ theme_classi
add_model = gam(formula = Mortality ~ s(Year,k = length(unique(data$Year)))                    + s(I
                  s(Week,k = length(unique(data$Week))),data=data)
add_fit = predict(add_model, data)
add_df = cbind(data,add_fit)
ggplot(add_df) + geom_line(aes(x=Time, y = Mortality,col="Original"))+
  geom_line(aes(x=Time, y = add_fit, col="Predicted"))+ theme_classic()
par(mfrow=c(3,1))
plot.gam(add_model, pch = 10, cex = 0.5,scheme = 1, residuals = T)
data <- read.csv2("data.csv",
                  sep =";",header = TRUE)
n = nrow(data)
set.seed(12345)
id=sample(1:n, floor(n*0.7))
train=data[id,]
test=data[-id,]
rownames(train) = 1:nrow(train)
rownames(test) = 1:nrow(test)
x_train = t(train[,-4703])
y_train = train[[4703]]
x_test = t(test[,-4703])
y_test = as.factor(test[[4703]])
#_____NSC classification____#
nsc_train = list(x = x_train, y = as.factor(y_train),
                 geneid=as.character(1:nrow(x_train)),
                 genenames=rownames(x_train))
model = pamr.train( nsc_train)
# cv fit
cv_model=pamr.cv(model,nsc_train, nfold = 10)
min = cv_model$threshold[which.min(cv_model$error)]
model1 = pamr.train(nsc_train,threshold = min)
slctd_features = pamr.listgenes(model1, nsc_train,
                                threshold = min, genenames=TRUE)
n_features = nrow(slctd_features)
top_features = slctd_features[1:10,]
```

```
#centroid plot
# a = pamr.plotcen(model1, nsc_train, threshold=1.3)
pamr.plotcv(cv_model)
kable(top_features, "latex", caption = "Top selected Features", booktabs = T) %>%
kable_styling(latex_options = "HOLD_position")
fit1 = pamr.predict(model1,x_test,threshold = min, type = "class")
conf_mat1 = table(y_test,fit1)
error_rates = function(conf_matrix)
{
  return((1- sum(diag(conf_matrix))/sum(conf_matrix))*100)
}
error1 = error_rates(conf_mat1)
kable(conf_mat1, "latex", caption = "Confusion Matrix Of NSC", booktabs = T) %>%
kable_styling(latex_options = "HOLD_position")
#_____elastic.net____#
elx_train = as.matrix(train[,-4703])
ely_train = as.matrix(train[[4703]])
elx_test = as.matrix(test[,-4703])
ely_test = as.matrix(test[[4703]])
EN_model <- cv.glmnet(x=elx_train,y=ely_train,
                          family = "binomial",alpha = 0.5)
opt_lambda = EN_model$lambda.min
fit2 = glmnet(x=elx_train,y=ely_train,family = "binomial",
              alpha = 0.5, lambda = opt_lambda)
prediction = predict(fit2,elx_test,s = opt_lambda,type = "class")
conf_mat2 = table(ely_test,prediction)
error2 = error_rates(conf_mat2)
en_feat = coef(fit2,opt_lambda)
en_feat = length(en_feat@Dimnames[[1]][en_feat@i + 1])
plot(EN_model)
kable(conf_mat2, "latex", caption = "Confusion Matrix Of EN", booktabs = T) %>%kable_styling(latex_optio
#____SVM_____#
svm_model = ksvm(Conference ~., data=train,kernel="vanilladot",
                 scaled=FALSE,type="C-svc")
fit3 <- predict(svm_model,test,type="response")
svm_feat = length(train[SVindex(svm_model)])
svm_top_feat = colnames(train[SVindex(svm_model)])
conf_mat3 = confusionMatrix(as.factor(fit3),as.factor(y_test))$table
error3 = error_rates(conf_mat3)
comp = data.frame("Model" = c("NSC","EN","SVM"),
                  "Error Rates" = c(error1, error2, error3),
  "Features" = c(n_features ,en_feat, svm_feat))
kable(conf_mat3, "latex", caption = "Confusion Matrix Of SVM", booktabs = T) %>%
kable_styling(latex_options = "HOLD_position")
kable(comp, "latex", caption = "Confusion Matrix ", booktabs = T) %>%
kable_styling(latex_options = "HOLD_position")
set.seed(12345)
p =  NULL
for (j in 1:4702)
{
  y = data[,j]
  p = rbind(p,data.frame(Features = colnames(data)[j],
    p.value  = t.test(y ~ Conference, data)$p.value))
```

```
}
p = p[order(p$p.value),]
m = nrow(p)
BH_value = integer(m)
critical_value = p$p.value[m]
BH_value[m] = p$p.value[m]
for (i in (m-1):1)
{
  adjustCalc = p$p.value[i]*(m/i)
  BH_value[i] = min(critical_value,adjustCalc)
  critical_value = BH_value[i]
}
p$BH_value= BH_value
df = p[which(p$BH_value<= 0.05), ]
df = df[order(df$p.value), ]
kable(df[1:10,], "latex", caption = "Top Selescted Features", booktabs = T) %>%
kable_styling(latex_options = "HOLD_position")
```