Final Project for CMDA 3654

Kennychen

Honor code:

"I have neither given nor received unauthorized assistance on this assignment." KC.

I receive help from "" and give help to "".

Part I

Data description

Introduction

Forests are human lungs. It covers almost one third of the earth's surface. It can bring a lot of fresh oxygen to cities and reduce the rate of earth warming. However, according to scientific reports, our forests are still insufficient and we need to continue to plant trees. In particular, China has economic losses caused by the lack of forests. Trees have the function of catching the land and locking water. The lack of trees will lead to soil erosion, causing rivers to overflow every year, and water loss will lead to desertification. Every fall, the wind will carry sand into Beijing, seriously affecting Beijing's air quality. Therefore trees are very important to humans.
Forest fire safety has now become a special issue that people need to consider. A large-scale fire will severely damage the forest and cause huge losses to people's wealth. Once the forest suffers a fire, the most intuitive hazard is to burn or burn the trees. Forest is a renewable resource with a long growth cycle. After a fire, it takes a long time to recover. And it is difficult to restore the forest to its original appearance, and the old plants will be greatly reduced. If it is repeatedly damaged by fire, it may become wasteland or even bare land. At the same time, forest burning will produce a lot of smoke, the main components of which are carbon dioxide and water vapor. In addition to water vapor, other substances will cause air pollution, endangering human health and the survival of wild animals.

Data Set Information

This date is from Department of Information Systems, University of Minho, Portugal. The purpose of this data is to predict the burning area of forest fires in northeastern Portugal by using meteorological and other data. At the same time, the donors analyze the data inside to make weather forecasts to reduce the loss caused by the fire.

Date Introduntion

1. X - x-axis spatial coordinate within the Montesinho park map: 1 to 9

2. Y - y-axis spatial coordinate within the Montesinho park map: 2 to 9
3. month - month of the year: ‘jan’ to ‘dec’
4. day - day of the week: ‘mon’ to ‘sun’
5. FFMC - FFMC index from the FWI system: 18.7 to 96.20
6. DMC - DMC index from the FWI system: 1.1 to 291.3
7. DC - DC index from the FWI system: 7.9 to 860.6
8. ISI - ISI index from the FWI system: 0.0 to 56.10
9. temp - temperature in Celsius degrees: 2.2 to 33.30
10. RH - relative humidity in %: 15.0 to 100
11. wind - wind speed in km/h: 0.40 to 9.40
12. rain - outside rain in mm/m2 : 0.0 to 6.4
13. area - the burned area of the forest (in ha): 0.00 to 1090.84

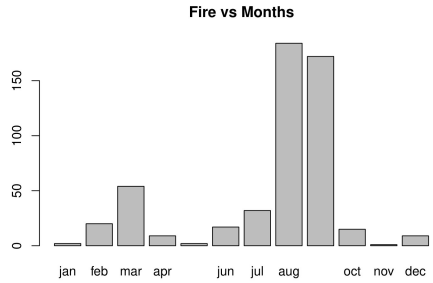
Model Setup

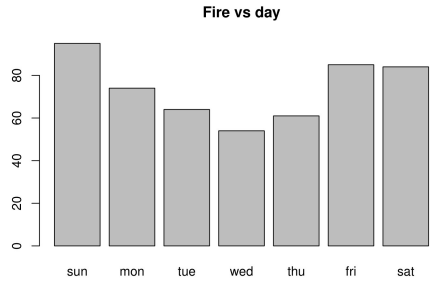
Methed

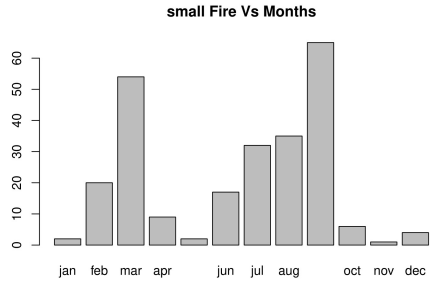
I Will use multiple regression to fit my model. since the date is small and contentious,which I think this is best fit.I think that the key to the occurrence of fire is environmental reasons, so I think temperature, humidity, wind, and rain are particularly important for fire. Because these are my main considerations.

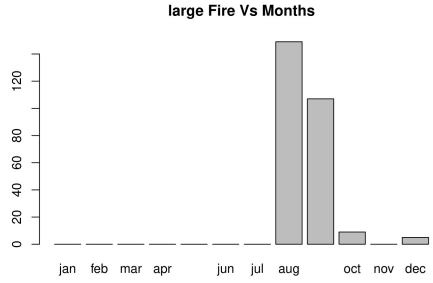
Visualize The Data

I used data to draw the relationship between fire and month, and the relationship between fire and each week, because fire is closely related to month. So I drew the relationship between major fires and months and the relationship between small fires and months.Through the graph, I found that the main time of fires is in August and September of each year, and the distribution is very even every week. It is possible that the probability of fires on Sundays is relatively high.I think the cause of the fire in August and September may be related to the relatively dry air at that time









Fit The Data

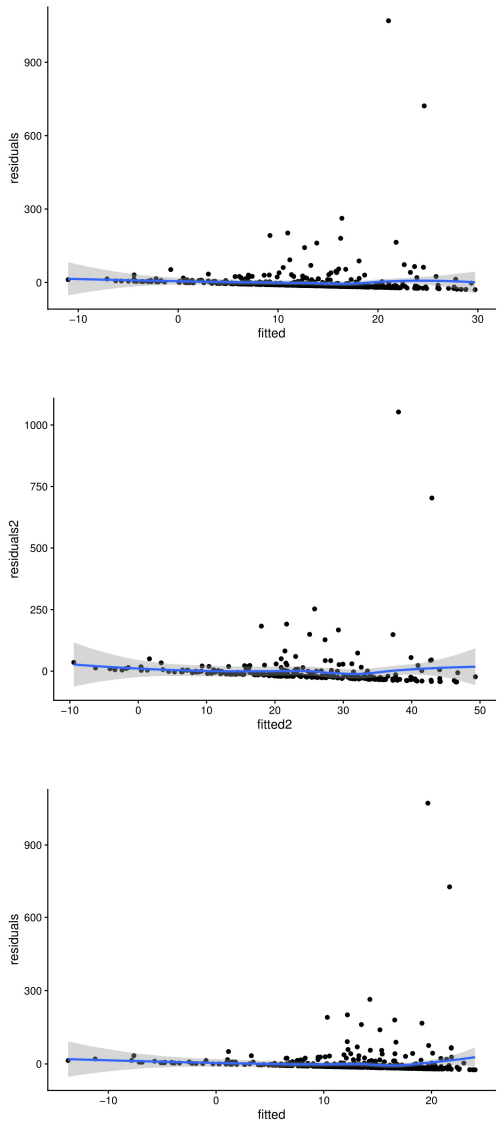
Because I first think that environmental factors such as temperature and humidity are the main causes of fires, I first fit the temperature, humidity, wind and rain. Get the corresponding answer, and then analyze it. Because I found that there are many small fires that did not cause major losses every year, which may

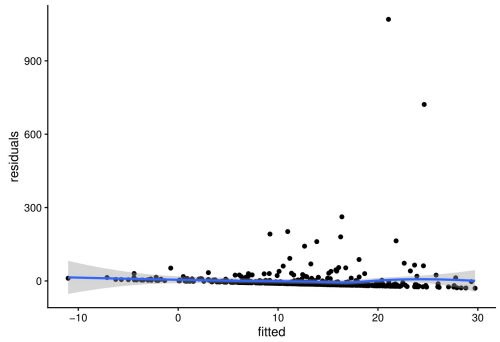
have data impact on some serious fires, so I separately analyzed the fit value of huge fires.

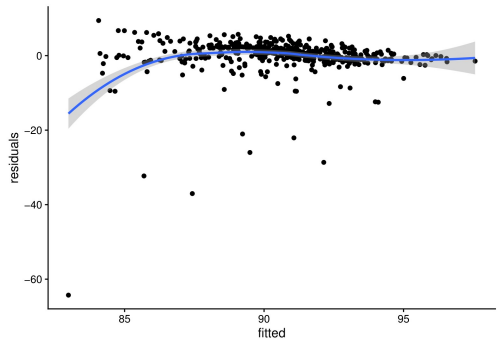
I first analyzed the relationship between the area and the environment from my perspective, and then I found that the temperature is the biggest influence factor on the area, so I started to study the transformation of the model to the temperature change. Although I found that log(temp) has a good effect on area. But because the R value between area and environment is too small to make the multiple linear regression model appear more perfect, so I decided to try the other four unknown values and try to find a suitable model.

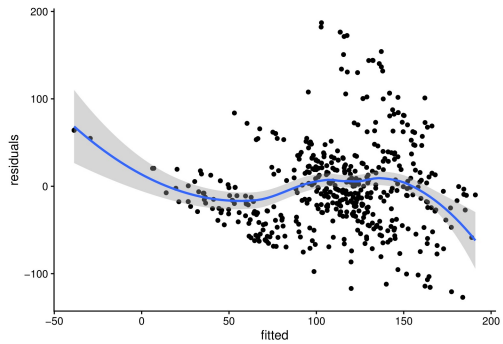
After trying the remaining four data, I found that the R value of ISI was the larger, At the same time relatively flat. Therefore I chose it to explore the relationship between ISI and environmental factors as my new model.

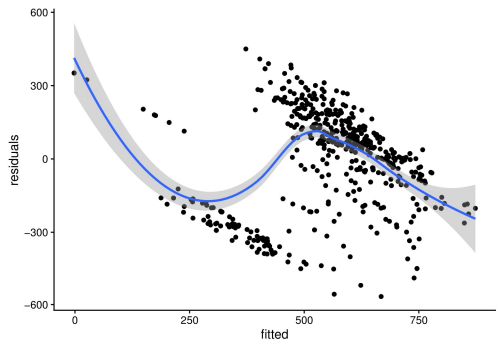
First of all, I will analyze the four major influencing factors and ISI, including log and square, as well as the relationship between the two. At the same time, the results are obtained through AIC analysis. I got my best template by looking at the linear relationship between ISI and other weather factors.ISI ~ temp + RH + wind + rain + log(wind) + temp:wind + temp:rain + wind:rain

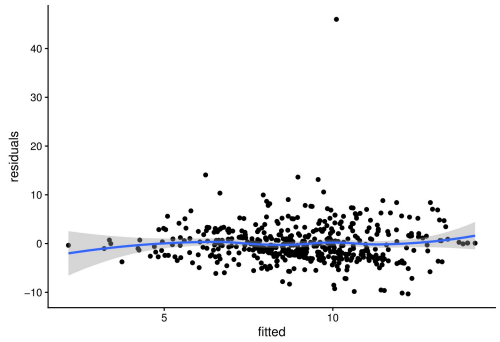


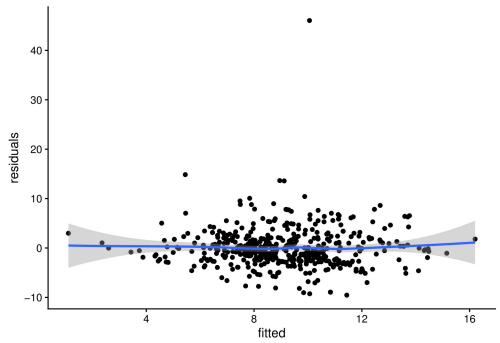

residuals
900
600
300
0
−10
0
10
20
30
fitted

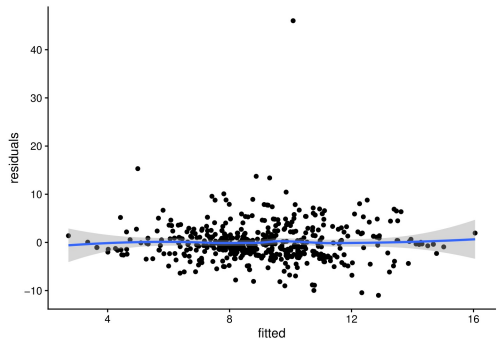

residuals
0
−20
−40
−60
85
90
95
fitted


residuals
200
100
0
−100
−50
0
50
100
150
200
fitted


residuals
600
300
0
−300
−600
0
250
500
750
fitted







Result

visualize the data

It can be seen from the above that there are more fires on Sundays, and there are basically individual fire houses every month, but the more serious ones occur in August and September.

optimal model

The output of the multivariate liner model: log(temp) + RH + wind + rain, but because of the small R value, no more accurate calculations have been made. But this is currently the closest template.
The output of the multivariate non-liner model: temp + RH + wind + rain + log(wind) + temp:wind + temp:rain + wind:rain.The reason using nonlinear regression can fit an enormous variety of curves.
In summary, all the equation fits poorly. I think there is a close relationship between temperature and area. But the linear regression between the four factors and the area is not good. By using other values, it is

found that the relationship between the four climatic factors and ISI is not too strong. It is possible that all the data and the four climatic factors are non-linear. But it is found that temperature is a more important determinant, so detecting temperature can effectively reduce forest fires
Each coefficient estimates the change in the mean response per unit increase in X when all other predictors are held constant.
An interaction is a special property of three or more variables, where two or more variables interact to affect a third variable in a non-additive manner.

Part II

Data description

Introduction

In today's era of pursuing intelligent analysis, people are slowly starting to focus on database analysis because with some analysis databases, people can quickly train computers to make judgments about big data. At the same time these databases have a large number of uncertainties that can make mental networks more intelligent. For example, Google recognizes a large number of images, which makes it now artificial intelligence has the ability to judge items. So these databases are very important for artificial intelligence.

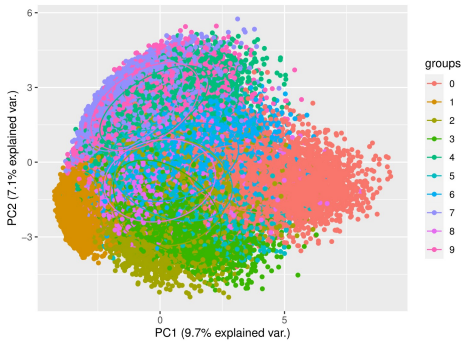
Data Set Information

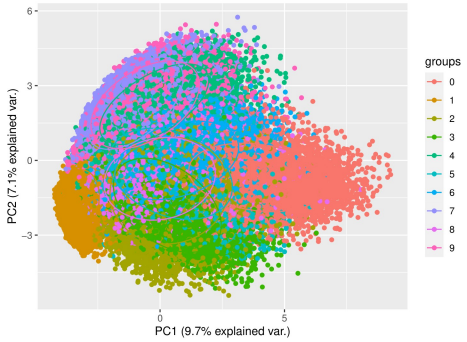
MNIST 's handwriting database is composed of 60,000 training sets and 10,000 test sets, which is a larger subset provided by NIST. The image is 28*28. By using this data for data analysis, we can train the computer's intelligent analysis level. And through intelligent analysis, let us quickly understand the data of MNIST

Model Setup

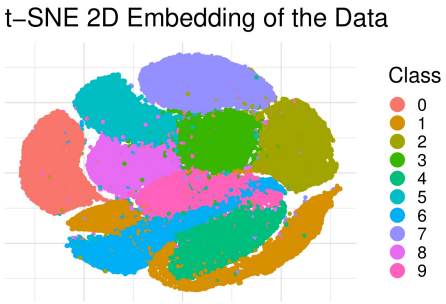
pca and t-SNE

PCA is an unsupervised machine learning algorithm for exploring the structure of high-dimensional data, mainly for data dimensionality reduction, through which more human-understandable features can be discovered and the processing of valuable information about the samples can be accelerated, in addition to applications in visualization.
The t-SNE nonlinear dimensional reduction algorithm finds patterns in the data by identifying observed clusters based on the similarity of data points with multiple features. Essentially it is a dimensional reduction and visualization technique. In addition the output of t-SNE can be used as input features for other classification algorithms. t-SNE can be used on almost all high-dimensional data sets and is widely used in image processing, natural language processing, genomic data, and speech processing.
We use pca and T-sne to visualize our data respectively, so that we do not need to read the whole data to have a very deep understanding of the data, and also greatly deepen our knowledge of the data. Especially for some large data, we can't get a very intuitive understanding, pca and T-sne can make us understand the data quickly.



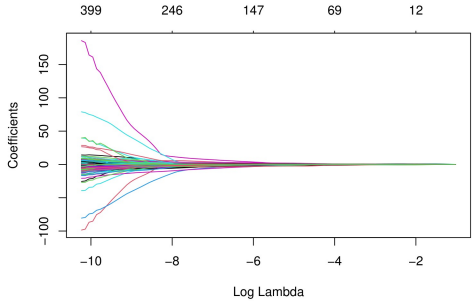


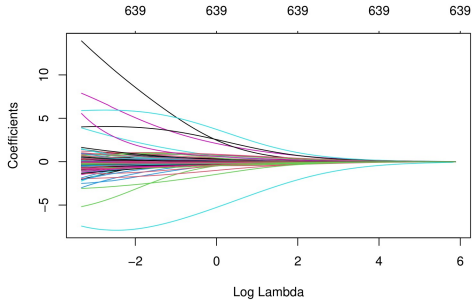

```
## Performing PCA
## Read the 60000 x 50 data matrix successfully!
## OpenMP is working. 1 threads.
## Using no_dims = 2, perplexity = 5.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
##  - point 10000 of 60000
##  - point 20000 of 60000
##  - point 30000 of 60000
##  - point 40000 of 60000
##  - point 50000 of 60000
##  - point 60000 of 60000
## Done in 323.49 seconds (sparsity = 0.000356)!
## Learning embedding...
## Iteration 50: error is 139.621792 (50 iterations in 15.36 seconds)
## Iteration 100: error is 139.621790 (50 iterations in 19.35 seconds)
## Iteration 150: error is 139.477668 (50 iterations in 16.11 seconds)
## Iteration 200: error is 125.196400 (50 iterations in 14.94 seconds)
## Iteration 250: error is 118.600399 (50 iterations in 13.90 seconds)
## Iteration 300: error is 6.376010 (50 iterations in 14.62 seconds)
## Fitting performed in 94.28 seconds.
```

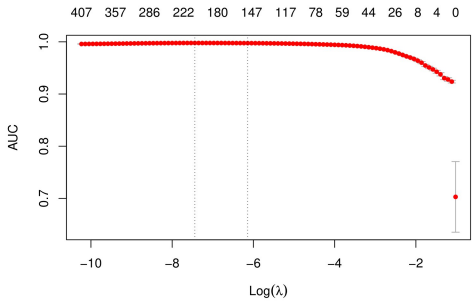


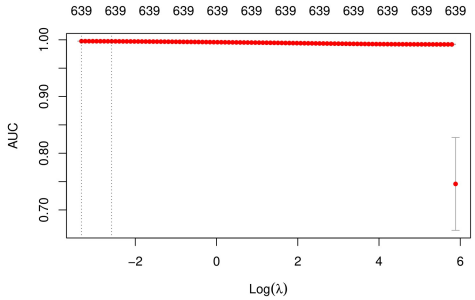
lda qda

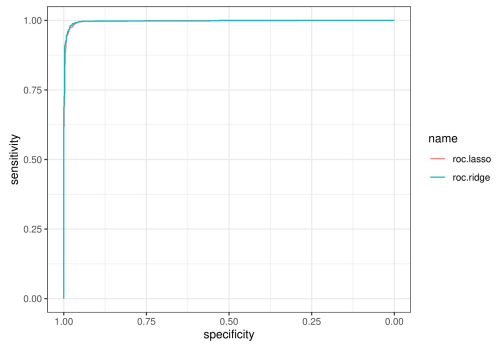
I try to train and test date with label 5 and 6.
I build a binary classifier using LASSO logistic regression and use the predicted probability on test data to calculate the auc.
The LDA and QDA algorithms are based on Bayes' theorem, and their classification method is different from Logistic regression. LDA is used for linear boundaries between classifiers, and QDA is used for finding non-linear boundaries between classifiers. So I use lda and qda to build the model.





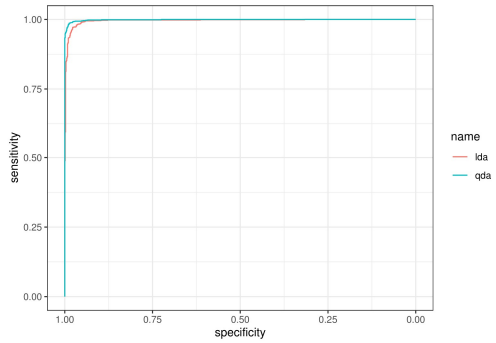






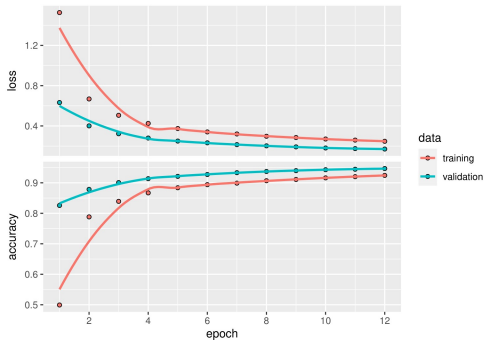

```
## [1] "class"     "posterior" "x"
```

```
## [1] "class"     "posterior"
```



CNN neural network

Because it is with ten classes and each image is 28*28=784, the predicted value of batch size is 128. I used this to build my cnm model and calculate it, and got my cnm record.




```
## Test loss: 0.1694017
```

```
## Test accuracy: 0.9474
```

Result

Through the data we found that mnisy has a high chance of being distinguish, while counting a very high accuracy.

pca and t-SNE

From the graphs we can see that both pca and t-SNE have the ability to graph data. However, because of the presence of some variables, the color of the pca image is messy and cannot be analyzed quickly. But t-SNE has a single color, and the ten areas are well separated, so you can understand the data structure very clearly.

lda qda

lda Area under the curve: 0.9955,which is 99.55% that will be distinguish
qda Area under the curve: 0.9989,which is 99.89% that will be distinguish
lasso Area under the curve: 0.9964,which is 99.64% that will be distinguish
ridge Area under the curve: 0.9972,which is 99.72% that will be distinguish

CNN neural network

Test loss: 0.1904875
Test accuracy: 0.9415

Code

```
library(cowplot)
library(ggplot2)
library(MASS)
library(tidyverse)

date = read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/forest-fires/forestfires.csv
# add rank and sort it
date$month = factor(date$month,levels= c("jan","feb","mar","apr","may","jun","jul","aug","sep","oct","n
date$day = factor(date$day,levels= c("sun","mon","tue","wed","thu","fri","sat"))
date$month = sort(date$month)
date$day = sort(date$day)
plot(date$month,main = "Fire vs Months")
plot(date$day,main = "Fire vs day")
small=date[which(date$area == 0),]
large=date[which(date$area > 0),]
plot(small$month,main = "small Fire Vs Months")
plot(large$month,main = "large Fire Vs Months")

fit <- lm(area ~ temp + RH + wind + rain, data = date)
df = data.frame(residuals = fit$residuals,fitted = fit$fitted.values)
ggplot(aes(x=fitted,y=residuals),data = df) +
  geom_point() +
  geom_smooth()

fit2 = lm(large$area ~ large$temp + large$RH + large$wind + large$rain, date = large)
df2 = data.frame(residuals2 = fit2$residuals,fitted2 = fit2$fitted.values)
ggplot(aes(x=fitted2,y=residuals2),data = df2) +
  geom_point() +
  geom_smooth()

fit3 <- lm(area ~ log(temp) + RH + wind + rain, data = date)
df3 = data.frame(residuals = fit3$residuals,fitted = fit3$fitted.values)
ggplot(aes(x=fitted,y=residuals),data = df3) +
  geom_point() +
  geom_smooth()

fit4 <- lm(area ~ temp^2 + RH + wind + rain, data = date)
df4 = data.frame(residuals = fit4$residuals,fitted = fit4$fitted.values)
ggplot(aes(x=fitted,y=residuals),data = df4) +
  geom_point() +
  geom_smooth()

fit5 <- lm(FFMC ~ temp + RH + wind + rain, data = date)
df5 = data.frame(residuals = fit5$residuals,fitted = fit5$fitted.values)
ggplot(aes(x=fitted,y=residuals),data = df5) +
  geom_point() +
  geom_smooth()

fit6 <- lm(DMC ~ temp + RH + wind + rain, data = date)
df6 = data.frame(residuals = fit6$residuals,fitted = fit6$fitted.values)
ggplot(aes(x=fitted,y=residuals),data = df6) +
```

```
  geom_point() +
  geom_smooth()

fit7 <- lm(DC ~ temp + RH + wind + rain, data = date)
df7 = data.frame(residuals = fit7$residuals,fitted = fit7$fitted.values)
ggplot(aes(x=fitted,y=residuals),data = df7) +
  geom_point() +
  geom_smooth()

fit8 <- lm(ISI ~ temp + RH + wind + rain, data = date)
df8 = data.frame(residuals = fit8$residuals,fitted = fit8$fitted.values)
ggplot(aes(x=fitted,y=residuals),data = df8) +
  geom_point() +
  geom_smooth()

fit9 = lm(ISI~ temp+ RH + wind + rain + temp*RH + temp*wind + temp*rain + RH*wind + RH*rain + wind*rain
df9 = data.frame(residuals = fit9$residuals,fitted = fit9$fitted.values)
ggplot(aes(x=fitted,y=residuals),data = df9) +
  geom_point() +
  geom_smooth()

fit10 = lm(ISI ~ temp + RH + wind + rain + log(wind) + temp:wind + temp:rain + wind:rain, data = date)
df10 = data.frame(residuals= fit10$residuals, fitted = fit10$fitted.values)
ggplot(aes(x = fitted, y= residuals),data=df10) +
  geom_point() +
  geom_smooth()
summary(fit)
summary(fit2)
summary(fit3)
summary(fit4)
summary(fit5)
summary(fit6)
summary(fit7)
summary(fit8)
summary(fit9)
summary(fit10)
AIC = stepAIC(fit9)
summary(AIC)
```

```
library(keras)
library(glmnet)
library(Rtsne)
library(ggbiplot)
library(Rtsne)
library(ggplot2)
library(MASS)
library(pROC)
library(ISLR)

mnist <- dataset_mnist()
#pca and stone
x.train <- mnist$train$x
y.train <- as.factor(mnist$train$y)
```

```r
x.test <- mnist$test$x
y.test <- as.factor(mnist$test$y)
x.train <- array_reshape(x.train, c(nrow(x.train), 784))
x.test <- array_reshape(x.test, c(nrow(x.test), 784))
x.train <- x.train / 255
x.test <- x.test / 255

#pca
mnist.pca = princomp(x.train)
mnist.pca.scaled = princomp(x.train, cor=0)
ggbiplot(mnist.pca,groups = as.factor(y.train),ellipse = TRUE,obs.scale = 1,var.scale = 1, var.axes=F)
ggbiplot(mnist.pca.scaled,groups = as.factor(y.train),ellipse = TRUE,obs.scale = 1,var.scale = 1,var.ax
#t-SNE
mnist.tsne <- Rtsne(x.train, dims = 2, perplexity=5, verbose=TRUE, max_iter = 300)
embedding <- as.data.frame(mnist.tsne$Y)
embedding$Class <- as.factor(y.train)
ggplot(embedding, aes(x=V1, y=V2, color=Class)) +
  geom_point(size=1.25) +
  guides(colour = guide_legend(override.aes = list(size=6))) +
  xlab("") + ylab("") +
  ggtitle("t-SNE 2D Embedding of the Data") +
  theme_light(base_size=20) +
  theme(strip.background = element_blank(),
  strip.text.x = element_blank(),
  axis.text.x = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks = element_blank(),
  axis.line = element_blank(),
  panel.border = element_blank())



#lda qda
x.train = mnist$train$x[which(mnist$train$y%in% c("5", "6")),,]
y.train = as.factor(mnist$train$y[which(mnist$train$y%in% c("5", "6"))])
x.test = mnist$test$x[which(mnist$test$y%in% c("5", "6")),,]
y.test = as.factor(mnist$test$y[which(mnist$test$y%in% c("5", "6"))])

x.train = array_reshape(x.train, c(nrow(x.train), 784))/255
x.test = array_reshape(x.test, c(nrow(x.test), 784))/255

fit.lasso = glmnet(x.train, y.train, family="binomial", alpha=1)
fit.ridge = glmnet(x.train, y.train, family="binomial", alpha=0)
plot(fit.lasso, xvar="lambda")
plot(fit.ridge, xvar="lambda")

cv.lasso= cv.glmnet(x.train, y.train, type.measure="auc", alpha=1,family="binomial")
cv.ridge= cv.glmnet(x.train, y.train, type.measure="auc", alpha=0,family="binomial")
plot(cv.lasso)
plot(cv.ridge)
##plot
yhat.lasso = predict(fit.lasso, s=cv.lasso$lambda.1se, newx=x.test, type="response")
yhat.ridge = predict(fit.ridge, s=cv.lasso$lambda.1se, newx=x.test, type="response")
```

```
fit_roc = roc(y.test,yhat.lasso)
fit_roc2 = roc(y.test,yhat.ridge)
fit_roc
fit_roc2
ggroc(list(roc.lasso = fit_roc,roc.ridge = fit_roc2))+theme_bw()

##part b
mnist.pca = princomp(x.train, cor=0)
z.train = as.data.frame(mnist.pca$scores[, 1:100])
z.train$y= y.train
z.test = as.data.frame(x.test %*% mnist.pca$loadings[,1:100])

lda.fit=lda(y~., data=z.train)
lda.fit
lda.pred=predict(lda.fit, as.data.frame(z.test))
names(lda.pred)
lda.class = lda.pred$class
lda_roc = roc(y.test, lda.pred$posterior[,2])
lda_roc

qda.fit = qda(y~., data=z.train)
qda.fit
qda.pred=predict(qda.fit, as.data.frame(z.test))
names(qda.pred)
qda.class = qda.pred$class
qda_roc = roc(y.test, qda.pred$posterior[,2])
qda_roc

ggroc(list(lda=lda_roc, qda=qda_roc)) + theme_bw()

#cnm
batch_size <- 128
num_classes <- 10
epochs <- 12

# Input image dimensions
img_rows <- 28
img_cols <- 28

# The data, shuffled and split between train and test sets
mnist <- dataset_mnist()
x_train <- mnist$train$x
y_train <- mnist$train$y
x_test <- mnist$test$x
y_test <- mnist$test$y

# Redefine  dimension of train/test inputs
x_train <- array_reshape(x_train, c(nrow(x_train), img_rows, img_cols, 1))
x_test <- array_reshape(x_test, c(nrow(x_test), img_rows, img_cols, 1))
input_shape <- c(img_rows, img_cols, 1)

# Transform RGB values into [0,1] range
x_train <- x_train / 255
```

```
x_test <- x_test / 255

cat('x_train_shape:', dim(x_train), '\n')
cat(nrow(x_train), 'train samples\n')
cat(nrow(x_test), 'test samples\n')

# Convert class vectors to binary class matrices
y_train <- to_categorical(y_train, num_classes)
y_test <- to_categorical(y_test, num_classes)

# Define model
model <- keras_model_sequential() %>%
 layer_conv_2d(filters = 16, kernel_size = c(2, 2), input_shape = c(28, 28, 1),strides = 2) %>% #CONV 1

  layer_max_pooling_2d(pool_size = c(2, 2), strides = 2) %>% #POOL 1

  layer_conv_2d(filters = 32, kernel_size = c(2, 2),strides = 1) %>% #CONV 2

  layer_max_pooling_2d(pool_size = c(2, 2), strides = 2) %>% #POOL 2

  layer_flatten() %>%
  layer_dense(units = 128, activation = 'relu') %>%
  layer_dropout(rate = 0.5) %>%
  layer_dense(units = num_classes, activation = 'softmax')

model %>% compile(
  loss = loss_categorical_crossentropy,
  optimizer = optimizer_adagrad(),
  metrics = c('accuracy')
)
histroy <- model %>% fit(
  x_train, y_train,
  batch_size = batch_size,
  epochs = epochs,
  validation_split = 0.2
)
#d
plot(histroy)
scores <- model %>% evaluate(
  x_test, y_test, verbose = 0
)

# Output metrics
cat('Test loss:', scores[[1]], '\n')
cat('Test accuracy:', scores[[2]], '\n')
model %>% evaluate(x_test, y_test,verbose = 0)
y_pred = model %>% predict(x_test) %>% k_argmax()
#sum(diag(confusionMatrix(y_pred,y_test)))/length(y_test)
```