

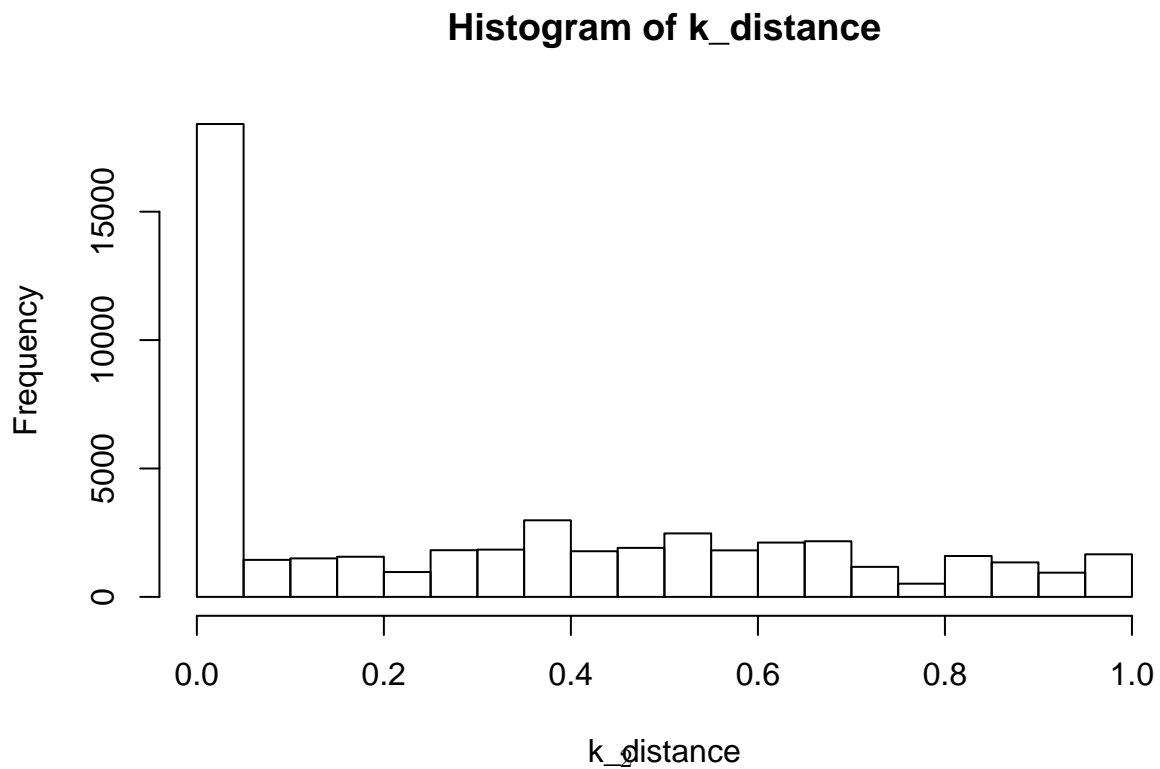
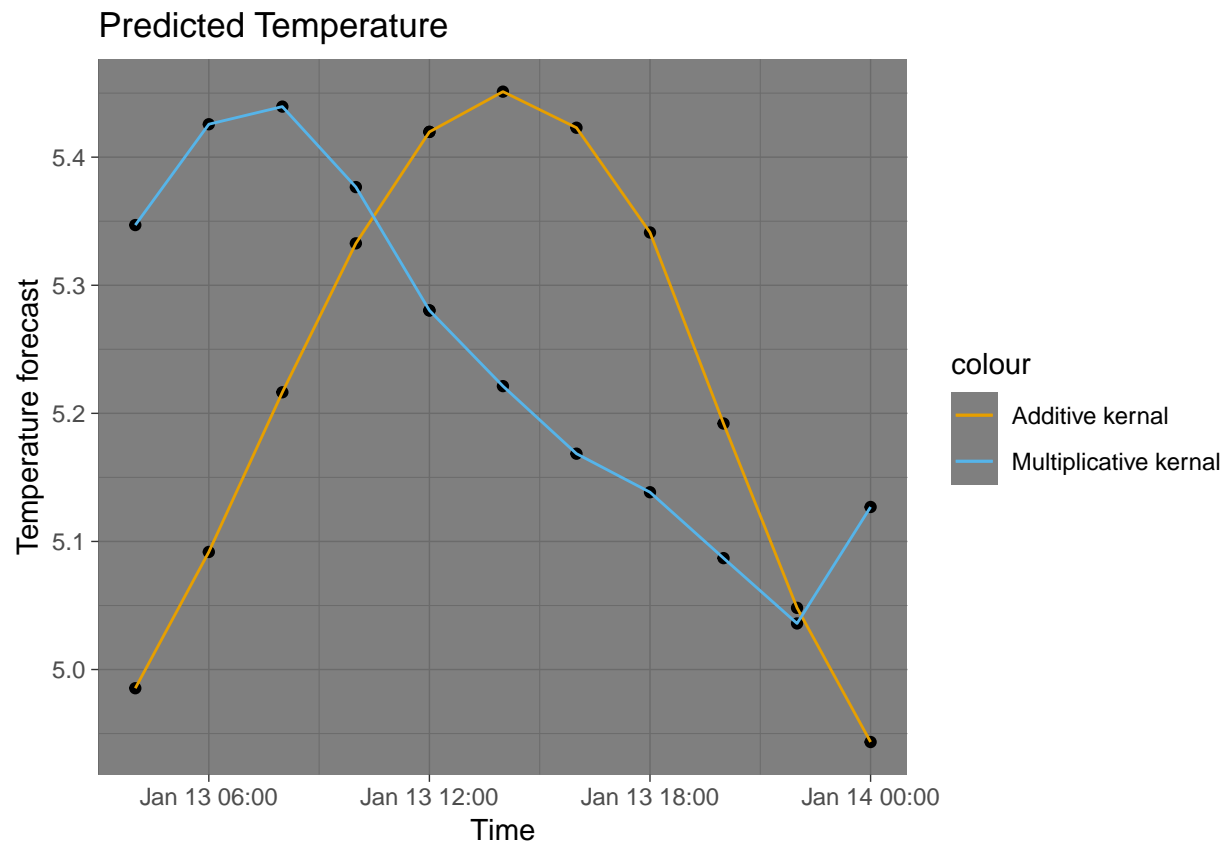
Machine Learning Lab 3

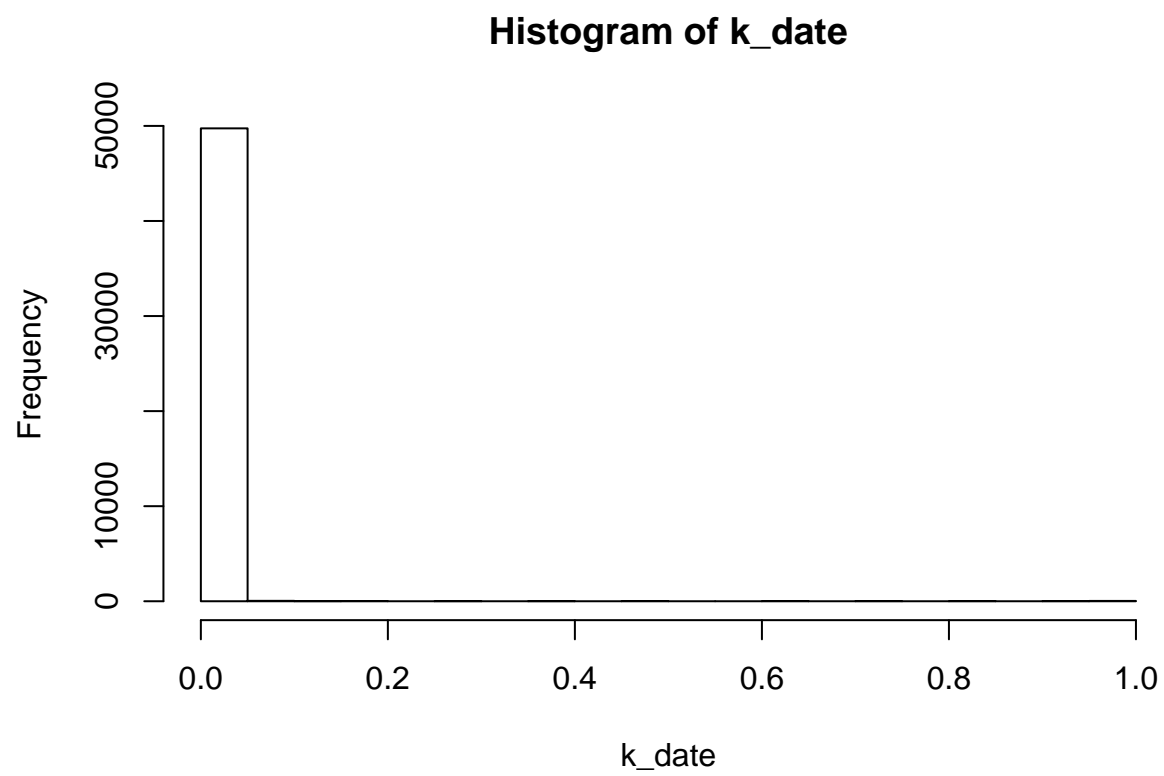
Omkar Bhutra(omkbh878)

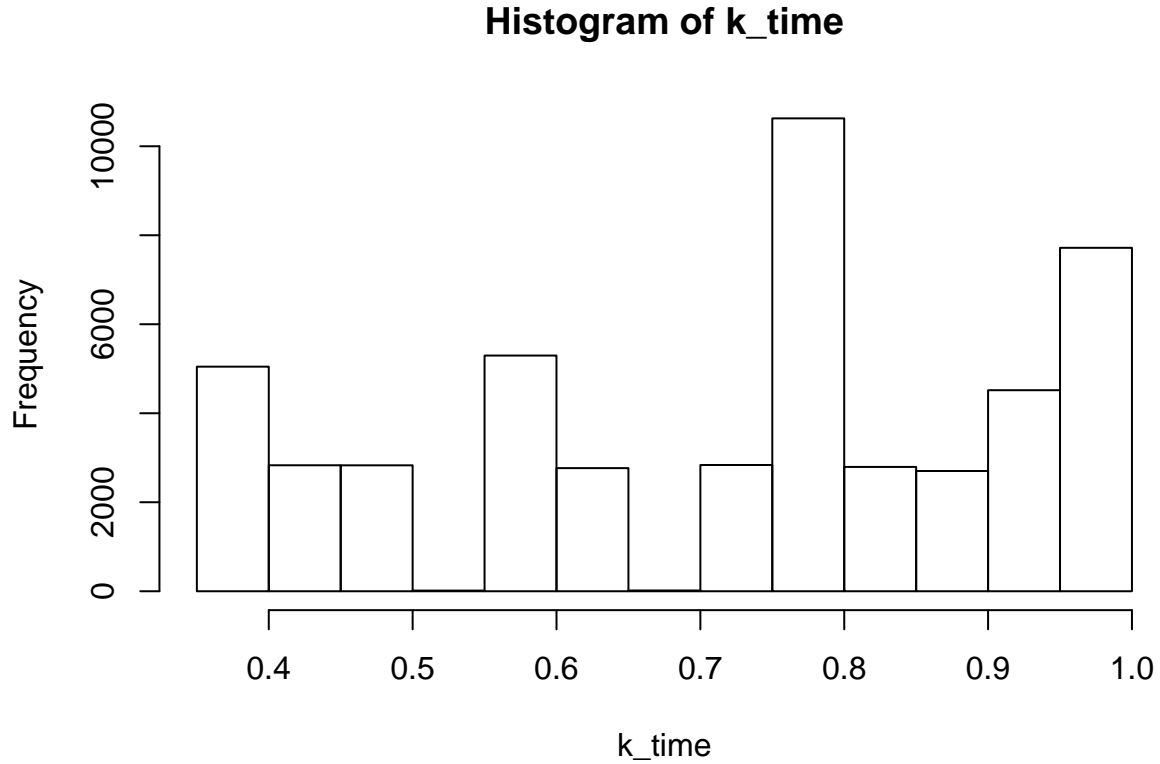
16 December 2018

Assignment 1.

Kernal methods:







The width of the distance kernel of 25Kms is chosen, Since Sweden is close to the arctic circle the temperature fluctuations remain uniform over large distances.

The width of the distance for days is taken as 7 as people generally talk about the weeks weather. It is noticeably uniform in any given week.

The width of the distance for the hours is taken as 12 as the temperature during any given day is defined by night and day which is 2 groups out of 24.

If we look at the histograms above, we see the histograms over the three kernels. As we can see we have in all three kernels values that are ranging from 0 to 1, but most of them are zero, and very few are close to one. This is reasonable since we are looking at distances only relatively close to our own position, since those are the position which are mostly related to our own weather. We are also looking at days, relatively close to the chosen day, since more than a months difference can give a major difference to air temperature. We are also looking at hours closely related to our chosen one, since just a small amount of hours can matter a lot for temperature.

The choice of kernel widths are sensitive means that more weight is given to closer points. A point can be close w.r.t. any of the 3 variables (distance, date, time). In short, all plots below show that the Kernel values decrease when the 2nd point is more distant (the 1st point is fixed). I.e. the Kernels are sensitive. They gives higher weights to closer points and smaller weights to distant points. Below, some examples with Unsuitable widths are shown. We can see that the Kernel values will diminish too fast or too slow. $h_distance = 0.25$, $h_date = 1$, $h_time = 0.25$ these kernel widths will make the kernel value diminish too fast and hence unsuitable. $h_distance = 1000$, $h_date = 200$, $h_time = 12$ these kernel widths will make the kernel value diminish too slowly and hence also unsuitable.

Assignment 2

Support vector machines

```
##          Predicted svm
## Actual Test nonspam spam
##    nonspam    1346    56
##    spam       155    744
```

```
## [1] "The misclassification rate is 0.0916992611907866"
```

```
##          Predicted svm
## Actual Test nonspam spam
##    nonspam    1340    62
##    spam       131    768
```

```
## [1] "The misclassification rate is 0.0838765754019991"
```

```
##          Predicted svm
## Actual Test nonspam spam
##    nonspam    1336    66
##    spam       125    774
```

```
## [1] "The misclassification rate is 0.0830073880921338"
```

The Misclassification error rates of the models are 0.0916, 0.0838 and 0.0830 for the models with width of 0.05 as the hyperparameter for the kernel of type Radial Basis. C is the cost of constraint violation. This is the 'C' Constant of the regularisation term in the Lagrange formulation. The purpose of this is to behave as a penalty term for violation of the rules of the classification so as to not overfit the model.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(geosphere)
library(kernlab)
library(ggplot2)
library(lubridate)
set.seed(1234567890)
stations <- read.csv("stations.csv")
temps <- read.csv("temps50k.csv")
st <- merge(stations, temps, by="station_number")
rm(stations, temps)
st$time <- as.POSIXct(st$time, format="%H:%M:%S")
a <- 58.4166
b <- 15.6333
hdist <- 250000
hdate <- 7
htime <- 12
date <- "2001-11-04"
timeseq <- c("04:00:00", "06:00:00", "08:00:00", "10:00:00", "12:00:00", "12:00:00", "14:00:00", "16:00:00",
```

```

timeseq <- as.POSIXct(timeseq,format="%H:%M:%S")

coords<-cbind(st$longitude, st$latitude)
ykernalsum<-c()
ykernalprod<-c()
final<-c()
for (i in 1:length(timeseq)){
  h_distance<-(distHaversine(coords,c(b,a))/hdist)
  k_distance<-exp(-(h_distance)^2)
  h_date <- abs(as.numeric(as.Date(st$date) - as.Date(date)))
  h_date[h_date > 182] <- 365 - h_date[h_date > 182]
  h_date <- h_date /hdate
  k_date <- exp(-(h_date)^2)
  h_time <- as.numeric(difftime(time1 = st$time ,time2= timeseq[i], units = "hours"))
  h_time <- abs(h_time)
  h_time[h_time > 12] = 24 - h_time[h_time > 12]
  h_time <- h_time / htime
  k_time <- exp(-(h_time)^2)
  ksum <- k_distance + k_date + k_time
  ykernalsum[i] <- sum(ksum*st$air_temperature) / sum(ksum)
  kprod <- k_distance * k_date * k_time
  ykernalprod[i] <- sum(kprod*st$air_temperature) / sum(kprod)
  df <- data.frame(Time = timeseq[i], ykernalsum = ykernalsum[i], ykernalprod = ykernalprod[i])
  final <- rbind(final, df)
}

p1 <- ggplot(final, aes(Time)) +
  geom_point(aes(y = ykernalsum)) +
  geom_point(aes(y = ykernalprod)) +
  geom_line(aes(y = ykernalsum, color = "Additive kernal")) +
  geom_line(aes(y = ykernalprod, color = "Multiplicative kernal")) +
  scale_color_manual(values=c("#E69F00", "#56B4E9")) +
  ylab("Temperature forecast") +
  theme_dark()+ggtitle("Predicted Temperature")
p1

hist(k_distance)
hist(k_date)
hist(k_time)

set.seed(1234567890)
data(spam)
n<-dim(spam)[1]
id<-sample(1:n,floor(n*0.5))
train<-spam[id,]
test<-spam[-id,]
xtrain<-as.matrix(train[,-58])
ytrain<-as.matrix(train[,58])
xtest<-as.matrix(test[,-58])
ytest<-as.matrix(test[,58])
xtrain2<-train[,-58]
svmmodel0.5<- ksvm(xtrain, ytrain, kernel="rbfdot",kpar=list(sigma=0.05),C=0.5)
svmmodel1<- ksvm(xtrain, ytrain, kernel="rbfdot",kpar=list(sigma=0.05),C=1)

```

```

svmmmodel5<- ksvm(xtrain, ytrain, kernel="rbfdot",kpar=list(sigma=0.05),C=5)

svmpredict0.5<-predict(svmmmodel0.5, xtest, type="response")
svmpredict1<- predict(svmmmodel1, xtest, type="response")
svmpredict5<- predict(svmmmodel5, xtest, type="response")

consvm0.5<- table(ytest, svmpredict0.5)
names(dimnames(consvm0.5)) <- c("Actual Test", "Predicted svm")
consvmres0.5<-caret::confusionMatrix(consvm0.5)
consvm0.5
mse3.1<-(1-(sum(diag(consvm0.5))/sum(consvm0.5)))
paste("The misclassificaiton rate is",mse3.1)

consvm1<- table(ytest, svmpredict1)
names(dimnames(consvm1)) <- c("Actual Test", "Predicted svm")
consvmres1<-caret::confusionMatrix(consvm1)
consvm1
mse3.2<-(1-(sum(diag(consvm1))/sum(consvm1)))
paste("The misclassificaiton rate is",mse3.2)

consvm5<- table(ytest, svmpredict5)
names(dimnames(consvm5)) <- c("Actual Test", "Predicted svm")
consvmres5<-caret::confusionMatrix(consvm5)
consvm5
mse3.3<-(1-(sum(diag(consvm5))/sum(consvm5)))
paste("The misclassificaiton rate is",mse3.3)

```