Predict response times for LAFD

Group name: K3

Members: Jingun Kwon: stylenote91@gmail.com
Junhyuk Jang: junhyuk.jang0@gmail.com
Myungwoo Nam: myungwoo0221@gmail.com

In this project, we were asked to predict response times for the Los Angeles Fire Department. We analyzed the models in order to specify which factors affect the prediction. We were provided with data of the last four years for all fire station in LA. The data set composed of training data, which have 2774370 observations and 11 variables, and the test data, which have 530352 observations and 10 variables.

When it comes to data cleaning we used three steps. First of all, we dropped some variables named row id, Incident ID, Emergency and Dispatch code from the the training data because we found no distinct pattern in those three variables. Moreover, Year and First.in.District variables are changed to factor. Dispatch.Sequence and Incident.Creation.Time..GMT are changed to numeric. For the testing dataset, we dropped Incident ID and Emergency.Dispatch.Code because we thought that they will not affect our prediction as much. In addition, we factorized variables Year, Dispatch.Status, and First.in.District and numericized Incident.Creation.Time..GMT. Because one of Unit.Type levels “Rp-Rehap Plug Buggy” appeared only once in our test data we changed this level into RA-ALS Rescue Ambulance” under the assumption that level “Rp-Rehap Plug Buggy” has no significance.

Secondly, we removed outliers from the data. We did this because we thought if we remove outliers, which are not related to the data, it will allow us to get more well-fitted model. We defined outliers with the significance level of 0.01 and cleaned rows which had

value either greater than 0.99 quantile or smaller than 0.01 quantile. We selected four variables named Incident.Creation.Time..GMT, year,Dispapatch.Sequence,PPE.Level and cleaned their outliers.

Lastly, we used the log transformation for the variable Incident.Creation.Time..GMT because this variable was skewed-right based on the summary statistics(mean is greater than the median). We assumed that the model achieved by normally distributed data would give us the better result.

	year	First.in.District	Dispatch.Sequence	Dispatch.Status	Unit.Type	PPE.Level	Incident.Creation.Time..GMT.	elapsed_time
1	2013	93	1	QTR	T - Truck	EMS	82830	263
2	2013	93	2	QTR	E - Engine	EMS	82830	263
3	2013	93	3	RAD	RA - ALS Rescue Ambulance	EMS	82830	367
4	2013	12	1	QTR	E - Engine	EMS	85834	248
6	2013	12	3	RAD	E - Engine	EMS	85834	216
7	2013	12	4	ONS	RA - ALS Rescue Ambulance	EMS	85834	203

	row.id	year	First.in.District	Dispatch.Sequence	Dispatch.Status	Unit.Type	PPE.Level	Incident.Creation.Time..GMT.
1	1792206	2013	11	1	QTR	E - Engine	EMS	11904
2	1843889	2013	11	3	RAD	RA - ALS Rescue Ambulance	EMS	11904
3	3150413	2013	63	1	QTR	T - Truck	EMS	53256
4	2943256	2013	63	2	QTR	E - Engine	EMS	53256
5	327355	2013	63	3	QTR	RA8xx - BLS Rescue Ambulance	EMS	53256
6	2573461	2013	104	1	QTR	E - Engine	EMS	73978

This is train and test set after cleaning and transforming.

When it comes to statistical learning method, we used linear regression combined with polynomial with degree 5 algorithms to implement the prediction. By using this method, we achieved MSE of 1432495.47090. Even though we have tried gradient boosting and random forest with the same cleaned dataset, we got higher MSEs than using the algorithm of

linear regression with polynomial degree five. In regards to polynomial degree selection, we found out more than polynomial degree of five is not significant. So, we decided to use degree 5 for our polynomial regression.

We observed that variables Dispatch.Sequence, and Incident.Creation.Time..GMT. had a curvilinear relationship with each other. In this situation adding polynomial regression into our model would give us a better fit. By using the model linear regression combined with polynomial with degree 5 we can not only accurately approximate the relationship but also avoid over-fitting. That is why we got a superior result with this model compared to the models with random forest, logistic regression and gradient boosting.

CODE

```
test<- read.csv("~/Downloads/testing.without.response.txt")

train <- read.csv("~/Downloads/lafdtraining updated (1).csv")

library(class)

library(FNN)

library(KRLS)

attach(train)

attach(test)

train=train[,-c(1:2)]

train=train[complete.cases(train),]

train <- subset(train,!(train$elapsed_time[Incident.Creation.Time..GMT.] >
quantile(train$elapsed_time[Incident.Creation.Time..GMT.], probs=c(.01, .99))[2] |
train$elapsed_time[Incident.Creation.Time..GMT.] <
quantile(train$elapsed_time[Incident.Creation.Time..GMT.], probs=c(.01, .99))[1]) )

train <- subset(train,!(train$elapsed_time[year] > quantile(train$elapsed_time[year],
probs=c(.01, .99))[2] | train$elapsed_time[year] < quantile(train$elapsed_time[year],
probs=c(.01, .99))[1]) )

train <- subset(train,!(train$elapsed_time[Dispatch.Sequence] >
quantile(train$elapsed_time[Dispatch.Sequence], probs=c(.01, .99))[2] |
train$elapsed_time[Dispatch.Sequence] < quantile(train$elapsed_time[Dispatch.Sequence],
probs=c(.01, .99))[1]) )
```

```
train <- subset(train,!(train$elapsed_time[PPE.Level] >
quantile(train$elapsed_time[PPE.Level], probs=c(.01, .99))[2] |
train$elapsed_time[PPE.Level] < quantile(train$elapsed_time[PPE.Level],
probs=c(.01, .99))[1]) )


train$year=as.factor(train$year)
train$First.in.District=as.factor(train$First.in.District)
train$Dispatch.Sequence=as.numeric(train$Dispatch.Sequence)
train$Incident.Creation.Time..GMT.=as.numeric(train$Incident.Creation.Time..GMT.)


train=train[,-3]
View(train)
str(train$Dispatch.Status)
str(test$Dispatch.Status)


test=test[,-c(2)]
test$year=as.factor(test$year)
test$Dispatch.Status=as.factor(test$Dispatch.Status)
test$First.in.District=as.factor(test$First.in.District)
test$Incident.Creation.Time..GMT.=as.numeric(test$Incident.Creation.Time..GMT.)
```

```
test=test[,-4]

test[is.na(test$Dispatch.Sequence),]$Dispatch.Sequence=

mean(test$Dispatch.Sequence,na.rm = T)

test[test$Unit.Type=="RP - Rehab Plug Buggy",]$Unit.Type="RA - ALS Rescue

Ambulance"

head(train)

head(test)

#5 factors , 2 continuous

sub_index=sample(1:nrow(train),size = length(train$year), replace=F)

subdat=train[sub_index,]

subtrain_index=sample(1:nrow(subdat),size = nrow(subdat)/2,replace = F)

subtrain=subdat[subtrain_index,]

subtest=subdat[-subtrain_index,]

head(subdat)

ptm<-proc.time()

fitpoly=lm(elapsed_time~year+Dispatch.Status+Unit.Type+PPE.Level+First.in.District


+polym(Dispatch.Sequence,log(Incident.Creation.Time..GMT.),degree=5,raw=T),data=subd

at)

proc.time()-ptm

lmpre=predict(fitpoly,test)
```

```
submit<-data.frame(row.id = test$row.id, elapsed_time = lmpre)

write.csv(submit,file = "gg.csv",row.names = F)

gg <- read.csv("~/gg.csv")

names(gg)[2] <- paste("prediction")

write.csv(gg,file = "gg.csv",row.names = F)
```