PRACTIUM 2 – MSDS696-FALL18

Week: 8 – Final Report

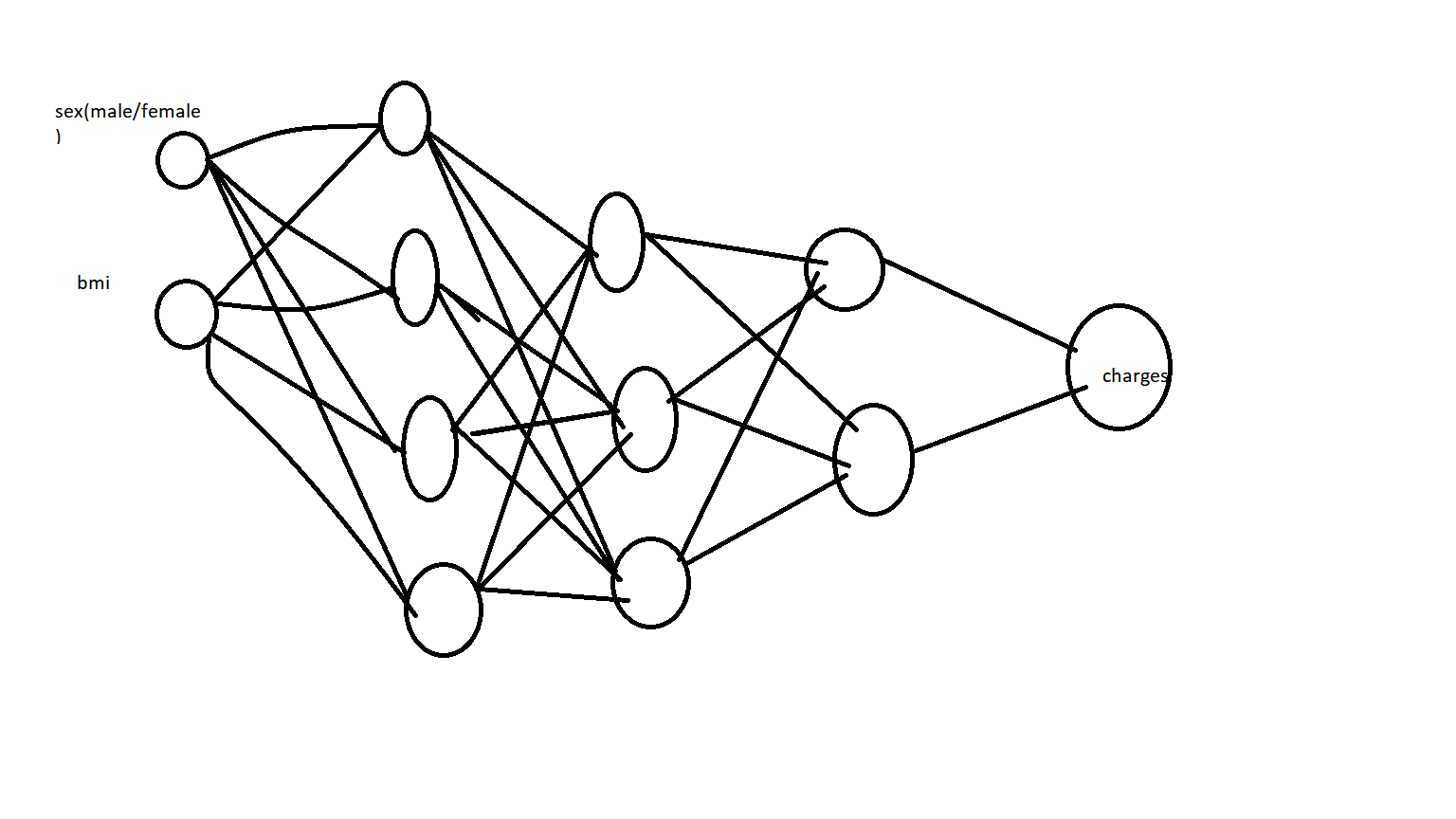
Name: Phong Nguyen

PROJECT TITLE: Deep Learning with TensorFlow: Applied Regression and Neural Network to predict the Health Insurance Cost.

PROJECT SUMMARY: Deep Learning is very subject. In this project, Deep Learning is applied to predict the healthcare insurance price. The technique in Deep Learning is applied TensorFlow. The Methods are Applied Regression and Neural Network. The raw data is downloaded from Kaggle.com [3].

**I: MILESTONES:**

* Overall, collect data and applied some basic analyze to find the relationship between variables in the dataset. There are three variables are used, “charges”, “age” and “bmi” for this project.
* There are two methods are used in this project. It’s Regression and Deep Neural Network. Regression is applied under the frame work of Generalized Additive Models (GAM). In GAM, Lamda-Mu-Sigma is a method that applied to calculate and predict the results.
* This is a big milestone for this project. Because TensorFlow is the main step entitle the project.



**II: METHODOLOGIES:**

LAMDA-MU-SIGMA method is used to applied for the Regression. General Additive Models is a frame work for this regression.

DEEP NEURAL NETWORK method is a second methods using to compare. The black-end for running DEEP NEURAL NETWROK is Tensorflow Deep Learning.

**III: EXPLORE THE DATA AND CREATE THE MODELS:**

Raw data for both male and female

> MFdata=caret::createDataPartition(y=mydata$sex, p=400/600,list=FALSE)

> trainset=mydata[MFdata,]

> testset=mydata[-MFdata,]

> head(trainset)

age sex bmi children smoker region charges

1 19 female 27.90 0 yes southwest 16884.924

3 28 male 33.00 3 no southeast 4449.462

5 32 male 28.88 0 no northwest 3866.855

6 31 female 25.74 0 no southeast 3756.622

7 46 female 33.44 1 no southeast 8240.590

10 60 female 25.84 0 no northwest 28923.137

> trainset%>%gather(charges,key="Outcome",value="Value")%>%

+ ggplot(aes(x=age,y=Value))+

+ geom\_point(aes(col=sex),alpha=0.2)+

+ geom\_smooth(aes(fill=sex,col=sex),alpha=0.5)+

+ theme\_bw()+

+ facet\_grid(Outcome~sex,scales = "free")+

+ scale\_fill\_manual(values=c("red","blue"))+

+ scale\_color\_manual(values=c("red3","blue3"))

`geom\_smooth()` using method = 'loess' and formula 'y ~ x'

> Mdata%>%gather(bmi,age,key="Predictor",value="Value")%>%

+ ggplot(aes(x=Value,y=charges))+

+ geom\_point(shape=21,col="red",fill="red",alpha=0.3)+

+ geom\_smooth(alpha=0.5,fill="red",col="red4")+

+ facet\_wrap(~Predictor,ncol=2,scales="free")+

+ theme\_bw()

`geom\_smooth()` using method = 'loess' and formula 'y ~ x'

> trainset%>%gather(charges,key="Outcome",value="Value")%>%

+ ggplot(aes(x=bmi,y=Value))+

+ geom\_point(aes(col=sex),alpha=0.2)+

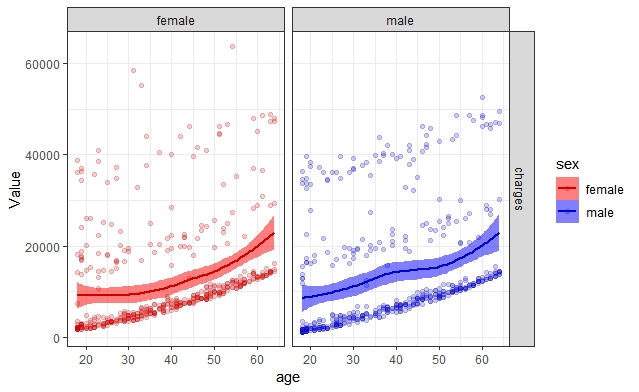
+ geom\_smooth(aes(fill=sex,col=sex),alpha=0.5)+

+ theme\_bw()+

+ facet\_grid(Outcome~sex,scales = "free")+

+ scale\_fill\_manual(values=c("red","blue"))+

+ scale\_color\_manual(values=c("red3","blue3"))



Create a separated data for Male:

> library(dplyr)

> library(MASS)

> Mdata=filter(mydata,sex=="male")%>%dplyr::select(-sex)

> Fdata=filter(mydata,sex=="female")%>%dplyr::select(-sex)

> summary(Mdata)

age bmi children smoker region charges

Min. :18.00 Min. :15.96 Min. :0.000 no :517 northeast:163 Min. : 1122

1st Qu.:26.00 1st Qu.:26.41 1st Qu.:0.000 yes:159 northwest:161 1st Qu.: 4619

Median :39.00 Median :30.69 Median :1.000 southeast:189 Median : 9370

Mean :38.92 Mean :30.94 Mean :1.115 southwest:163 Mean :13957

3rd Qu.:51.00 3rd Qu.:34.99 3rd Qu.:2.000 3rd Qu.:18990

Max. :64.00 Max. :53.13 Max. :5.000 Max. :62593

> MTraindata=caret::createDataPartition(y=Mdata$charges, p=400/505,list=FALSE)

> Mtraindata=Mdata[MTraindata,]

> Mtestdata=Mdata[-MTraindata,]

> head(Mtraindata)

age bmi children smoker region charges

1 18 33.770 1 no southeast 1725.552

2 28 33.000 3 no southeast 4449.462

3 33 22.705 0 no northwest 21984.471

4 32 28.880 0 no northwest 3866.855

5 37 29.830 2 no northeast 6406.411

6 25 26.220 0 no northeast 2721.321

> head(Mtestdata)

age bmi children smoker region charges

7 23 34.40 0 no southwest 1826.843

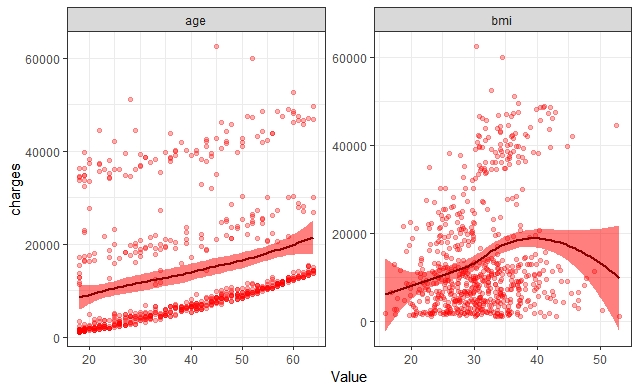
9 19 24.60 1 no southwest 1837.237

16 31 36.30 2 yes southwest 38711.000

25 38 37.05 1 no northeast 6079.672

28 48 28.00 1 yes southwest 23568.272

32 43 27.36 3 no northeast 8606.217



Create a separated data for female:

> FTraindata=caret::createDataPartition(y=Fdata$charges, p=325/400,list=FALSE)

> Ftraindata=Mdata[FTraindata,]

> Ftestdata=Mdata[-FTraindata,]

> Fdata=filter(mydata,sex=="female")%>%dplyr::select(-sex)

>

> head(Fdata)

age bmi children smoker region charges

1 19 27.90 0 yes southwest 16884.924

2 31 25.74 0 no southeast 3756.622

3 46 33.44 1 no southeast 8240.590

4 37 27.74 3 no northwest 7281.506

5 60 25.84 0 no northwest 28923.137

6 62 26.29 0 yes southeast 27808.725

Create LMS Models: Male data

> library(gamlss)

> numCores<-detectCores()

> MaleModel=gamlss(data=Mtraindata,

+ charges~1,

+ sigma.formula = charges~1,

+ nu.formula = charges~1,

+ family=BCCG(mu.link="log"),

+ trace=FALSE,

+ parallel="multicore",

+ ncpus = numCores)

> MaleModel1=stepGAICAll.A(MaleModel, scope=list(lower=~1,

+ upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ sigma.scope = list(lower=~1,

+ upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ k=log(length(trainM)),

+ trace=FALSE,

+ parallel="multicore",

+ ncpus = numCores)

---------------------------------------------------

---------------------------------------------------

ERROR: stepGAICAll has failed trying to fit the model for mu forward.

Maybe the mu model is too complicated for the data.

The model given is the final before the crush.

---------------------------------------------------

> MaleModel1=stepGAICAll.A(MaleModel, scope=list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),sigma.scope = list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ k=log(length(Mtraindata)),trace=FALSE,parallel="multicore",ncpus = numCores)

---------------------------------------------------

Start: AIC= 23078.76

charges ~ 1

---------------------------------------------------

Start: AIC= 19361.86

~1

---------------------------------------------------

Start: AIC= 19361.86

~1

---------------------------------------------------

Start: AIC= 19361.86

~1

---------------------------------------------------

Start: AIC= 19361.86

charges ~ poly(bmi, 3) + poly(age, 2) + pb(age)

---------------------------------------------------

Warning messages:

1: In RS() : Algorithm RS has not yet converged

2: In RS() : Algorithm RS has not yet converged

> summary(MaleModel1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Family: c("BCCG", "Box-Cox-Cole-Green")

Call: gamlss(formula = charges ~ poly(bmi, 3) + poly(age, 2) + pb(age), sigma.formula = ~1, nu.formula = ~1, family = BCCG(mu.link = "log"),

data = Mtraindata, trace = FALSE, parallel = "multicore", ncpus = numCores)

Fitting method: RS()

------------------------------------------------------------------

Mu link function: log

Mu Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -3.704e+01 6.063e-02 -610.9 <2e-16 \*\*\*

poly(bmi, 3)1 1.037e-06 1.427e+00 0.0 1

poly(bmi, 3)2 -4.134e-07 1.423e+00 0.0 1

poly(bmi, 3)3 -1.967e-07 1.418e+00 0.0 1

poly(age, 2)1 1.604e-06 1.437e+00 0.0 1

poly(age, 2)2 -6.327e-08 1.417e+00 0.0 1

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------

Sigma link function: log

Sigma Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 3.102e+01 4.287e-07 72356720 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------

Nu link function: identity

Nu Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 5.000e-01 4.287e-07 1166190 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------

NOTE: Additive smoothing terms exist in the formulas:

i) Std. Error for smoothers are for the linear effect only.

ii) Std. Error for the linear terms may not be reliable.

------------------------------------------------------------------

No. of observations in the fit: 544

Degrees of Freedom for the fit: 8.000555

Residual Deg. of Freedom: 535.9994

at cycle: 3

Global Deviance: 19347.53

AIC: 19363.53

SBC: 19397.92

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Create LMS model: Female data

> FemaleModel=gamlss(data=Mtraindata,

+ charges~1,

+ sigma.formula = charges~1,

+ nu.formula = charges~1,

+ family=BCCG(mu.link="log"),

+ trace=FALSE,

+ parallel="multicore",

+ ncpus = numCores)

>

> FemaleModel1=stepGAICAll.A(FemaleModel, scope=list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),sigma.scope = list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ + k=log(length(Ftraindata)),trace=FALSE,parallel="multicore",ncpus = numCores)

Error: unexpected '=' in:

"FemaleModel1=stepGAICAll.A(FemaleModel, scope=list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),sigma.scope = list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ k="

> FemaleModel1=stepGAICAll.A(FemaleModel, scope=list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),sigma.scope = list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ k=log(length(Ftraindata)),trace=FALSE,parallel="multicore",ncpus = numCores)

---------------------------------------------------

---------------------------------------------------

ERROR: stepGAICAll has failed trying to fit the model for mu forward.

Maybe the mu model is too complicated for the data.

The model given is the final before the crush.

---------------------------------------------------

> FemaleModel2=stepGAICAll.A(FemaleModel, scope=list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),sigma.scope = list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ k=log(length(Ftraindata)),trace=FALSE,parallel="multicore",ncpus = numCores)

---------------------------------------------------

---------------------------------------------------

ERROR: stepGAICAll has failed trying to fit the model for mu forward.

Maybe the mu model is too complicated for the data.

The model given is the final before the crush.

---------------------------------------------------

> FemaleModel2=stepGAICAll.A(FemaleModel, scope=list(lower=~1,

+ upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ sigma.scope = list(lower=~1,

+ upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ k=log(length(Ftraindata)),

+ trace=FALSE,

+ parallel="multicore",

+ ncpus = numCores)

---------------------------------------------------

---------------------------------------------------

ERROR: stepGAICAll has failed trying to fit the model for mu forward.

Maybe the mu model is too complicated for the data.

The model given is the final before the crush.

---------------------------------------------------

> FemaleModel1=stepGAICAll.A(FemaleModel, scope=list(lower=~1, upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ sigma.scope = list(lower=~1,

+ upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ k=log(length(Ftraindata)),)

---------------------------------------------------

---------------------------------------------------

ERROR: stepGAICAll has failed trying to fit the model for mu forward.

Maybe the mu model is too complicated for the data.

The model given is the final before the crush.

---------------------------------------------------

> Ftraindata=Fdata[FTraindata,]

>

> Ftestdata=Fdata[-FTraindata,]

> FemaleModel2=stepGAICAll.A(FemaleModel, scope=list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),sigma.scope = list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ + k=log(length(Ftraindata)),trace=FALSE,parallel="multicore",ncpus = numCores)

Error: unexpected '=' in:

"FemaleModel2=stepGAICAll.A(FemaleModel, scope=list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),sigma.scope = list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ k="

> FemaleModel2=stepGAICAll.A(FemaleModel, scope=list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),sigma.scope = list(lower=~1,upper=~poly(age,2)+poly(bmi,3)+pb(age)),

+ k=log(length(Ftraindata)),trace=FALSE,parallel="multicore",ncpus = numCores)

---------------------------------------------------

Start: AIC= 23078.76

charges ~ 1

---------------------------------------------------

Start: AIC= 19361.86

~1

---------------------------------------------------

Start: AIC= 19361.86

~1

---------------------------------------------------

Start: AIC= 19361.86

~1

---------------------------------------------------

Start: AIC= 19361.86

charges ~ poly(bmi, 3) + poly(age, 2) + pb(age)

---------------------------------------------------

Warning messages:

1: In RS() : Algorithm RS has not yet converged

2: In RS() : Algorithm RS has not yet converged

> summary(FemaleModel2)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Family: c("BCCG", "Box-Cox-Cole-Green")

Call: gamlss(formula = charges ~ poly(bmi, 3) + poly(age, 2) + pb(age), sigma.formula = ~1, nu.formula = ~1, family = BCCG(mu.link = "log"),

data = Mtraindata, trace = FALSE, parallel = "multicore", ncpus = numCores)

Fitting method: RS()

------------------------------------------------------------------

Mu link function: log

Mu Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -3.704e+01 6.063e-02 -610.9 <2e-16 \*\*\*

poly(bmi, 3)1 1.037e-06 1.427e+00 0.0 1

poly(bmi, 3)2 -4.134e-07 1.423e+00 0.0 1

poly(bmi, 3)3 -1.967e-07 1.418e+00 0.0 1

poly(age, 2)1 1.604e-06 1.437e+00 0.0 1

poly(age, 2)2 -6.327e-08 1.417e+00 0.0 1

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------

Sigma link function: log

Sigma Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 3.102e+01 4.287e-07 72356720 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------

Nu link function: identity

Nu Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 5.000e-01 4.287e-07 1166190 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------

NOTE: Additive smoothing terms exist in the formulas:

i) Std. Error for smoothers are for the linear effect only.

ii) Std. Error for the linear terms may not be reliable.

------------------------------------------------------------------

No. of observations in the fit: 544

Degrees of Freedom for the fit: 8.000555

Residual Deg. of Freedom: 535.9994

at cycle: 3

Global Deviance: 19347.53

AIC: 19363.53

SBC: 19397.92

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Create the model for Tensorflow Deep Neural Network.

> library(keras)

> library(tensorflow)

Attaching package: ‘tensorflow’

The following object is masked from ‘package:caret’:

train

> Male\_train\_data<-trainM%>%dplyr::select(-1)%>%as.matrix()%>%as.array.default(dimnames=NULL)

Error in eval(lhs, parent, parent) : object 'trainM' not found

> Male\_train\_data<-Mtraindata%>%dplyr::select(-1)%>%as.matrix()%>%as.array.default(dimnames=NULL)

> dimnames(train\_data) <- NULL

Error in dimnames(train\_data) <- NULL : object 'train\_data' not found

> dimnames(Male\_train\_data) <- NULL

> Male\_train\_targets<-Mtraindata%>%.$TLC%>%as.array()

Error in dim(x) <- length(x) : attempt to set an attribute on NULL

> dimnames(Male\_train\_data) <- NULL

> Male\_train\_targets<-Mtraindata%>%.$TLC%>%as.array()

Error in dim(x) <- length(x) : attempt to set an attribute on NULL

> library(recipes)

Attaching package: ‘recipes’

The following object is masked from ‘package:stringr’:

fixed

The following object is masked from ‘package:stats’:

step

>

>

> library(stringr)

> library(recipes)

>

>

> cleaneddata<-recipe(Mtraindata,charges~sex+age)%>%

+ step\_center(all\_predictors())%>%

+ step\_scale(all\_predictors())

Error in eval(predvars, data, env) : object 'sex' not found

> cleaneddata<-recipe(Mtraindata,charges~bmi+age)%>%

+ step\_center(all\_predictors())%>%

+ step\_scale(all\_predictors())

> std\_func<-prep(cleaneddata,training=Mtraindata,retain=F)

> std\_test<-bake(cleaneddata, new\_data=Mtestdata)

Error: At least one step has not been trained. Please run `prep`.

> std\_test<-bake(cleaneddata, new\_data=Mtraindata)

Error: At least one step has not been trained. Please run `prep`.

> std\_test<-bake(cleaneddata, new\_data=Mtestdata)

Error: At least one step has not been trained. Please run `prep`.

> head(Mtraindata)

age bmi children smoker region charges

1 18 33.770 1 no southeast 1725.552

2 28 33.000 3 no southeast 4449.462

3 33 22.705 0 no northwest 21984.471

4 32 28.880 0 no northwest 3866.855

5 37 29.830 2 no northeast 6406.411

6 25 26.220 0 no northeast 2721.321

> std\_test<-bake(std\_func, new\_data=Mtestdata)

> std\_train<-bake(std\_func, new\_data=Mtraindata)

> male\_train\_data<-std\_train%>%

+ dplyr::select(-charges)%>%

+ as.matrix()%>%

+ as.array.default(dimnames=NULL)

> male\_test\_data<-std\_test%>%

+ dplyr::select(-charges)%>%

+ as.matrix()%>%

+ as.array.default(dimnames=NULL)

> male\_train\_targets<-Mtraindata%>%

+ dplyr::select(charges)%>%

+ as.matrix()%>%

+ as.array.default(dimnames=NULL)

> male\_test\_targets<-Mtestdata%>%

+ dplyr::select(charges)%>%

+ as.matrix()%>%

+ as.array.default(dimnames=NULL)

> build\_model <- function() {

+ model <- keras\_model\_sequential() %>%

+ layer\_dense(units = 128, activation = "relu",

+ input\_shape = dim(male\_train\_data)[[2]]) %>%

+ layer\_dense(units = 128, activation = "relu") %>%

+ layer\_dense(units = 128, activation = "relu") %>%

+ layer\_dense(units =128, kernel\_regularizer = regularizer\_l2(0.001))%>%

+ layer\_dense(units = 1)

+

+ model %>% compile(

+ optimizer = "rmsprop",

+ loss = "mae",

+ metric= "msle"

+ )

+ }

> model <- build\_model()

> model %>% fit(male\_train\_data, male\_train\_targets,

+ epochs = 200, batch\_size = 30,verbose=0,

+ validation\_split = 0.1) -> deepneural

> summary(model)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

========================================================================================================================================================================

dense\_1 (Dense) (None, 128) 384

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 128) 16512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_3 (Dense) (None, 128) 16512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_4 (Dense) (None, 128) 16512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_5 (Dense) (None, 1) 129

========================================================================================================================================================================

Total params: 50,049

Trainable params: 50,049

Non-trainable params: 0

Female Model

> Fcleaneddata<-recipe(Ftraindata,charges~bmi+age)%>%

+ step\_center(all\_predictors())%>%

+ step\_scale(all\_predictors())

> Fstd\_func<-prep(Fcleaneddata,training=Ftraindata,retain=F)

> Fstd\_test<-bake(Fstd\_func, new\_data=Ftestdata)

> Fstd\_train<-bake(Fstd\_func, new\_data=Ftraindata)

> female\_train\_data<-Fstd\_train%>%

+ dplyr::select(-charges)%>%

+ as.matrix()%>%

+ as.array.default(dimnames=NULL)

> female\_test\_data<-Fstd\_test%>%

+ + dplyr::select(-charges)%>%

+ + as.matrix()%>%

+ + as.array.default(dimnames=NULL)

Error in dplyr::select(-charges) : object 'charges' not found

>

> female\_test\_data<-Fstd\_test%>%

+ dplyr::select(-charges)%>%

+ as.matrix()%>%

+ as.array.default(dimnames=NULL)

> female\_train\_targets<-Ftraindata%>%

+ dplyr::select(charges)%>%

+ as.matrix()%>%

+ as.array.default(dimnames=NULL)

>

> > female\_test\_targets<-Ftestdata%>%

Error: unexpected '>' in ">"

> dplyr::select(charges)%>%

+ as.matrix()%>%

+ as.array.default(dimnames=NULL)

Error in dplyr::select(charges) : object 'charges' not found

> female\_test\_targets<-Ftestdata%>%

+ dplyr::select(charges)%>%

+ as.matrix()%>%

+ as.array.default(dimnames=NULL)

> build\_model\_female <- function() {

+ model\_female <- keras\_model\_sequential() %>%

+ layer\_dense(units = 128, activation = "relu",

+ input\_shape = dim(female\_train\_data)[[2]]) %>%

+ layer\_dense(units = 128, activation = "relu") %>%

+ layer\_dense(units = 128, activation = "relu") %>%

+ layer\_dense(units =128, kernel\_regularizer = regularizer\_l2(0.001))%>%

+ layer\_dense(units = 1)

+

+ model\_female %>% compile(

+ optimizer = "rmsprop",

+ loss = "mae",

+ metric= "msle"

+ )

+ }

> model\_female <- build\_model\_female()

> model\_female %>% fit(female\_train\_data, female\_train\_targets,

+ epochs = 200, batch\_size = 30,verbose=0,

+ validation\_split = 0.1) -> deepneuralfemale

> summary(model\_female)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

========================================================================================================================================================================

dense\_6 (Dense) (None, 128) 384

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_7 (Dense) (None, 128) 16512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_8 (Dense) (None, 128) 16512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_9 (Dense) (None, 128) 16512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_10 (Dense) (None, 1) 129

========================================================================================================================================================================

Total params: 50,049

Trainable params: 50,049

Non-trainable params: 0

**IV: RESULTS:**

Final Steps, Comapares both Models and Predict the model:

> DeepNeuralNetweok=predict(model,male\_test\_data)

> LamdaMS=predict(m2,newdata=Mtestdata,type="response")

Error in predict(m2, newdata = Mtestdata, type = "response") :

object 'm2' not found

> Truth=Mtestdata$charges

> DeepNeuralNetweok=predict(model,male\_test\_data)

> LamdaMS=predict(MaleModel1,newdata=Mtestdata,type="response")

new prediction

New way of prediction in pb() (starting from GAMLSS version 5.0-3)

> PredictMaleData=cbind(Truth,DeepNeuralNetweok,LamdaMS)%>%as\_data\_frame()

> colnames(PredictMaleData)=c("Truth","DeepNeutralNetwork","LamdaMS")

> PredictMaleData$age = Mtestdata$age

> library(mlr)

Attaching package: ‘mlr’

The following object is masked from ‘package:tensorflow’:

train

The following object is masked from ‘package:caret’:

train

The following object is masked from ‘package:gamlss.data’:

db

> library(tensorflow)

> regression.task= mlr::makeRegrTask(id = "Maledata", data=Mtestdata, target = "Charges")

Error in makeSupervisedTask("regr", data, target, weights, blocking, coordinates, :

Column names of data doesn't contain target var: Charges

> regression.task= mlr::makeRegrTask(id = "Maledata", data=Mtestdata, target = "charges")

> regression.lrn = makeLearner("regression.glm")

Error in makeLearner("regression.glm") :

Couldn't find learner 'regression.glm'

Did you mean one of these learners instead: regr.h2o.glm regr.glm regr.h2o.gbm

> regression.lrn = makeLearner("regr.glm")

> dummy=mlr::train(regression.lrn,regression.task)

> predDeepNeuralNetwork=predict(dummy,regression.task)

> predLamdaMS=predict(dummy,regression.task)

> predDeepNeuralNetwork$data$response<-DeepNeuralNetweok

> predLamdaMS$data$response<-LamdaMS

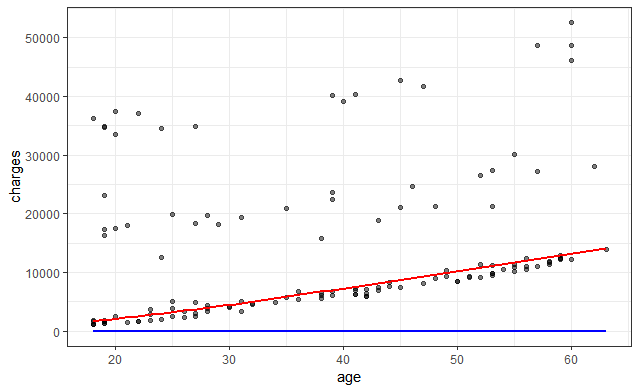
> PredictMaleData%>%ggplot()+geom\_point(aes(x=age,y=Truth),col="black",alpha=0.5)+

+ geom\_smooth(aes(x=age,y=DeepNeuralNetweok),col="red",fill="red",alpha=0.2)+

+ geom\_smooth(aes(x=age,y=LamdaMS),col="blue",fill="blue",alpha=0.2)+theme\_bw()+scale\_y\_continuous("charges")

`geom\_smooth()` using method = 'loess' and formula 'y ~ x'

`geom\_smooth()` using method = 'loess' and formula 'y ~ x'



Notes:

Deep Neural Network is the Red Lines

LMS Regression is Blue Light.

**V:CONCLUSIONs:**

Based on the graph, Deep Neural Network produces more accurate then LMS Regression. The graph show that, the older age will pay more heath care insurance cost in a year.

[1] “Generalized Additive Models for Location Scale and Shape (GAMLSS) in R” Rigby RA. and Stasinopoulos DM <https://www.jstatsoft.org/article/view/v023i07/v23i07.pdf>

[2] <https://people.richland.edu/james/lecture/m170/ch08-int.html>

[3] <https://www.kaggle.com/mirichoi0218/insurance/version/1>