Universidad Autónoma del Estado de México

Facultad de Ciencias

Diplomado en Machine Learning

Proyecto Final Módulo 1

Ejercicio 1

Autor: José de Jesús Rodríguez Barreto

Introducción

En este proyecto se ha puesto en práctica principalmente el tratamiento de los datos pues aunque pareciera algo bastante sencillo en realidad es un reto a resolver mediante suficiente pruebas, por lo que a grandes rasgos se vieron las variables y que tan relacionadas estaban con la variable que se busca predecir, después de esto se buscó tratar los outliers así como los datos faltantes para posteriormente estandarizar los datos mediante diferentes técnicas entre las que están, la normalización z, normalización de máximos y mínimos y una normalización robusta, pasando ya al siguiente paso que fue la construcción del algoritmo y el análisis de resultados.

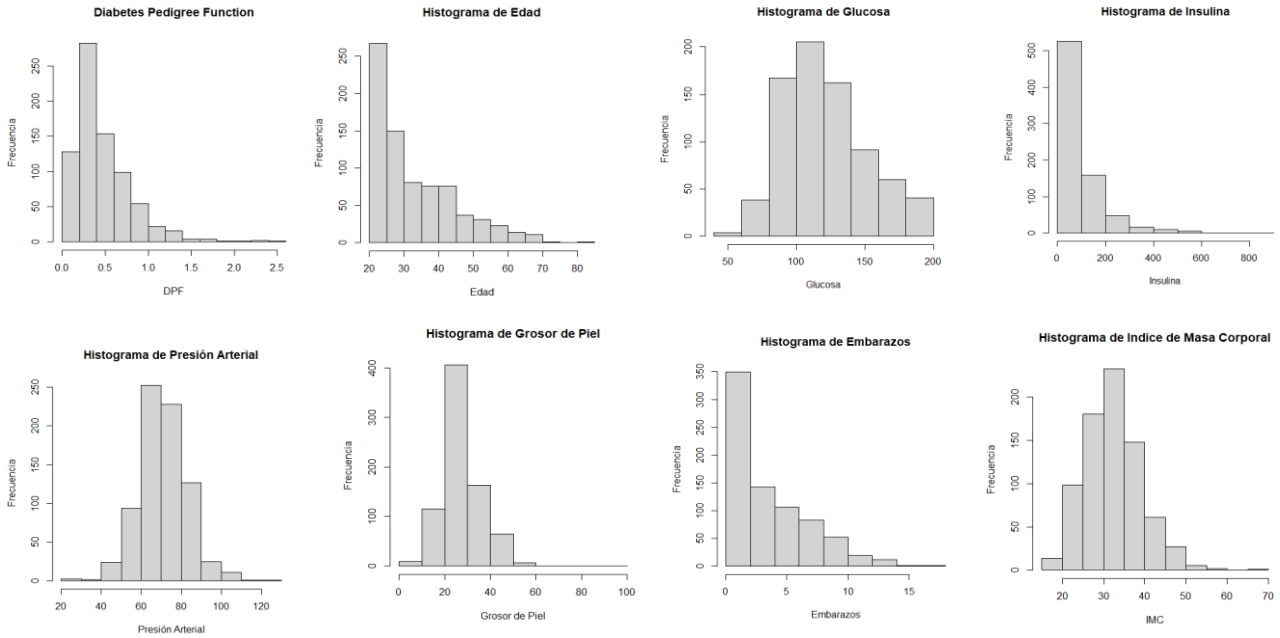
Análisis de las variables

La información usada en este proyecto proviene de la página web kaggle la cual sustrajo la base de datos del National Institute of Diabetes and Digestive and Kidney Diseases. Esta base de datos está conformada por pacientes mujeres a partir de 21 años, consta de 9 variables, 8 de las cuales son independientes entre sí y una dependiente que es la que se busca predecir, estas variables son:

- Pregnancies: Número de veces que una mujer ha estado embarazada.
- Glucose: Concentración de glucosa.
- BloodPressure: Presión sanguínea.
- SkinThickness: El grosor de la piel en el Tríceps.
- Insulin: La medida de la insulina.
- BMI: Índice de masa corporal.
- Age: Edad de las pacientes.
- DiabetesPedigreeFunction: función que evalúa la propensión a padecer diabetes basada en el historial familiar.
- Outcome: Diagnostico 0 para los que no padecen diabetes y 1 para los que si padecen diabetes

Preprocesamiento de datos.

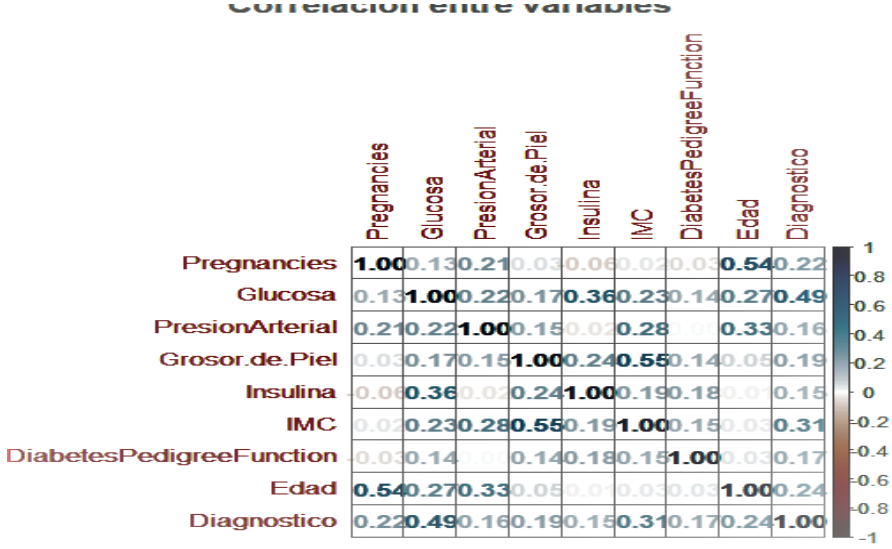
Iniciamos el preprocesamiento tratando los datos faltantes para esto graficamos los datos de la base de datos, así vemos el tipo de grafica que tienen:



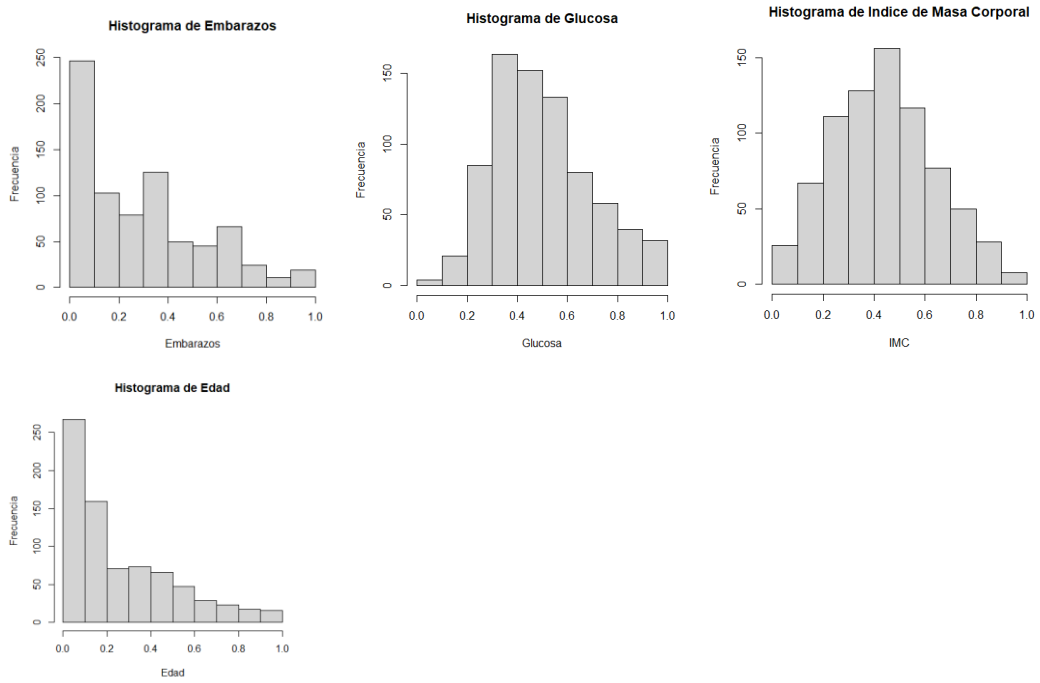
Si nos damos cuenta tenemos medidas como la glucosa la insulina o la presión arterial que no pueden ser 0 para modificar estos datos vacíos vemos el tipo de grafica que tienen si es una grafica normal como la de la glucosa o el índice de masa corporal los 0 podemos reemplazarlos por el valor del promedio, pero si no lo es como se ve en la grafica de la insulina se sustituye el 0 por el valor de la mediana pues el promedio se ve muy afectado por los outliers. Ya que se hizo esto es momento de tratar con los outliers, en la grafica se puede ver cómo hay valores máximos que están muy separados de las gráficas para encontrar estos valores se ha considerado como un outlier aquel valor que se encuentra alejado del tercer cuartil la distancia de 1.5 veces el rango intercuartílico, esto es:

$$O > Q_3 + 1.5\, IQR$$

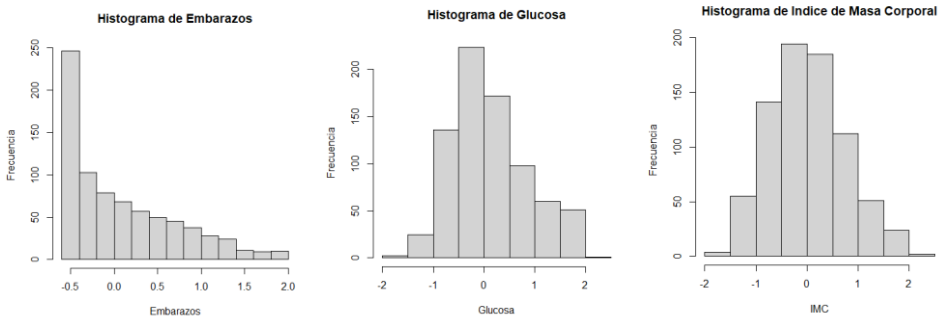
Después de que estos sean encontrados son reemplazados por la mediana al encontrarse la mayoría lo suficientemente lejos para afectar la media.
A continuación, se evaluó la correlación de las variables para saber cuáles afectan más a la predicción mediante un mapa de calor el cual se presenta a continuación:

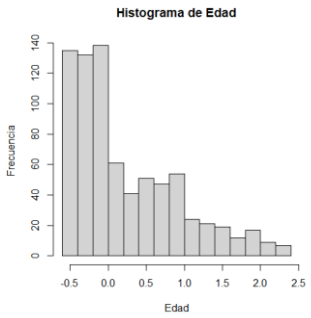


De esta manera podemos observar cuales son las variables que más se relacionan con nuestra variable a estudiar dándonos cuenta de que no todos los datos tienen una relación importante con nuestro objeto de estudio por lo que se opto por eliminar de la base de datos las variables con una correlación menor al 20%, lo que nos lleva a descartar las variables de Presión Arterial, Grosor de Piel, Insulina y Diabetes Pedigree Function. Ya con las variables bien seleccionadas y sin valores faltantes ni outliers procedemos a normalizar los datos la primera normalización que se busco fue de máximos y mínimos, pero al ver las graficas nos damos cuenta de que no funciono pues estas aún se ven sesgadas exceptuando a las que ya se veían normales.

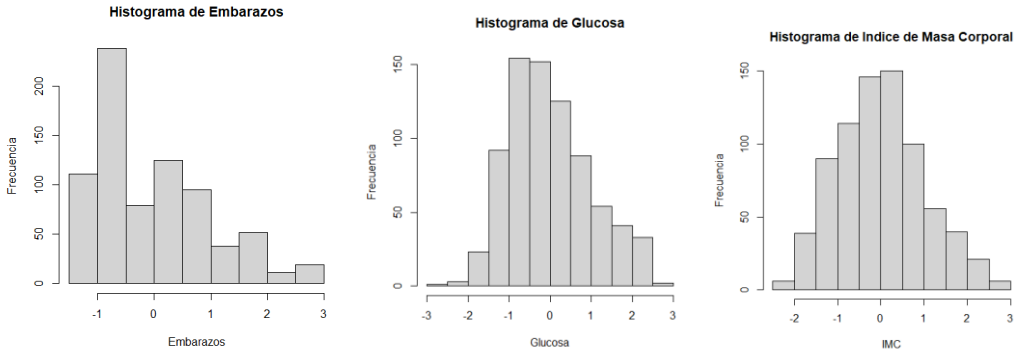


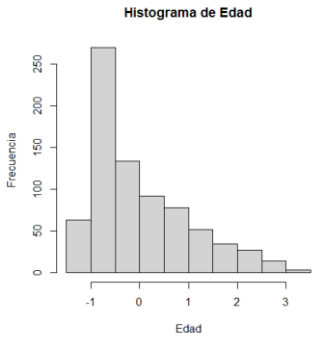
Por lo que se recurrió a una normalización robusta que consiste en restarle a la medida la media y dividirlo entre el rango intercuartílico lo que nos deja las siguientes graficas:





Con estas vemos que las graficas siguen sesgadas por lo que sigue hacer una normalización estándar que nos deja las siguientes graficas:





La cual nos deja unos datos con unas graficas mas estables por lo que es una transformación en la que podemos trabajar sin problema.

Descripción del Algoritmo

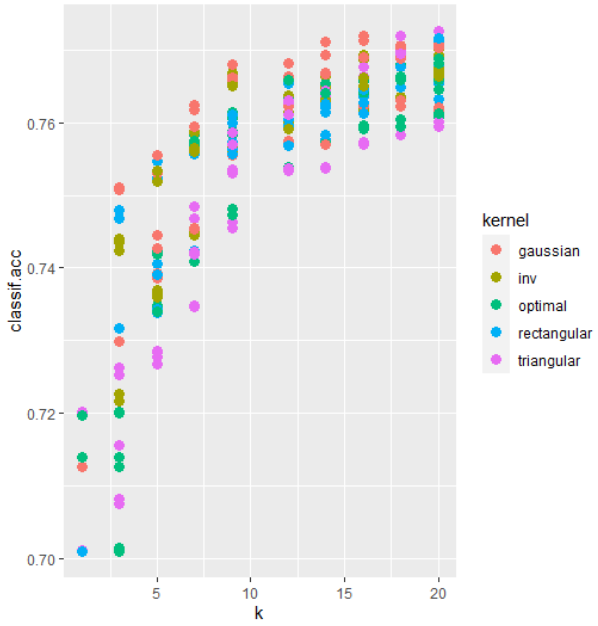
El método por emplear será el de k-vecinos más cercanos ponderados el cual consiste en buscar mediante la proximidad entre datos la asociación con los demás. Es un método no paramétrico, en el que una nueva observación se coloca en la clase de las observaciones del conjunto de entrenamiento mas cercanas respecto a las variables empleadas, para esto se emplea el concepto de distancia. A parte del concepto de distancias se contempla que las observaciones mas cercanas tengan un mayor peso en la decisión que aquellos vecinos más alejados de la nueva observación, para esto se usa una función llamada kernel que evaluara la distancia, la cual puede ser rectangular, triangular, gaussiano, inverso entre otros. Para este proyecto usaremos los mencionados antes.

Construcción del modelo.

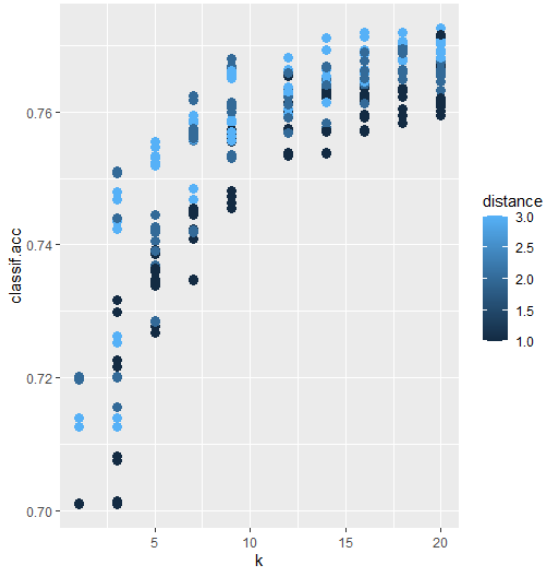
El modelo fue construido siguiendo el k-vecinos ponderados auto calibrados, para esto se le pidió que buscara entre 1 y 20 vecinos más cercanos el número optimo y las distancias entre 1 y 3, el kernel que daría los pesos a buscar fue dado a elegir entre rectangular, triangular, gaussiano e inverso, se uso una validación cruzada repetida con 10 divisiones y 5 repeticiones que al ver las graficas se puede ver que el mejor modelo esta dado por un k de 20 una distancia de 3 y un kernel triangular, como se apreciara en la siguiente imagen.

```
> ibusqueda$archive$best()
    k distance     kernel scale classif.acc runtime_learners
1: 20        3 triangular FALSE   0.7726418             1.36
                                uhash  x_domain           timestamp
1: 5a410c1d-3eab-46aa-8b85-a2fd6299a41f <list[4]> 2022-01-06 19:32:10
   batch_nr
1:      127
```

Sin embargo, en las siguientes graficas se pueden ver de una manera más clara estos resultados.



En la gráfica de la izquierda se puede apreciar como en u k de veinte el kernel triangular tiene una precisión más alta.



En esta grafica se puede apreciar como la distancia de 3 con el azul más claro es la que consiguió la precisión más alta.

A continuación se busco que mediante una validación cruzada anidada el hacer una estimación más confiable del desempeño predictivo del modelo para esto en la validación externa se usó una de retención con un ratio de 0.7 y para la interna se usó una validación cruzada simple con 10 divisiones, ya con esto hecho se hizo un objeto que sirviera para llevar a cabo el proceso de selección del modelo y se combinaron las dos validaciones con la tarea, para con el modelo optimo poder entrenar utilizando todo el conjunto de datos.

Análisis de resultados.

Al final del entrenamiento, se hizo una predicción sobre todo el conjunto de datos la cual nos arrojó la siguiente matriz de confusión:

```
          truth
response   1   0
       1 200  55
       0  68 445
acc :  0.8398; ce  :  0.1602; dor :  23.7968; f1  :  0.7648
fdr :  0.2157; fnr :  0.2537; fomr:  0.1326; fpr :  0.1100
mcc :  0.6440; npv :  0.8674; ppv :  0.7843; tnr :  0.8900
tpr :  0.7463
```

Donde podemos ver que de 768 pacientes clasifico 645 correctamente dándonos una precisión del 83.98% que redondeando podría tomarse como un 84%, entre los datos mal clasificados podemos ver que a 68 pacientes con diabetes los clasifico como pacientes sanos y a 55 pacientes sanos los clasifico como pacientes con diabetes.

Conclusiones

En conclusión, la calidad real del modelo después de la validación anidada aumento un 7% aproximadamente lo cual es un resultado bastante aceptable sin embargo debe de existir una mejor forma de clasificar los resultados pues un error del 16% aun es demasiado amplio en cuestiones de salud pues pone en riesgo tanto a los

pacientes que no recibirán tratamiento como a aquellos que recibirán un tratamiento que no necesitan. El proceso de construcción del clasificador fue bastante interesante siendo la parte más complicada a mi parecer el preprocesamiento de los datos pues en un principio pensaba que con las normalizaciones me podría hacer cargo de los outliners, pero al final me di cuenta de que tenía que darles un tratamiento aparte. Este tipo de aprendizaje automático tiene principalmente aplicaciones en cualquier clasificación entre niveles como los factores como pueden ser especies para biología, si dar o no un crédito en las instituciones financieras entre otras. Estar practicando constantemente este tipo de conocimientos es muy importante pues te ayuda a dominar tanto los comandos como la lógica detrás de estos pues el solo estudiar la teoría te hace lento, por lo que practicas como está ayudan a que los conocimientos teóricos se agilicen cuando llegue la hora de ponerlos a trabajar de forma profesional.

Anexo: Código.

```
datos <- read.csv("diabetes.csv")
hist(diabdat$Pregnancies, xlab = "Embarazos", ylab = "Frecuencia", main = "Histograma de Embarazos")
hist(diabdat$Glucosa, xlab = "Glucosa", ylab = "Frecuencia", main = "Histograma de Glucosa")
hist(datos$PresionArterial, xlab = "Presión Arterial", ylab = "Frecuencia", main = "Histograma de Presión Arterial")
hist(datos$Grosor.de.Piel, xlab = "Grosor de Piel", ylab = "Frecuencia", main = "Histograma de Grosor de Piel")
hist(datos$Insulina, xlab = "Insulina", ylab = "Frecuencia", main = "Histograma de Insulina")
hist(diabdat$IMC, xlab = "IMC", ylab = "Frecuencia", main = "Histograma de Indice de Masa Corporal")
hist(datos$DiabetesPedigreeFunction, xlab = "DPF", ylab = "Frecuencia", main = "Diabetes Pedigree Function")
hist(diabdat$Edad, xlab = "Edad", ylab = "Frecuencia", main = "Histograma de Edad")
hist(datos$Diagnostico, xlab = "Diagnostico", ylab = "Frecuencia", main = "Histograma de Diagnostico")
x <- cor(datos, method = "pearson")
corrplot(x, method = "number", title = "Correlacion entre variables")
datos <- subset(datos, select = -c(PresionArterial, Insulina, DiabetesPedigreeFunction, Grosor.de.Piel))
datos <- mutate_at(datos,c("Diagnostico"), as.factor)
diabdat <- datos
robust_scalar <- function(x){(x-median(x))/(quantile(x,probs = .75)-quantile(x,probs= .25))}
diabdat <- subset(diabdat, select = -c(Diagnostico))
outliers <- function(data, lowLimit, highLimit){
              data[data < lowLimit] <- mean(data)
              data[data > highLimit] <- median(data)
              data}
diabdat$Pregnancies <- outliers(diabdat$Pregnancies, quantile(diabdat$Pregnancies, probs = 0.25)-(1.5*(quantile(diabdat$Pregnancies, probs = 0.75)-quantile(diabdat$
diabdat$IMC <- outliers(diabdat$IMC, quantile(diabdat$IMC, probs = 0.25)-(1.5*(quantile(diabdat$IMC, probs = 0.75)-quantile(diabdat$IMC, probs = 0.25))), quantile(d
diabdat$Edad <- outliers(diabdat$Edad, quantile(diabdat$Edad, probs = 0.25)-(1.5*(quantile(diabdat$Edad, probs = 0.75)-quantile(diabdat$Edad, probs = 0.25))), quant
diabdat <- as.data.frame(lapply(diabdat, standarize))
diabdat <- diabdat %>% mutate_each_(list(~scale(.) %>% as.vector), vars = c("Pregnancies","IMC","Edad","Glucosa"))
diabdat$Diagnostico <- datos$Diagnostico
str(diabdat)
tareadiab <- TaskClassif$new(id="clasif", backend=diabdat, target="Diagnostico")
atm <- lrn("classif.kknn")
atm$param_set
soluciones <- ParamSet$new(list(ParamInt$new("k",1,20), ParamInt$new("distance",1,3), ParamFct$new("kernel",c("rectangular","triangular","gaussian","inv","optimal")
busqueda <- mlr3tuning::tnr("grid_search")
vc <- rsmp("repeated_cv",folds=10,repeats=5)
ibusqueda <- TuningInstanceSingleCrit$new(task=tareadiab, learner = atm, resampling = vc, measure = msr("classif.acc"), search_space = soluciones, terminator=trm("n
future::plan("multisession")
busqueda$optimize(ibusqueda)
ibusqueda$archive$best()
ggplot(ibusqueda$archive$data, aes(x=k,y=classif.acc,col=scale)) + geom_point(size=3)
ggplot(ibusqueda$archive$data, aes(x=k,y=classif.acc,col=kernel)) + geom_point(size=3)
ggplot(ibusqueda$archive$data, aes(x=k,y=classif.acc,col=distance)) + geom_point(size=3)
vcexterna<-rsmp("holdout", ratio=0.7)
vcinterna<-rsmp("cv", folds=10)
optim.learner1<-AutoTuner$new(learner=atm,
                              vcinterna,msr("classif.acc"),
                              soluciones,
                              terminator = trm("none"),
                              tuner=busqueda)
future::plan("multisession")
resultados1<-resample(tareadiab,optim.learner1,vcexterna)
resultados1$aggregate(measures=msr("classif.acc"))
modelo_final1<-optim.learner1$train(tareadiab)
prediccion<-modelo_final1$predict_newdata(diabdat)
confusion_matrix(truth = prediccion$truth, response=prediccion$response, positive = "1")
```