Predicción de enfermedad cardíaca mediante KN-Neighbors

Angel Barra Muñoz
angel.barra@usach.cl
Introducción a la ciencia de datos, Ingeniería Física
7 de marzo, 2022

Resumen

Se analizó el conjunto de datos *heart* en formato *csv*, los cuales poseían distintas características de pacientes y si estos, tenían o no, una enfermedad cardíaca. Las predicciones se realizaron mediante el método de predicción *KN-Neighbors* con K = 15, tomando un 70% de los datos para entrenamiento, y el otro 30% para la validación, obteniendo así, una precisión del 84% para el conjunto de entrenamiento, y un 86% para el conjunto de validación.

1 Introducción

Las enfermedades cardiovasculares son el número uno de muertes a nivel mundial, con un estimado de 17.9 millones de vidas por año, lo que corresponde en proporción a un 31% de los fallecimientos a lo largo del mundo, 4 de cada 5 muertes por enfermedades cardiovasculares son por ataques al corazón, y un tercio de estas ocurren de forma prematura antes de los 70 años. El algoritmo *KN-Neighbors* o por sus siglas *KNN* es un algoritmo que trabaja mediante la etiqueta de valores cercanos a otros. Para predecir, por ejemplo, la pertenencia de un punto, si decimos que K=1, *KNN* tomará al punto más cercano y le otorgará dicha pertenencia al que buscábamos, por lo que, si aumentamos el valor de K a estaremos considerando un cúmulo de datos mayor para hallar la pertenencia del punto en cuestión. Como acotación, cabe destacar que a mayor valor de K no necesariamente se obtiene una mayor precisión.

2 Metodología

De la web de kaggle obtuvimos el archivo de datos *heart.csv* que contiene un total de 918 sujetos con 12 características, incluyendo si el sujeto posee o no una enfermedad cardíaca, la cual corresponde a *HeartDisease*, la que consideramos como nuestra variable a predecir.

```
hd = pd.read_csv('heart_1.csv')
hd.head()
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

Figure 1: Visualización del archivo *heart.csv*

Como se puede observar en la figura 1 hay columnas con datos no numéricos, para solucionar esto utilizamos el comando *.replace* para poder asignar a estas variables, valores numéricos, de 0 a n, siendo n el número de opciones dentro de una columna. Luego con el comando *.drop* para crear la variable **x** sin el valor correspondiente a *HeartDisease* y así este guardarlo en una variable **y**.
Con las dos variables ya creadas y como podemos ver en la figura 1, al aplicar el comando *.replace* obtendremos valores con distintos órdenes de magnitud, lo que presentaría un problema por lo que se utilizó un reescalamiento de las variables mediante el comando *preprocessing.MinMaxScaler()*, para que así todas las variables quedarán entre valores de 0 y 1.
Tras esto aplicamos el comando *.train_ test_ split()*, para crear las variables **xtrain** e **ytrain** para

el entrenamiento y **xtest** e **ytest** para la validación.
Luego realizamos un ciclo *for* para conocer cual era el valor de K para nuestro método, obteniendo así la siguiente figura:

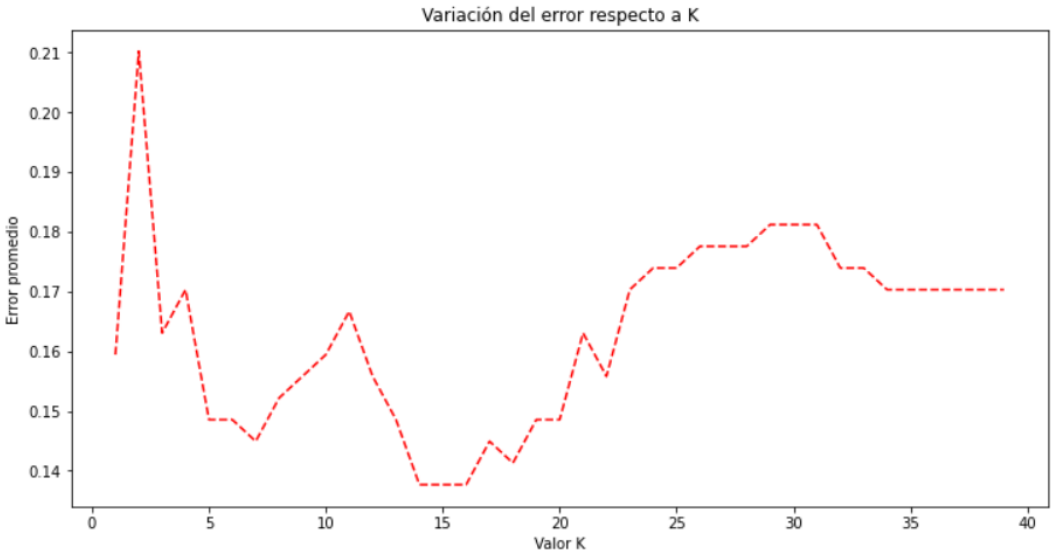

Figure 2: Error en función de K

Como se puede observar en la figura 2, el error mínimo es para los valores de K=14, 15 y 16, por que arbitrariamente utilizaremos el número 15.
Para aplicar *KNN* se utilizó el comando *KNeighborsClassifier(n_ neighbors=15)* con los datos de entrenamiento, de la forma *classifier.fit(xtrain, ytrain)*. el entrenamiento hecho, creamos la variable **ypred** con el conjunto **xtest** a través del comando *classifier.predict()*.
Con los resultados listos se generó una matriz de confusión para la observación de estos.

3 Resultados

Los resultados arrojados por el alogirtmo tienen una precisión del 84% para el conjunto de entrenamiento y un 86% para el conjunto de validación con la siguiente matriz de confusión:

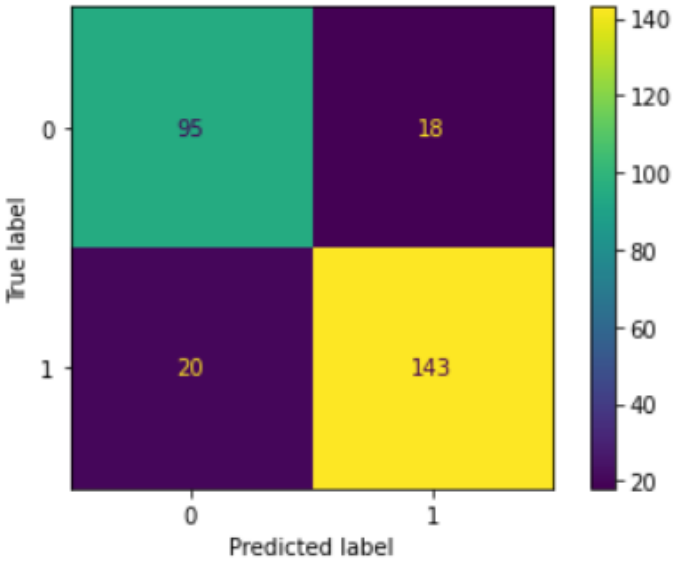

Figure 3: Matriz de confusión

Podemos ver que nuestro modelo predijo 95 casos de personas sin ninguna enfermedad cardíaca que eran correctos, mientras que predijo otros 20 que corresponden a casos de personas que si tienen enfermedades cardíacas, por lo que estaban equivocados. Para el caso de sujetos con enfermedades cardíacas predijo un total de 143 que sí tienen una enfermedad, mientras que predijo otros 18 que no tenían problemas al corazón.
El error asociado a este modelo es de un 14%.
Para buscar una mayor precisión decidimos buscar las variables con mayor correlación, para esto utilizamos la herramienta *mutual_info_classif* que nos permite hacerlo.
Del total de variables, las 4 con mayor correlación corresponden a *'ChestPainType'*, *'EcerciseAngina'*,*'Oldpeak'* y *'ST_Slope'*. Y aplicamos la misma forma para obtener el valor óptimo de K, como se ve a continuación:

Figure 4: Error en función de K

Donde podemos notar que el valor óptimo es K=5. Con estas variables y K=5 obtuvimos la siguiente matriz de confusión:

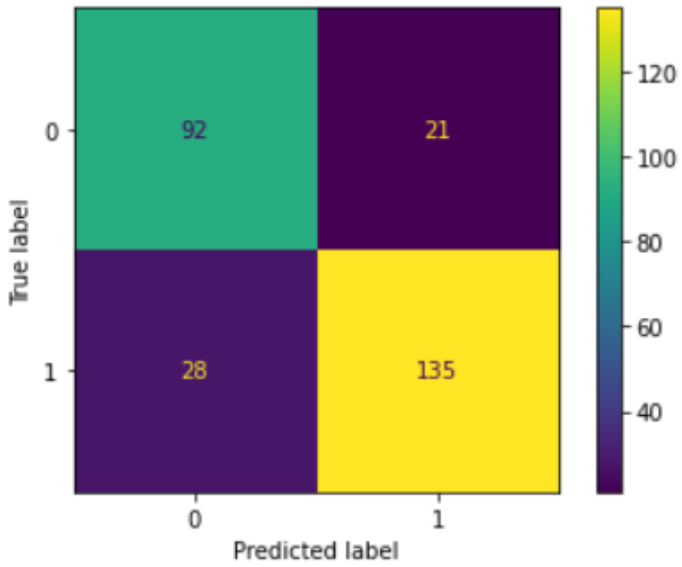

Figure 5: Matriz de confusión con las 4 variables

En esta matriz de confusión podemos ver que el modelo en este caso predijo que habrían 92 sujetos sanos que efectivamente lo son, mientras que también predijo 28 que son personas con enfermedades cardiacas, por otra parte, predijo a 135 sujetos con enfermedades cardíacas que efectivamente lo eran, mientras que predijo 21 que corresponden al grupo de las sanas.
El error en este caso fue de un 18% mientras que la precisión del modelo de entrenamiento fue de un 86% y el de validación de un 82%.
Como podemos ver al disminuir el número de variables perdimos precisión en nuestro modelo.

4 Conclusiones

Como objetivo se propuso establecer un modelo de predicción mediante *KN-Neighbors* para predecir según distintas características si un sujeto está propenso o no a padecer enfermedades al corazón. De las dos prediccione que utilizamos consideramos que la primera tuvo un una mejora en la precisión en el grupo de validación, con un 86% aumentando en un 2% la precisión del grupo de entrenamiento. Lo que en términos médicos es una buena aproximación al momento de querer monitorear a un paciente de enfermedad cardíaca, ya que tendriamos conocimiento de un 86% de quienes tienen problemas al corazón.
Como vimos al disminuir el grupo de x a las variables con mayor correlación perdimos un 4% de precisión en el seguimiento de sujetos con enfermedades cardíacas, para nuestro conjunto de datos, esto representaría una pérdida de aproximadamente 37 sujetos.
Para un estudio más detallado se propone buscar relaciones de una en una las variables con el diagnóstico, de esta forma podríamos observar relaciones más detalladas, directas y con mayor precisión en la predicción de enfermedades cardíacas.

5

References

[1] https://www.kaggle.com/fedesoriano/heart-failure-prediction

[2] Python Data Science Handbook, Essental Tool for working with data, Jake VanderPlas