# Caso de estudio 3

## Predicción de enfermedad de corazón

En la actualidad, las enfermedades cardíacas son la principal casusa de muerte a nivel mundial[1] . La enfermedad isquémica del corazón fue responsable del 16% de las muertes globales en el 2019. En los Estados Unidos, la principal causa de muerte es otra enfermedad cardíaca llamada enfermedad de la arteria coronaria[2].

Además de esto, entre 3 a 5% de las admisiones a hospitales se deben a pacientes con insuficiencia cardíaca. Un 2% de los costos médicos de los países desarrollados son resultado de dicha condición.[3]

Debido a este problema mundial, una de las disciplinas en las que se aborda el tema con la intención de mitigar los riesgos y costos, mejorando así la calidad de vida de las personas es Machine Learning. Utilizando diferentes técnicas de aprendizaje automático, se puede estimar con cierto grado de confianza la presencia, el subtipo, y la severidad de una enfermedad cardíaca. Más aún, se pueden predecir eventos como desestabilizaciones, re-hospitalizaciones y mortalidad.

**2- Descripción de los conjutos de datos utilizados**

Los conjuntos de datos utilizados en este caso de estudio son 4 datasets distintos. Cada uno de estos contiene medidas de 76 atributos (los mismos atributos en cada dataset), pero los datasets provienen de diferentes instituciones médicas. Estas son: Cleveland Clinic Foundation, Hungarian Institute of Cardiology, Budapest, V.A. Medical Center, Long Beach, CA y University Hospital, Zurich, Switzerland.

Los datasets contienen entre 123 y 294 ejemplos cada uno.

Algunos de los atributos a destacar, utilizados en estudios previos son

1. Age: edad en años del paciente. Posee una distribución gaussiana con media 53, y desvío 9.
2. Sex: 1-Masculino 0-Femenino. Predominan los ejemplos masculinos por un gran margen (711 contra 188)
3. Cp: Tipo de dolor en el pecho. Valores de 1 a 4. El valor 4 es el predominante (asintomatico, 485 ejemplos), mientras que el 1 es el más escaso con solo 45 ejemplos. Los valores 2 y 3 poseen entre 150 y 200 ejemplos.
4. Trestbps: Presión sanguínea en reposo (medida en mm Hg al ingresar al hospital). Posee una distribución gaussiana. Tiene valores faltantes.
5. Chol: colesterol en suero (en mg/dl). Distribución no se asemeja a una conocida.
6. Fbs: nivel de azucar en ayuno > 120 mg/dl (1 = verdadero; 0 = falso). 0 es mucho más prevalente en el dataset (674 vs 135)
7. Restecg: electrocardiograma en reposo (0-Normal, 1-having ST-T wave abnormality, 2-showing probable or definite left ventricular hypertrophy by Estes' criteria). 0 es el valor más prevalente, 1 y 2 tienen cardinalidades parecidas (177 y 182)
8. Thalach: ritmo cardíaco máximo alcanzado. Se asemeja a una distribución normal con un leve sesgo hacia la izquierda.
9. Exang: angina inducida por ejercicio (1-Si, 0-No). Existen 514 ejemplos para el valor 0 y 330 para el valor 1.
10. Oldpeak: ST depression induced by exercise relative to rest. Distribución no se asemeja a una conocida.
11. Slope: the slope of the peak exercise ST segment (1-upsloping, 2-flat, 3-downsloping). El caso más común es 2, seguido por 1 y el menos prevalente es el 3.
12. Ca: number of major vessels (0-3) colored by flourosopy. Distribución sesgada hacia la derecha. A medida que se aumenta el número de vasos, los ejemplos disminuyen.
13. Thal: 3 = normal; 6 = fixed defect; 7 = reversable defect. Posee 477 valores faltantes. Mayor prevalencia en 3 y 7 (alrededor de 180 ejemplos).
14. Num: diagnosis of heart disease (angiographic disease status) (0- < 50% diameter narrowing, 1- > 50% diameter narrowing). Los posibles valores enteros van del 0 al 4. #0=404, #1=191, #2=130, #3=132, #4=42. 0 significa que el paciente no posee la condición y los valores de 1 a 4 indican la severidad de la misma

Los datos poseen atributos faltantes y están representados con el valor -9.

El dataset de Cleveland posee unas pocas filas corruptas por encoding al final del archivo. Las primeras 282 filas de las 293 no tienen problemas, por lo que esas serán las utilizadas en los siguientes pasos del caso de estudio.

**3- Preparación previa de los datos**

**Selección de atributos previa**

Debido a la gran cantidad de valores faltantes en algunos atributos, decido descartar los que tengan más de 33% de valores faltantes. También descarto name e id.

Columnas descartadas:

Name, Id, ccf, pncaden, cigs, years, dm, famhist, smoke, thaltime, slope, rldv5, ca, restckm, exerckm, restef, restwm, exeref, exerwm, thal, thalsev, thalpul, earlobe, ramus, om2, cathef, junk.

**Estrategias de imputación**

Para los predictores con más valores nulos dentro de los que no fueron descartados y de tipo categóricos, decido elegir un valor definido que represente que ahí existe un valor nulo. Imputo con el valor -1. Luego, decido imputar los restantes predictores categóricos con su moda (estos tienen menos de 19 valores nulos que equivale aproximadamente a un 2% de los ejemplos). Además, estos poseen una cardinalidad extremadamente alta en su moda, por lo que esta imputación, con un grado de seguridad muy alto, representa el valor exacto que tendrían estos ejemplos. Por último, se decide imputar los últimos valores faltantes con la media, ya que la gran mayoría de estos poseen distribuciones gaussianas o uniformes y valores enteros o reales. Hay 3 predictores en los que hubiera escogido una estrategia de imputación utilizando la mediana por su distribución fuertemente sesgada hacia la derecha (predictores ridv5e, proto y met, véase la imagen a continuación). Se podría aplicar una transformación logarítmica a estos 3 predictores para obtener una distribución más parecida a una gaussiana.

(Distribuciones simil-gaussianas)

Imagen que contiene interior, computadora, computer, grande

Descripción generada automáticamente

(Distribuciones fuertemente sesgadas hacia la derecha)

Una captura de pantalla de una red social

Descripción generada automáticamente

Para el modelado con KNN, se aplicará además una normalización a los datos, ya que este modelo requiere dicha normalización.

Una nota importante sobre la preparación previa de los datos es que la única estrategia que se realizará previo a los bloques anidados de cross validation es la estrategia de imputación con el valor conocido -1 que representa que el valor es uno no conocido. Esto no depende del set de datos elegido por un Split específico de entrenamiento/test, ya que este valor siempre será el mismo.

La historia es diferente cuando hablamos de la media, moda y la transformación específica que se aplica para normalizar. Estas transformaciones sí dependen de los datos que se eligen, por lo que normalizar o imputar con uno de estos valores antes de separar los conjuntos en entrenamiento y testeo contribuye con un fenómeno llamado contaminación accidental. Dicho fenómeno causa que las estimaciones de las métricas que realizamos para validar nuestros modelos no sean realistas, siendo más optimistas que el rendimiento verdadero que tendrá nuestro modelo.

Por los argumentos expuestos en el párrafo anterior, estas transformaciones se realizarán para los conjuntos de entrenamiento (por ejemplo, se toma la media y desviación estándar del conjunto de entrenamiento), y luego se transforma a los datos de testeo con los valores hallados para la transformación de los datos de entrenamiento.

Diagrama

Descripción generada automáticamenteEn la imagen a continuación, se pueden observar los operadores dentro del proceso de CV externo. Se puede apreciar que la media y moda (no hay un bloque de normalización debido al algoritmo random forest) se calculan para el conjunto de entrenamiento, y previo a realizar nuestras predicciones con el modelo en el conjunto de testeo, en el bloque Apply Model (3) se aplican las imputaciones deseadas.

**4- Algoritmos y modelos**

En cuanto al algoritmo de aprendizaje para generar el modelo de predicción de enfermedades del corazón, claramente debemos considerar los algoritmos supervisados de clasificación.

Utilizando los estudios Citados por el paper "Heart Failure: Diagnosis, Severity Estimation and Prediction of Adverse Events Through Machine Learning Techniques"[3] , diversos grupos de investigadores utilizaron diferentes algoritmos de ML, por lo que elegir uno a priori sin probar una gran variedad de ellos sería un enfoque bastante ingenuo y sesgado en mi opinión. Debido a esto, utilizaré una variedad de algoritmos y elegiré el que mejor se adecúe al dataset elegido.

Como primer paso, propongo utilizar Naive bayes como benchmark inicial de performance. Este utilizará la corrección de Laplace para prevenir el error provocado por el caso de que exista un valor en el set de test que no haya estado en el set de entrenamiento. Me parece pertinente remarcar que se utilizará sampling estratificado en el bloque de cross validation (siempre utilizaré este tipo de sampling, pero este es especialmente importante para Naive Bayes). Luego, consideraré una serie de algoritmos utilizados por los investigadores de los papers consultados.

Dentro de estos, se podría justificar no utilizar knn por un consumo alto de memoria debido a un dataset de entrenamiento grande, pero con los bajos costos de memoria de hoy en día, este no debería ser un impedimento para un conjunto de datos que pese un par de megas. El orden cuadrático para realizar predicciones (n\*(n-1)/2) igualmente podría suponer un problema. Como nuestro dataset no es uno muy grande (no supera los 1000 ejemplos), no habría que descartar este algoritmo. Si utilizaramos otra herramienta como Python-scikitlearn, se debería aplicar one-hot encoding para los predictores categóricos, pero en rapidminer esto no es necesario, ya que se puede utilizar una mixed measure llamada mixed euclidean distance. Para los valores nominales, la distancia es 0 si ambos valores son iguales y 1 si son diferentes. Para valores numéricos, se utiliza la distancia euclídea. Por esto, para que todos los predictores tengan el mismo rango, se normaliza con la transformación min-max (de 0 a 1)

Luego de obtener las métricas para el modelo de KNN, utilizaré difentes tipos de ensambles inicialmente Random Forest y alguna variante de Boosted Trees como AdaBoost o GradientBoostedTrees (Rapidminer) debido a su alta performance para casos de la industria estudiada. Además, como otra justificación para utilizar estos métodos, no existe un requerimiento de que las predicciones tengan que ser extremadamente rápidas, por lo que a utilización de un modelo de Random Forest con miles de árboles no sería un problema. Como los boosted trees son muy susceptibles a outliers, se deberá analizar cuidadosamente la presencia de estos.

SVM es un algoritmo que no podrá ser utilizado para este caso en específico pues se poseen atributos categóricos, no soportados por este modelo.

Como este es mi último caso de uso, decido utilizar toda la batería de conceptos aprendidos en el curso para generar modelos robustos con una validación correcta con la menor contaminación posible. También, se tomarán en cuenta las distintas métricas que podemos obtener de cada modelo para hacer las comparaciones de los modelos obtenidos

Como en este caso un falso negativo para la clase 1 (predecir que un paciente no posee la enfermedad cuando en realidad la tiene) es extremadamente costoso en cuanto a costos monetarios y la vida de las personas, considero que lo más importante es obtener las mejores métricas posibles, sin importar la complejidad del modelo, y hay que tomar especial precaución en la sensitivity o recall del modelo. Es de suma importancia priorizar esta métrica para la clase 1.

Todos los bloques de cross validation utilizados en la solución, usan k=5 debido al altísimo costo computacional necesario para tanto entrenar los mejores modelos posibles (utilización de estrategias evolutivas para selección de atributos y para tuning de hiperparámetros) como aplicar una correcta validación de los mismos. Esta correcta validación implica bloques de cross validation anidados.

Naive Bayes

Diagrama

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

KNN

Diagrama

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Random Forest

Se optimizan los hiperparámetros:

1. subset\_ratio entre 0.2 y 1. ([4] Se recomienda utilizar la raíz de la cantidad de predictores)
2. number\_of\_trees entre 500 y 3000.

Diagrama

Descripción generada automáticamente

GradientBoostedTrees

Se optimizan los hiperparámetros:

1. Number\_of\_trees: de 50 a 500
2. Learning\_rate: de 0.05 a 1
3. Maximal\_depth: de 1 a 10

Diagrama

Descripción generada automáticamente

Primeros resultados con random forest: Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Resultados con gradientBoostedTrees

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**5- Conclusiones generales sobre el caso abordado y la viabilidad u oportunidad de aplicación de técnicas de machine learning en el mismo.**

Bibliografía

[1] <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>

[2] Application of Machine Learning Algorithms to Predict Coronary Artery Calcification With a Sibship-Based Design Yan V. Sun1,\* , Lawrence F. Bielak1, Patricia A. Peyser1, Stephen T. Turner2, Patrick F. Sheedy II3, Eric Boerwinkle4, and Sharon L.R. Kardia (2008)

[3] Heart Failure: Diagnosis, Severity Estimation and Prediction of Adverse Events Through Machine Learning Techniques Evanthia E. Tripoliti, Theofilos G. Papadopoulos , Georgia S. Karanasiou, Katerina K. Naka, Dimitrios I. Fotiadis (2016)

[4] https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/