MLMMCD (Machine Learning Model for Mortality from Cardiac Disease), Prediceinfartos-inador

Omar Steck Espinel Santamaria [1] , Daniel Santiago Rincón Poveda [2]
{espinelo [1] , daniel_rincon [2] }@javeriana.edu.co

Introducción

Las enfermedades cardiovasculares son un gran conjunto de enfermedades que tienen una relación directa con el corazón o el sistema circulatorio [1], estas enfermedades son clasificadas por [1] como: hipertensión arterial (presión alta), cardiopatía coronaria (infarto de miocardio), enfermedad cerebrovascular (apoplejía), enfermedad vascular periférica, insuficiencia cardíaca, cardiopatía reumática, cardiopatía congénita y miocardiopatías.

Las enfermedades cardiovasculares son la principal causa de muertes en el mundo, donde en el año 2015, el 31% de las muertes registradas a nivel mundial, fueron debido a estas causas[2], estas enfermedades aquejan en su mayoría a países con niveles de ingresos bajos y medios, y dentro de estos países, estas enfermedades toman vidas de hombres y mujeres en porcentajes similares[1]. En Colombia se presenta en [3] que para el año 2015 las muertes por enfermedad cardiaca le cuestan al país 6.4 billones de pesos cada año, y que por infartos al miocardio mueren 29000 personas por año, lo que representa 80 muertes en promedio diarias.

A lo largo de los años, los médicos han desarrollado varias escalas con las cuales evaluar la probabilidad de una persona de sufrir una falla de miocardio y sus probabilidad de sobrevivir a este, algunas de estas son:

- TIMI: La Trombosis en Infarto del Miocardio (siglas en inglés) es un grupo de estudio que desarrolló una escala de riesgo que evalúa la probabilidad de sufrir una falla de miocardio o la probabilidad de sufrir una segunda falla dentro de un periodo de tiempo generalmente establecido en un año. Esta escala tiene en cuenta factores como edad, uso de aspirina en los últimos 7 días, cambios de al menos 0.5 mm en el electrocardiograma, historial familiar y sexo. [4][5]
- GRACE: La escala GRACE (Regristro Global de Episodios de Síndrome Coronario Agudo, siglas en inglés) es un puntaje de riesgo que se basa en evidencias, esta escala permite estimar la probabilidad de poseer una enfermedad coronaria, el riesgo de infarto por el mismo motivo o la muerte hospitalaria dentro de seis meses posterior al diagnóstico. Esta escala tiene en cuenta características como: edad, presión arterial sistólica, frecuencia cardiaca, creatinina y escala Killip para falla cardiaca. [5][6]
- Clasificación Killip y Kimball: Es una estratificación creada por Thomas Killip y John Kimball, la cual es un resultado luego de la observación de 250 personas, esta clasificación permite establecer cuánto evoluciona una afección por infarto agudo del miocardio o la muerte en el primer mes luego de la falla. Esta clasificación toma en

cuenta propiedades como presencia de estertores, tercer ruido cardíaco, aumento de presión venosa yugular o edema pulmonar agudo.

Aunque estos porcentajes pueden ser calculados manualmente, en la actualidad se pueden crear modelos mediante Inteligencias Artificiales (IAs), estas tomarán un conjunto con los datos de los pacientes y a partir de ellos, con nuevos datos podrán predecir la probabilidad de deceso de la persona.

Con esto se puede evidenciar la importancia de la inteligencia artificial en la prevención de enfermedades cardiacas. Lo que este proyecto busca, es proveer un modelo de machine learning que pueda detectar el impacto de enfermedades cardiacas o de eventos relacionados a enfermedades cardiacas.

En este caso para el desarrollo del proyecto se usa la información disponible en [10] un conjunto de datos compuesto de 12 características clínicas que pueden relacionarse a eventos de falla cardiaca en el paciente, en especial eventos de falla cardiaca que terminan en la muerte del paciente.

De acuerdo a la persona que publica estos datos: *Las personas con enfermedades cardiacas o alto riesgo cardiovascular por la presencia de uno o más factores de riesgo necesitan detección temprana y tratamiento para lo que un modelo de machine learning puede ser de gran ayuda*[10].

El objetivo de este proyecto es producir un modelo de *machine learning*, que mediante las 12 características incluidas en el conjunto de datos, ofrezca una predicción con la mayor exactitud conseguible sobre el evento de muerte por falla cardiaca para un paciente con esas características. Esto se logra mediante el análisis de una gran variedad de casos con las características mencionadas y el desenlace fatal o no de una falla cardiaca en el paciente.

Nota: antes de comenzar el desarrollo se considera oportuno aclarar que la información aquí expuesta es de acceso público y los datos usados en este proyecto se basan enteramente en datos clínicos anónimos, donde no se revela información que permita identificar o revelar la identidad del paciente.

Desarrollo

Esta sección utiliza términos técnicos de machine learning, programación y ciencia de datos para los cuales se asume que el lector posee los conocimientos básicos para comprenderlos y por ende no se provee un glosario ,marco teórico o explicación detallada de los términos técnicos aquí incluidos. El conocimiento y comprensión de dichos conceptos quedan a cargo del lector.

Análisis de los datos:

Al momento de leer y analizar los datos, se identificaron las doce características que los conforman y la etiqueta de salida de estos datos. a continuación se muestran las 12 características con una breve descripción de cada una y acompañada por el tipo de característica entre cualitativa y cuantitativa de acuerdo a [10]:

- **Age**: Edad del paciente en años, característica cuantitativa.
- **Anaemia**: Anemia, decremento de células o hemoglobina en la sangre del paciente, característica cuantitativa de tipo booleano 1 para presente 0 para ausente.
- **Creatinine_phosphokinase**: Cantidad de la enzima CPK en la sangre, característica cuantitativa (mcg/L).
- **Diabetes**: Padecimiento de diabetes por parte del paciente, característica cuantitativa de tipo booleano, 1 para diabetes 0 para paciente sin la enfermedad.
- **Ejection** Fraction: Porcentaje de la sangre que deja el corazón en cada contracción, característica cuantitativa porcentaje.
- **High_blood_pressure**:Hipertensión del paciente, característica cuantitativa de tipo booleano, 1 si el paciente padece hipertensión, 0 si no.
- **Platelets**: plaquetas en la sangre del paciente , característica cuantitativa (kilo plaquetas/ml).
- **Serum_creatinine**: Cantidad de suero de creatinina en la sangre, característica cuantitativa (mg/dL).
- **Serum_sodium**: Cantidad de suero de sodio en la sangre, característica cuantitativa (mEq/L).
- **Sex**: Sexo del paciente, se representa mediante una característica cuantitativa booleana 0 si es mujer 1 si es hombre.
- **Smoking**: Si el paciente fuma o no, característica cuantitativa booleana 1 si fuma 0 si no.
- **Time**: Cantidad de días de seguimiento al paciente después de sufrir el ataque cardíaco para conocer el estado del paciente, característica cuantitativa.

Así mismo se describe la etiqueta:

- **DEATH EVENT**: Evento de fallecimiento del paciente durante el tiempo de seguimiento, característica cuantitativa booleana 1 si el paciente fallece 0 si sobrevive.

Algunos elementos importantes a destacar sobre el conjunto de datos son:

- Todos los datos tanto las características como la etiqueta son numéricos, incluso los cualitativos son booleanos con valores de cero y uno, por lo que no es necesario transformar datos cualitativos a cuantitativos en el acondicionamiento de los datos.
- Todos los datos están completos, esto quiere decir que no hay filas con información faltante de un paciente.
- La etiqueta de salida es un valor binario, por lo que el modelo debe predecir un evento binario.

Selección del tipo de problema y solución:

De acuerdo con lo visto en el dataset se genera un análisis sobre el tipo de problema para este caso, donde se identifica el tipo de solución que se busca en base a los datos.

Dado que se busca conocer el valor de una etiqueta de salida a partir de la información obtenida de las características de entrada, el modelo propuesto debe ser un **clasificador**.

Conociendo que la etiqueta de salida del modelo toma solo valores binarios, el modelo propuesto es **de una sola clase**.

Finalmente dado que la información suministrada para el entrenamiento posee las etiquetas de salida para distintas características de entrada el modelo de aprendizaje será **supervisado**.

A partir de esta información se sabe que se desea obtener un clasificador de una sola clase, que sea entrenado mediante un algoritmo supervisado, una vez identificadas las características del modelo se procede a buscar el algoritmo que cumpla con la mayor precisión para esta aplicación y que cumpla con estas características.

Acondicionamiento de los datos:

Para entrenar y seleccionar alguno de los algoritmos posibles es necesario entrenar los algoritmos con los datos para evaluar el desempeño del modelo, por esto es necesario cargar el dataset, no obstante antes de usar estos datos para entrenar es necesario realizar modificaciones a la información para no solo hacerla usable para las diferentes funciones sino hacer que esta esté dispuesta de la mejor forma para obtener el mayor desempeño de los modelos.

Para esto el procedimiento usado es:

1. Se carga la información en variables que puedan ser leídas y modificadas según sea necesario.
2. Realizar la partición de los datos, con el fin de hacer validación cruzada y obtener una mejor medida sobre cómo se desempeña el modelo fuera de los datos de entrenamiento se selecciona un porcentaje de los datos para realizar las pruebas, en este caso se realiza una partición que se considera óptima, de acuerdo a la experiencia de los realizadores del proyecto 70% de los datos para entrenamiento y 30% de los datos para validación. Así mismo se realiza la separación entre características y etiquetas.
3. Se escalizan los datos con el objetivo de normalizar los diferentes valores de las características y obtener mejor desempeño sobretodo en algoritmos que trabajen por distancia, haciendo que cada característica contribuya de manera proporcional a la distancia final para encontrar el valor deseado.

   Con el objetivo de encontrar el mejor escalado para los datos, se prueban tres diferentes localizaciones, para las cuales se ejecutan los diferentes modelos y se selecciona el escalado que aporta mejores resultados de acuerdo con los criterios establecidos.

Métricas usadas para el desempeño y validación del modelo:

En el modelo debido a la naturaleza de la aplicación se consideran importantes tres cosas:

- Que el modelo permite predecir el fallecimiento o no del paciente con respecto a unas características con exactitud, maximizar la cantidad de verdaderos positivos.
- Que la cantidad de falsos negativos sea mínima.
- Es preferible obtener un falso positivo que un falso negativo, ya que un falso positivo no está en riesgo de fallecer, mientras que un falso negativo si.

Debido a esto se seleccionan tres métricas a tener en cuenta en el proyecto:

- La accuracy: como medida de cuantos aciertos positivos se obtiene.
- El Matthews Correlation Coefficient (MCC): como medida de los aciertos de clasificación del modelo tanto positivos como negativos.
- Curva ROC: como métrica de desempeño del clasificador para los umbrales de clasificación, mostrando la relación entre positivos y negativos correctos del modelo.

La forma de validación consiste en entrenar el algoritmo con los datos de prueba, realizar predicciones de los datos de entrenamiento y comparar los resultados de los datos de entrenamiento con las etiquetas verdaderas de ese conjunto para evaluar las diferentes métricas seleccionadas.

Algoritmos usados y selección de hiper parámetros:

Dado que se requiere un algoritmo clasificador supervisado de una sola clase, se listan los diferentes algoritmos conocidos por los encargados del proyecto que cumplen con estas características, estos son:

- Logistic regression
- SVM (Support Vector Machines)
- KNN (k-nearest neighbors)
- ANN (Artificial Neural Networks)

Para el primer algoritmo la *logistic regression*, se seleccionan los hiperparametros que permitan obtener el mejor desempeño de este algoritmo.

Para lograr el mejor desempeño, se escoge una penalización de tipo l2 ya que se considera apta para este problema, se usa un random state para poder obtener siempre la misma respuesta de forma consistente y para la regularización, se selecciona un valor entre 0.1 y 2, ya que para valores menores y mayores a este el desempeño del MCC para el algoritmo se mantiene estable siendo menor que el valor en esta franja.

Para obtener el mejor valor de regularización se prueban 200 puntos diferentes en el intervalo mencionado y se analiza el valor de MCC del algoritmo seleccionando el valor que consiga un MCC mayor.

En el caso de las Support Vector Machines, el hiper parámetro a seleccionar es el kernel de la función, para ellos se entrena el algoritmo de SVM con los datos de prueba para todos los kernel,se evalúa su accuracy y su MCC con los datos de entrenamiento y al final se selecciona el algoritmo con el kernel que provee un valor máximo de accuracy y SVM en caso de un empate se da más peso a un algoritmo con un mejor MCC siempre y cuando la diferencia en accuracy no sea mayor a 5 puntos porcentuales.

Para KNN es necesario escoger el número de k que se van a tomar para calcular la distancia de los puntos, con el objetivo de obtener el mejor k para la aplicación, se grafica el desempeño de diferentes k desde 1 hasta 30 para seleccionar el k que presente mayor desempeño y se usa ese k para el entrenamiento del modelo una vez se entrena con el mejor k, se evalúa el desempeño del modelo par los datos de entrenamiento y validación.

Para la ANN, hay más de un elemento que seleccionar para los hiperparametros, para la cantidad de capas se escoge una arquitectura con 3 capas escondidas, ya que se considera que tres capas es una cantidad suficiente para darle a la solución la no linealidad suficiente en caso de ser necesaria para obtener el mejor desempeño de clasificación.

Para la selección de neuronas se usa un criterio de ⅔ el tamaño de la capa de entrada +1, tomado de [11]

Para la función de activación de las neuronas se prueban diferentes formas de activación provistas por la librería y se evalúa el desempeño para esa función de activación, seleccionando la función que permita obtener el mejor desempeño en las dos métricas.

La selección del algoritmo usado se realiza escogiendo el mejor valor para ambas métricas de desempeño y una ROC con área bajo la curva mayor.

Resultados

Escalado:

Primero para probar el escalado se prueban los diferentes algoritmos usando diferente escalado.

Esta medición se realiza dos veces una con hiperparametros asumidos y una segunda vez con los hiper parámetros optimizados para obtener el mejor desempeño, en ambos casos el resultado del escalado con mejor desempeño es el mismo, a continuación se muestran los datos tomados con los hiperparametros optimizados.

	Estándar		MinMax		Robust	
	MCC	ACC	MCC	ACC	MCC	ACC
LR	0.52	0.78	0.49	0.77	0.49	0.77
SVM	0.58	0.81	0.49	0.77	0.48	0.77
KNN	0.49	0.77	0.28	0.69	0.42	0.74
ANN	0.48	0.77	0.46	0.76	0.32	0.70

De acuerdo con estos resultados el mejor método de escalado para la aplicación es el escalado estándar, por lo que los resultados a continuación usan

Logistic Regression

Para seleccionar el mejor valor para la regularizacion se crea una función que recorra un arreglo, al ejecutarla se encuentra el mejor valor de regularización:

```
Mejor MCC de  0.5228131741408373

Conseguido con regularización de 0.415075376884422
```

SVM

Para seleccionar el mejor kernel de SVM se mide el desempeño de los diferentes kernel y se selecciona el mejor, al obtener los datos de desempeño se obtienen los siguientes datos:

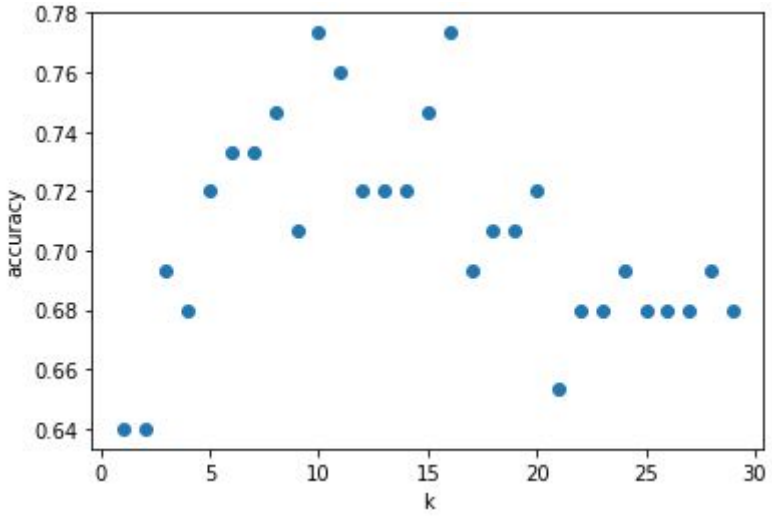
```
**RESULTS FOR  linear  KERNEL***
```

```
matthews_corrcoef linear 0.5228131741408373
Accuracy linear 0.7866666666666666

**RESULTS FOR  poly 2 KERNEL***
matthews_corrcoef poly degree 2 0.06496495357242561
Accuracy poly degree2 0.6266666666666667

**RESULTS FOR  poly 3 KERNEL***
matthews_corrcoef poly degree 3 0.40688576184834985
Accuracy poly degree3 0.7333333333333333

**RESULTS FOR  sigmoid  KERNEL***
matthews_corrcoef sigmoid 0.5541466051858356
Accuracy sigmoid 0.8

**RESULTS FOR  rbf  KERNEL***
matthews_corrcoef rbf 0.5850371197585715
Accuracy rbf 0.8133333333333334
```

Para este algoritmo los mejores valores en desempeño se obtienen con el kernel RBF, por lo que se selecciona este kernel como el mejor para SVM en esta aplicación.

KNN

En el caso de KNN se selecciona el primer K con el mayor accuracy, para ello se grafica el accuracy para diferentes valores de k en el intervalo 1 a 30



Se imprime el valor a usar y su accuracy:

```
Max score  0.7733333333333333

found at k =  10
```

ANN

Finalmente para la ANN se prueban las diferentes funciones de activación y se guardan los valores de accuracy y MCC para estas con el fin de seleccionar la mejor, los valores obtenidos en las métricas seleccionadas se muestran a continuación:

	ANN Estandar	
	MCC	ACC
Relu	0.48	0.77
Sigmoid	0.43	0.74
Softmax	0.44	0.74
Tanh	0.41	0.75

Por lo que se escoge finalmente la activación Relu para la red neuronal.

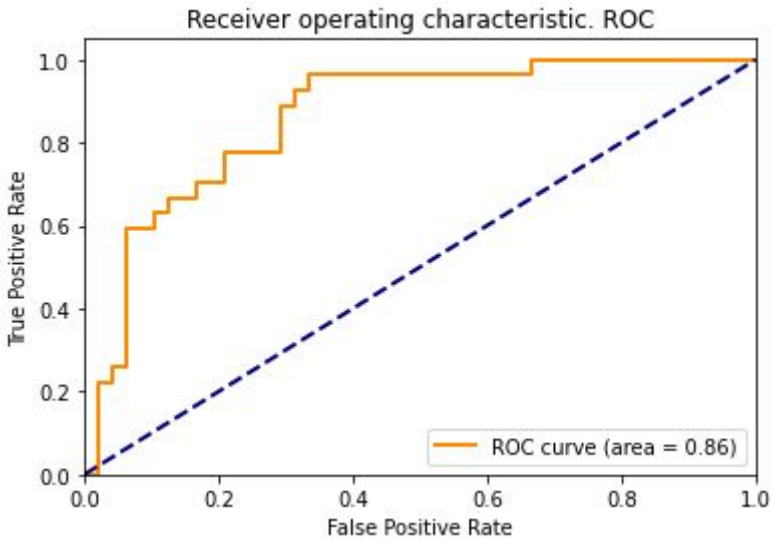
Finalmente con todos los hiper parámetros y el escalado seleccionados se ejecutan de nuevo los algoritmos para obtener el mejor algoritmo para entrenar el modelo, se guardan los datos y se muestran a continuación para comparación los diferentes valores de MCC y ACC

	Estándar	
	MCC	ACC
LR	0.52	0.78
SVM	0.58	0.81
KNN	0.49	0.77
ANN	0.48	0.77

Seleccionando el algoritmo SVM con kernel RBF como el algoritmo con mejor desempeño para la aplicación con un MCC y accuracy de:

```
matthews_corrcoef rbf 0.5850371197585715

Accuracy rbf 0.8133333333333334
```

Con una ROC de:



Que no solo tiene una buena proporción de TPR y FPR sino que adicionalmente posee la ROC más grande entre los diferentes algoritmos.

Conclusiones

Al final se obtiene un algoritmo con la capacidad de detectar con un 81% de accuracy si un paciente con unas características específicas fallece a causa de sufrir una falla cardiaca, aunque es deseable un valor mayor de accuracy, este ha sido el máximo conseguible para esta aplicación.

Teniendo en cuenta el Accuracy, la ROC y el MCC de 0.58 en el modelo, se puede afirmar que el modelo es capaz de clasificar de forma correcta ya sea como positivo o como negativo a la mayoría de los pacientes, lo cual sirve como una herramienta útil para la detección temprana de riesgos mortalidad asociados a falla cardiaca, por ende se considera logrado el objetivo

Dada la aplicación se considera más importante que no haya falsos negativos (la predicción de que un paciente no está en riesgo de padecer un ataque cardiaco letal, cuando en realidad sí lo está) que falsos positivos, por lo que la selección del modelo tiene esto en cuenta.

Una de las razones para escoger MCC sobre F1 es debido a su fórmula, mientras que el F1 analiza los casos positivos, lo cual es bastante útil, el MCC analiza también los casos negativos. Esto permite obtener una mayor certeza sobre los negativos, ya que queremos disminuir el número de falsos negativos así como tener un buen número de verdaderos positivos.

Después de repetir las pruebas por todos los clasificadores con las diferentes escalizaciones, se verifica que la escalizacion estándar es la más adecuada para la aplicación, esto tiene sentido dado que para los diferentes datos médicos provistos tienen diferentes significados los valores, como un valor alto en una característica es positivo mientras en otra lo contrario, tras realizar la verificación contra el robust que centralizaba y

removía la mediana, se comprueba que estos datos son significativos, así que se escoge la escalizacion estándar.

Referencias

[1] Organización Mundial de la Salud, “¿Qué son las enfermedades cardiovasculares?” https://www.who.int/cardiovascular_diseases/about_cvd/es/ (accessed Nov. 24, 2020).

[2] Federación Española del Corazón, “Las cifras de la enfermedad cardiovascular,” 2018. https://fundaciondelcorazon.com/blog-impulso-vital/3264-las-cifras-de-la-enfermedad-cardiovascular.html (accessed Nov. 24, 2020).

[3] World Heart Federation y Sociedad Colombiana de Cardiología y Cirugía Cardiovascular, “ENFERMEDAD CARDIOVASCULAR.” [Online]. Available: https://www.world-heart-federation.org/wp-content/uploads/2017/11/infografia-WHF.pdf.

[4] Wikipedia, “TIMI.” https://en.wikipedia.org/wiki/TIMI (accessed Nov. 24, 2020).

[5] ““Validación de las escalas TIMI y GRACE para el síndrome coronario agudo en una cohorte contemporánea de pacientes’’,” Acta Medica Colomb., 2015.

[6] Wikipedia, “Escala de GRACE.” https://es.wikipedia.org/wiki/Escala_de_GRACE (accessed Nov. 24, 2020).

[7] Wikipedia, “Clasificación de Killip y Kimball.” https://es.wikipedia.org/wiki/Clasificación_de_Killip_y_Kimball (accessed Nov. 24, 2020).

[8] T. Killip and J. T. Kimball, “Treatment of myocardial infarction in a coronary care unit. A Two year experience with 250 patients,” Am. J. Cardiol., 1967, doi: 10.1016/0002-9149(67)90023-9.

[9] Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020). https://doi.org/10.1186/s12911-020-1023-5

[10] Larxel, “Heart Failure Prediction 12 clinical features por predicting death events“” 2020. https://www.kaggle.com/andrewmvd/heart-failure-clinical-data

[11] Erna Nababan,“How to decide the number of hidden layers and nodes in a hidden layer?” 2020. https://www.researchgate.net/post/How-to-decide-the-number-of-hidden-layers-and-nodes-in-a-hidden-layer#:~:text=The%20number%20of%20hidden%20neurons,size%20of%20the%20input%20layer