Predicting Infectious State of Hepatitis C Virus Affected Patient's Applying Machine Learning Methods

Estudiante 1: Juan Diego Gallego Giraldo
Código: 5993

Estudiante 2: Oscar Eduardo Buritica Atehortúa
Código: 17202

Sistemas Inteligentes 2

Ingeniería en Sistemas y Computación

Universidad de Caldas

Profesor: Felipe Buitrago Carmona

Manizales, 18 de octubre 2023

Contenido

1. Diseño del experimento

a. Definición del problema

El virus de la hepatitis C es una de las principales causas de enfermedades hepáticas en todo el mundo. Sin embargo, diversas investigaciones realizadas por científicos expertos en este tema intentan reducir la influencia de este virus que se encuentra tan presente.

Más de 160 millones de personas están infectadas por este virus y cada año mueren más de 35000 personas debido a este problema. Se trata de una enfermedad infecciosa que causa hepatitis crónica, carcinoma hepatocelular y cirrosis hepática. La hepatitis C ataca al hígado y genera acumulación cicatricial del tejido lo que se conoce como fibrosis.

Con el pasar del tiempo se han propuesto métodos no invasivos para detectar la fibrosis y cirrosis en pacientes con VHC para obtener información reproducible y precisa.

Para esta investigación se ha propuesto un modelo basado en Machine Learning o aprendizaje automático para explorar síntomas significativos y analizar la situación de los pacientes, permitiendo clasificar estados hepáticos de los pacientes infectados por el virus.

Los algoritmos utilizados son árboles de decisión (DT), K-nearest neighbor (KNN), Random Forest, Máquinas de soporte vectorial entre otros.

La muestra se realiza sobre pacientes egipcios que se encuentren en la UCI (1385) y a partir de estos se instancian variables de la enfermedad.

Para el submuestreo se utilizan las instancias de múltiples variables que afectan directamente al paciente y ayudan al modelo de aprendizaje automático a identificar características apropiadas. Estas evidencias se pueden encontrar en Descripción de columnas. Finalmente, se obtienen resultados y se han clasificado en grupos, donde el rango va de los que no presentan condiciones hepáticas hasta una condición hepática grave.

El resultado de la investigación clasifica a los pacientes, en lesiones que tenga el hígado durante el tratamiento según la escala histológica de referencia y determina el nivel de la fibrosis hepática que va de 0 a 4, donde:

F0: sin fibrosis
F1: fibrosis portal (**336**)
F2: pocos septos (**332**)
F3: muchos septos (**355**)
F4: cirrosis (**362**)

Nota: Grupo determinado + Condición hepática + Cantidad de paciente

En resumen, después del uso de varios modelos de ML, recibir los datos, tratarlos y realizar un balance con SMOTE, se realiza una clasificación para generar un subconjunto de datos y luego clasificarlos de mejor manera para aumentar la predicación del entrenamiento.

Esto permite determinar que KNN muestra el resultado más alto para las diferentes métricas de todos los conjuntos de datos. Este resultado ha sido útil para analizar a fondo y tomar decisiones frente a la enfermedad infecciosa del virus de hepatitis C.

Sin embargo, la tarea presente es comprobar si los resultados obtenidos realizando nuevamente la aplicación de los algoritmos coinciden y buscar la manera de optimizar la información para comprobar la hipótesis planteada.

b. Descripción de las columnas del Dataset.

Nombre de la variable	Tipo	Descripción
Gender	Binary	[Male], [Female]
Fever	Binary	[Absent], [Present]
Nausea/Vomting	Binary	[Absent], [Present]
Headache	Binary	[Absent], [Present]
Diarrhea	Binary	[Absent], [Present]
Fatigue & generalized bone ache	Binary	[Absent], [Present]
Jaundice	Binary	[Absent], [Present]
Epigastric pain	Binary	[Absent], [Present]
Age	Integer	
BMI	Integer	Body Mass Index
WBC	Integer	White blood cells
RBC	Integer	Red blood cells
HGB	Integer	Hemoglobin
Plat	Integer	Platelets
AST 1	Integer	aspartate transaminase ratio
ALT 1	Integer	alanine transaminase ratio 1 week

ALT4	Integer	alanine transaminase ratio 4 weeks
ALT 12	Integer	alanine transaminase ratio 12 weeks
ALT 24	Integer	alanine transaminase ratio 24 weeks
ALT 36	Integer	alanine transaminase ratio 36 weeks
ALT 48	Integer	alanine transaminase ratio 48 weeks
ALT after 24 w	Integer	after 24 warnings alanine transaminase ratio 24 weeks
RNA Base	Integer	Carga viral inicial
RNA 4	Integer	Carga viral después de 4 semanas
RNA 12	Integer	Carga viral después de 12 semanas
RNA EOT	Integer	RNA end-of-treatment
RNA EF	Integer	RNA Elongation Factor
Baseline histological Grading	Categorical	
Baselinehistological staging	Categorical	

c. Planteamiento de hipótesis o pregunta orientadora

A partir del proceso de selección de los síntomas más comunes según la OMS para la enfermedad de la hepatitis C, en los campos del dataset ¿Se podría determinar si la métrica de especificidad aumenta con respecto a las métricas ya aplicadas en la investigación, para los algoritmos de KNN, DT y MVS, y con la selección de estas características se podría confirmar que para la investigación los resultados mejorarán?

2. Desarrollo del experimento

a. Análisis estadístico de los datos

El análisis estadístico evidencia lo visto durante el semestre de la materia de Sistemas Inteligentes II hasta la fecha actual. Este análisis, lo dividimos en 3 partes: KNN**(1)**, Árboles de decisiones- DT**(2)** y Máquinas de soporte vectorial**(3)**.

1. **K-nearest neighbor (KNN)**

   Para la implementación de este algoritmo, se debe tener en cuenta lo que se realiza en el proceso.

   Como primera medida se ha cargado el Dataset para realizar una limpieza del mismo, encontrando que esta limpio, no hay muchos datos atípicos y no se encuentra con nulos que puedan interferir en el entrenamiento.

   Luego, se analiza el comportamiento del dataset implementando el algoritmo de **K-nearest neighbor (KNN)** para relacionar los datos obtenidos con los datos que contiene el artículo. Para todos los algoritmos implementados en la solución de la investigación las métricas daban relativamente baja, lo que hizo que se implementara una estrategia de balanceo llamada SMOTE, este algoritmo genera nuevas instancias sintéticas de la clase minoritaria y de esta manera equilibrar la muestra de datos y mejorar el rendimiento del modelo de clasificación.

   Para la aplicación del proceso, se obtienen las métricas a partir del dataset con todas las variables (Modelo original) menos las de tipo categorical que pueden afectar los resultados. Se obtienen las métricas (Accuracy, Precision, Recall, F1-score & especificidad) quedando similar e incluso mejor a los resultados planteados por el artículo. **(Fila 1. Imagen1)**

   Sin embargo, a pesar de que los resultados dan un panorama excelente de que vamos por buen camino, la aplicación del algoritmo SMOTE mejora bastante las métricas. Se obtienen las métricas después de utilizar el algoritmo de balanceo (Código en Colab). Se evidencia un aumento proporcional quedando casi perfecto y demostrando un entrenamiento óptimo para una excelente predicción. **(Fila 3. Imagen1)**

   En la **HIPÓTESIS**, se planteó que la OMS relaciona la enfermedad de hepatitis C con los siguientes síntomas:

TABLA OMS

- **Fiebre**
- **Cansancio**
- Pérdida de apetito
- **Nauseas y vomito**
- **Dolor abdominal**
- Orina oscura
- **Diarrea**
- **Dolor en las articulaciones**
- Color amarillento en los ojos y piel

Así que, se decide utilizar las variables en común de los **síntomas** que relaciona la **OMS** (Resaltados) con los presentes en el dataset y a partir de estos genera un nuevo dataset con variables numéricas y eliminando las variables categóricas para la implementación nuevamente el algoritmo de KNN.

Se esperaba que con menos columnas y sin el uso de variables numéricas las métricas que arroja el algoritmo debían ser mejores, pero después de realizar el proceso, se ha descubierto que ha pasado todo lo contrario. La disminución de columnas en el dataset implica una reducción notoria en los resultados de las métricas estimadas para el algoritmo utilizado. Este aplicado para el modelo con las variables utilizadas por la OMS y el mismo modelo aplicando SMOTE. **(Fila 2 y Fila 3).**

A continuación, se adjunta una tabla con los resultados de la descripción anteriormente mencionada:

Casos	Accuracy	Precision	Recall	F1-score	Especificidad	Manhattan	Euclidean	Minkowski
Modelo original sin variables: Baseline histological Grading, Baselinehistological staging	0,27075	0,2575	0,26	0,2575	0,26	0,282	0,27	0,27
Modelo con las variables utilizadas por la OMS	0,2527	0,2525	0,25	0,2475	0,25	0,253	0,253	0,253
Modelo original aplicando SMOTE	0,9958	0,995	0,995	0,995	0,995	0,996	0,995	0,996
Modelo con las variables de OMS y SMOTE	0,3955	0,4075	0,395	0,3925	0,395	0,395	0,395	0,395

Para soportar estos datos, se ha implementado el algoritmo con k = 4 donde k es la cantidad de vecinos que utiliza el modelo. Este número se debe a que después de varias pruebas, se identificó que con esta cantidad de vecinos van relacionados directamente con los 5 grupos identificados al comienzo del problema **(F0: sin fibrosis, F1: fibrosis portal, F2: pocos septos, F3:muchos septos, F4: cirrosis)**, pero el ajuste a 4 se debe a que no hay pacientes sin fibrosis reduciendo el grupo clasificatorio en 4 **(F1: fibrosis portal, F2: pocos septos, F3: muchos septos, F4: cirrosis).**

Para entender el tema de mejor manera, se adjunta las siguientes imágenes de cómo se comporta el algoritmo de acuerdo a cada vecino, y como en k = 4 se obtienen las mejores métricas para el dataset original **(Que no se le quitan las columnas)** mientras que para el dataset modificado **(Con las variables de la OMS- numéricas)** el comportamiento es bastante extraño donde las métricas no siguen la dependencia de k y varían.

Las siguientes imágenes contienen el experimento aplicando las distancias (Manhattan, Euclidean, Minkowski), que se utiliza para el aumento de métricas (ajustes), y que en este caso da el soporte para lo que se mencionó en el párrafo anterior.

- **Modelo original con distancias aplicadas:**

```
Experimento k= 1  distancia = manhattan  Accuracy= 0.22743682310469315
Experimento k= 1  distancia = euclidean  Accuracy= 0.23104693140794225
Experimento k= 1  distancia = minkowski  Accuracy= 0.22382671480144403
---------------------------------------
Experimento k= 2  distancia = manhattan  Accuracy= 0.23826714801444043
Experimento k= 2  distancia = euclidean  Accuracy= 0.2527075812274368
Experimento k= 2  distancia = minkowski  Accuracy= 0.22743682310469315
---------------------------------------
Experimento k= 3  distancia = manhattan  Accuracy= 0.2527075812274368
Experimento k= 3  distancia = euclidean  Accuracy= 0.26714801444043323
Experimento k= 3  distancia = minkowski  Accuracy= 0.23104693140794225
---------------------------------------
Experimento k= 4  distancia = manhattan  Accuracy= 0.2815884476534296
Experimento k= 4  distancia = euclidean  Accuracy= 0.27075812274368233
Experimento k= 4  distancia = minkowski  Accuracy= 0.2743682310469314
---------------------------------------
Experimento k= 5  distancia = manhattan  Accuracy= 0.26714801444043323
Experimento k= 5  distancia = euclidean  Accuracy= 0.23465703971119134
Experimento k= 5  distancia = minkowski  Accuracy= 0.22382671480144403
---------------------------------------
Experimento k= 7  distancia = manhattan  Accuracy= 0.259927797833935
Experimento k= 7  distancia = euclidean  Accuracy= 0.23826714801444043
Experimento k= 7  distancia = minkowski  Accuracy= 0.23104693140794225
---------------------------------------
```

❖ **Modelo original después de aplicar SMOTE con distancias aplicadas:**

```
Experimento k= 1  distancia = manhattan  Accuracy= 0.9965472593871385
Experimento k= 1  distancia = euclidean  Accuracy= 0.9963314630988347
Experimento k= 1  distancia = minkowski  Accuracy= 0.9963314630988347
---------------------------------------
Experimento k= 2  distancia = manhattan  Accuracy= 0.9961156668105309
Experimento k= 2  distancia = euclidean  Accuracy= 0.9956840742339231
Experimento k= 2  distancia = minkowski  Accuracy= 0.9961156668105309
---------------------------------------
Experimento k= 3  distancia = manhattan  Accuracy= 0.9961156668105309
Experimento k= 3  distancia = euclidean  Accuracy= 0.9958998705222271
Experimento k= 3  distancia = minkowski  Accuracy= 0.9963314630988347
---------------------------------------
Experimento k= 4  distancia = manhattan  Accuracy= 0.9961156668105309
Experimento k= 4  distancia = euclidean  Accuracy= 0.9958998705222271
Experimento k= 4  distancia = minkowski  Accuracy= 0.9963314630988347

Experimento k= 5  distancia = manhattan  Accuracy= 0.9961156668105309
Experimento k= 5  distancia = euclidean  Accuracy= 0.9958998705222271
Experimento k= 5  distancia = minkowski  Accuracy= 0.9961156668105309
---------------------------------------
Experimento k= 7  distancia = manhattan  Accuracy= 0.9958998705222271
Experimento k= 7  distancia = euclidean  Accuracy= 0.9956840742339231
Experimento k= 7  distancia = minkowski  Accuracy= 0.9958998705222271
---------------------------------------
```

❖ **Modelo con variables de la OMS con distancias aplicadas:**

```
Experimento k= 1  distancia = manhattan  Accuracy= 0.24548736462093862
Experimento k= 1  distancia = euclidean  Accuracy= 0.24548736462093862
Experimento k= 1  distancia = minkowski  Accuracy= 0.24548736462093862
---------------------------------------
Experimento k= 2  distancia = manhattan  Accuracy= 0.26714801444043323
Experimento k= 2  distancia = euclidean  Accuracy= 0.26714801444043323
Experimento k= 2  distancia = minkowski  Accuracy= 0.26714801444043323
---------------------------------------
Experimento k= 3  distancia = manhattan  Accuracy= 0.2779783393501805
Experimento k= 3  distancia = euclidean  Accuracy= 0.2779783393501805
Experimento k= 3  distancia = minkowski  Accuracy= 0.2779783393501805
---------------------------------------
Experimento k= 4  distancia = manhattan  Accuracy= 0.2527075812274368
Experimento k= 4  distancia = euclidean  Accuracy= 0.2527075812274368
Experimento k= 4  distancia = minkowski  Accuracy= 0.2527075812274368

Experimento k= 5  distancia = manhattan  Accuracy= 0.23104693140794225
Experimento k= 5  distancia = euclidean  Accuracy= 0.23104693140794225
Experimento k= 5  distancia = minkowski  Accuracy= 0.23104693140794225
---------------------------------------
Experimento k= 7  distancia = manhattan  Accuracy= 0.23465703971119134
Experimento k= 7  distancia = euclidean  Accuracy= 0.23465703971119134
Experimento k= 7  distancia = minkowski  Accuracy= 0.23465703971119134
---------------------------------------
```

- **Modelos con variables de la OMS después de aplicar SMOTE con distancias aplicadas:**

```
Experimento k= 1  distancia = manhattan  Accuracy= 0.37138541217091064
Experimento k= 1  distancia = euclidean  Accuracy= 0.37138541217091064
Experimento k= 1  distancia = minkowski  Accuracy= 0.37138541217091064
--------------------------------------
Experimento k= 2  distancia = manhattan  Accuracy= 0.3690116529995684
Experimento k= 2  distancia = euclidean  Accuracy= 0.3690116529995684
Experimento k= 2  distancia = minkowski  Accuracy= 0.3690116529995684
--------------------------------------
Experimento k= 3  distancia = manhattan  Accuracy= 0.3892965041001295
Experimento k= 3  distancia = euclidean  Accuracy= 0.3892965041001295
Experimento k= 3  distancia = minkowski  Accuracy= 0.3892965041001295
--------------------------------------
Experimento k= 4  distancia = manhattan  Accuracy= 0.3955545964609409
Experimento k= 4  distancia = euclidean  Accuracy= 0.3955545964609409
Experimento k= 4  distancia = minkowski  Accuracy= 0.3955545964609409
--------------------------------------
Experimento k= 5  distancia = manhattan  Accuracy= 0.40591281829952525
Experimento k= 5  distancia = euclidean  Accuracy= 0.40591281829952525
Experimento k= 5  distancia = minkowski  Accuracy= 0.40591281829952525
--------------------------------------
Experimento k= 7  distancia = manhattan  Accuracy= 0.4151920586965904
Experimento k= 7  distancia = euclidean  Accuracy= 0.4151920586965904
Experimento k= 7  distancia = minkowski  Accuracy= 0.4151920586965904
--------------------------------------
```

En resumen, el análisis estadístico para el algoritmo de KNN, se puede confirmar que su entrenamiento arroja resultados óptimos y una predicción casi que perfecta, la cual se puede ajustar mucho mejor con experimentos como el de la aplicación de las distancias. Asimismo, se confirma que la eliminación de columnas **NO** optimiza los resultados del entrenamiento, más por el contrario empeora las métricas y descontrolar el comportamiento del algoritmo de acuerdo a la cantidad de vecinos **(K)** que utilice.

2. Árboles de decisión (DT)

Este algoritmo se implementa para ver el comportamiento del dataset de acuerdo al modelo de árboles de decisión (DT). Esta permite dar un panorama diferente de otro entrenamiento de modelo de Machine Learning.

El proceso de ejecución es similar al de KNN, donde:

1. Se importan las librerías necesarias
2. Se carga el dataset
3. Se genera una copia del dataset para no afectar el original
4. Se aplica el algoritmo de árboles de decisión
5. Se saca la matriz de confusión
6. Se obtiene las matrices buscadas
7. Se genera el árbol gráfico
8. Se aplica SMOTE al dataset sin quitar variables
9. Se aplica SMOTE al dataset con las variables de la OMS
10. Se realizan la validaciones cruzadas

Después de la ejecución de este algoritmo podemos denotar su comportamiento de acuerdo con las características que se le han agregado y se obtiene :

Casos	Accuracy	Precision	Recall	F1-score	Especificidad	Validación cruzada
Modelo original sin variables: Baseline histological Grading, Baselinehistological staging	0,27436	0,2725	0,275	0,27	0,275	0,2368
Modelo original aplicando SMOTE	0,9838	0,985	0,985	0,9825	0,985	0,2483
Modelo con las variables utilizadas por la OMS (ver tabla OMS)	0,9827	0,98	0,9825	0,98	0,9825	0,2559
Modelo con las variables de OMS y SMOTE (ver tabla OMS)	0,4589	0,46	0,4575	0,46	0,4575	0,227

En primera instancia se demuestra con la implementación de DT que las métricas obtenidas concuerdan con las métricas de la investigación realizada por el equipo de científicos. Sin embargo con el uso del algoritmo SMOTE se pudo mejorar estas dejando en un rango casi perfecto. Esta información corrobora el trabajo realizado por ellos en el articulo.

Luego, para intentar mejorar estas métricas más de lo que se encuentran, aplicamos la misma idea que se aplicó para el modelo de KNN, donde se pretende que con la quitada de columnas, en este caso, dejando, las que se definió en tabla OMS, aumenta, pero lo que pasó fue lo mismo, las métricas disminuyeron considerablemente.

En el tratamiento de datos, hubo un problema presentando en esta implementación, las métricas al comienzo no se acercaron a lo que teníamos de muestra, quedando demasiado bajas, pero, después de investigar se encontró que con el uso del **parámetro "criterion"** se utiliza como función de medida de calidad de corte mejorando considerablemente el entrenamiento y permitiendo obtener las métricas de la tabla anterior.

De esta misma manera, se utilizó la **técnica de validación cruzada** que nos permite calcular la estimación de riesgo, es decir, que el modelo pueda cometer errores en datos no vistos. El árbol, se dividió en 5 subárboles o pliegues de muestra para calcular el riesgo de cada uno y luego se sacó el promedio **(Ver en la última columna de la tabla),** a mayor cantidad de

submuestras el riesgo es menor, pero en este caso a pesar de ser pequeño, 5/25 con esta nueva métrica se permitió evaluar mejor la calidad y la capacidad de generalizar el modelo y ayudó en la toma de decisiones para mejorar las demás métricas que no satisfacen el objetivo de la implementación.

ARBOL DE DECISION

Link Celda: Parcial 1 inteligentes arboles de decision.ipynb

3. Máquinas de soporte vectorial (SVM)

La aplicación de este algoritmo arroja resultados que se comportan en contra de los dos primeros algoritmos, ver la siguiente tabla:

Casos	Accuracy	Precision	Recall	F1-score	Especificidad
Modelo original sin variables: Baseline histological Grading, Baselinehistological staging	0,9782	0,98	0,9775	0,9775	0,9775
Modelo eliminando variables y aplicando SMOTE	0,2671	0,0675	0,25	0,105	0,2475

Para el entrenamiento de este modelo, es obligatorio la implementación del algoritmo SMOTE, el no aplicarlo, implica que los resultados sean nulos o no se de el entrenamiento correcto. Después de aplicar SMOTE los resultados obtenidos se ven reflejados en la tabla anterior.

Se evidencia que al dejar el dataset con todas las columnas y realizar el entrenamiento las métricas son bastante buenas, mientras que al eliminar columnas y dejar solo las que se ven en la tabla de OMS, hay una disminución bastante de las métricas.

Esto se debe a que el modelo interpreta todos los registros como pacientes con cirrosis, dando una mala predicción. Con el balanceo que produce SMOTE, se produce un espacio de más alta dimensión a nivel de datos o registros, adicionalmente en la investigación se menciona el uso del kernel RBF para ayudar en la transformación de los datos y llevar a cabo la clasificación de los datos.

Luego de tener mejor clasificados los datos, se utilizó el parámetro gamma: auto para controlar la función del kernel y evitar el **sobreajuste.**

b. Implementación de algoritmos

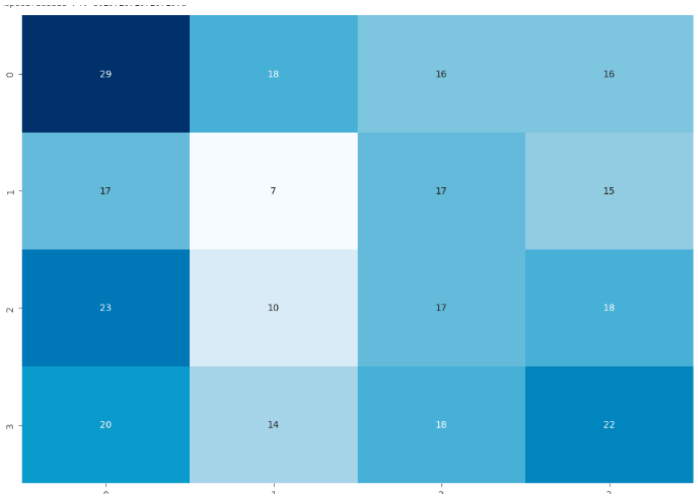
MODEL-ID	ALGORITMO	ACCURACY	PRECISSION	RECALL	F1-SCORE	NOTE
A001	KNN without SMOTE	**27.07%**	**25,5%**	**26%**	**25.75%**	
A002	KNN with SMOTE	**99.58%**	**99.5%**	**99.5%**	**99.5%**	Se aplicó el algoritmo **SMOTE** y luego KNN
A003	DT without SMOTE	**27.43%**	**27.25%**	**27.5%**	**27%**	se aplicó criterion='log_loss' para correr el árbol de decisión
A004	DT with SMOTE	**99.38%**	**98,5%**	**98.5%**	**98.25%**	se aplicó criterion='log_loss' para correr el árbol de decisión
A005	MVS without variable and with SMOTE	**27%**	**6,7%**	**25%%**	**10.5%**	Se aplico kernel **RBF**, luego la funcion de decision de forma **(OVR)** y finalmente se hizo de **gamma ='auto'**
A006	MVS with SMOTE	**98%**	**98%**	**98%**	**98%**	Se aplico kernel **RBF** luego la funcion de decision de forma **(OVR)** y finalmente se hizo de **gamma ='auto'**

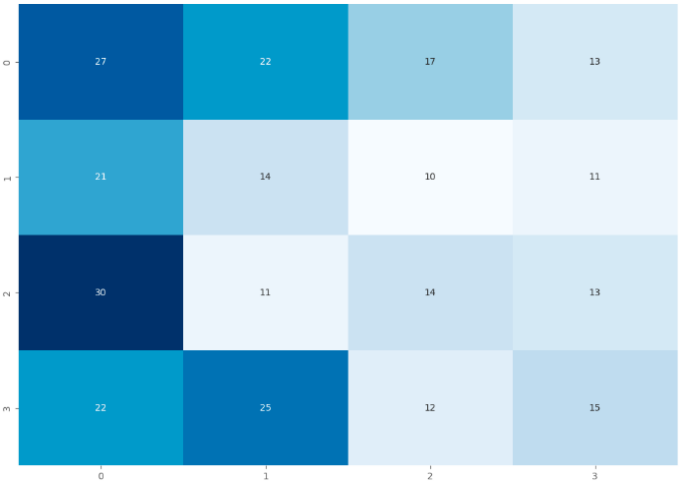
3. Análisis e interpretación de los resultados

a. Análisis comparativo sobre las matrices de confusión

1. Matrices de confusión para el algoritmo KNN:

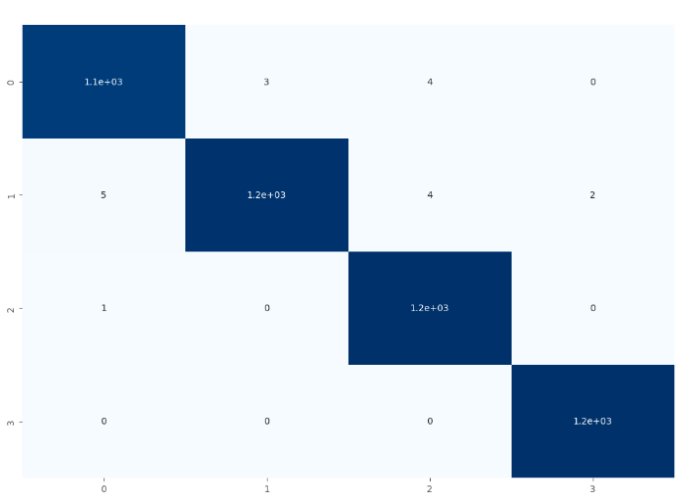
❖ **Modelo original sin SMOTE (1) VS Modelo modificado sin SMOTE (2)**

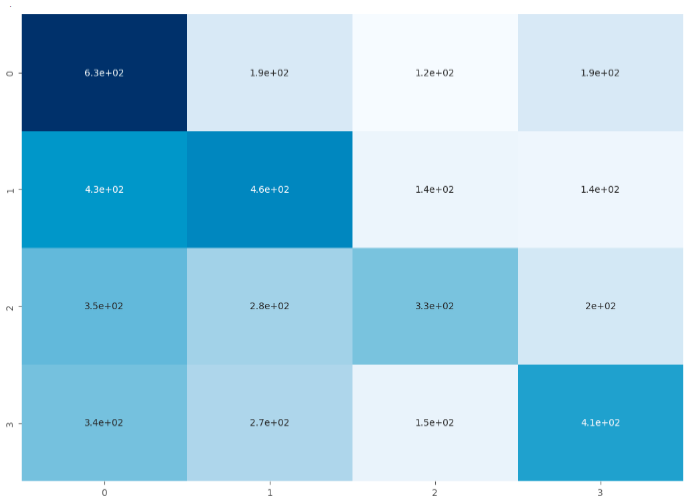




ANÁLISIS: En estos modelos se logra mostrar que el cambio de las variables del dataset por variables únicamente categóricas partiendo del modelo original como se muestra en las matrices de confusión, logra una reducción en los falsos positivos de cada categoría. Por lo tanto se puede evidenciar alguna mejora en el funcionamiento de algoritmo en su versión original, haciendo el cambio de las variables.

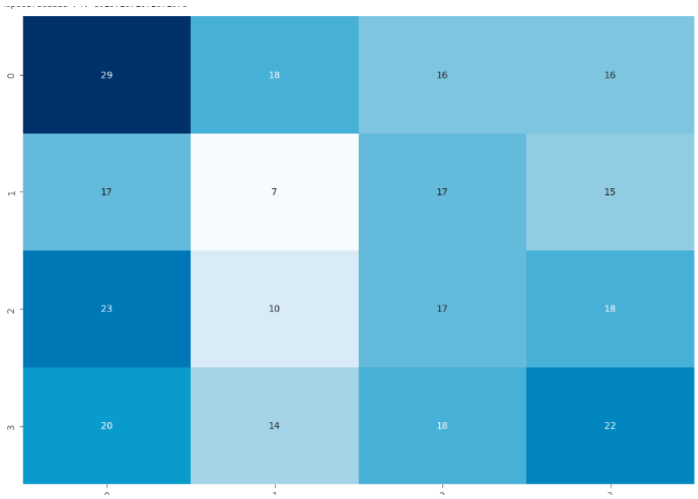
❖ **Modelo original después de aplicar SMOTE (1) VS Modelo con las variables de OMS-numéricas después de aplicar SMOTE (2)**

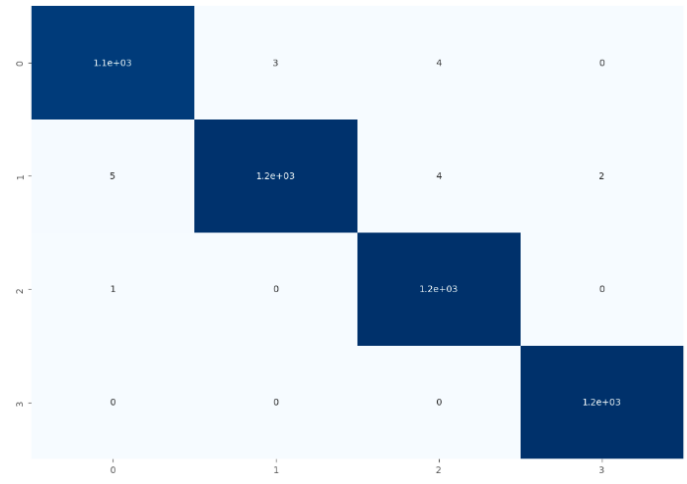




ANÁLISIS: En estos modelos haciendo el uso del algoritmo SMOTE, que se menciona en el artículo. se logra una mejora en el funcionamiento del modelo. Para lo cual se logra incrementar la precisión de las predicciones, sin embargo al realizar la eliminación de las variables del dataset y al correr el algoritmo SMOTE, se puede evidenciar que no logra incrementar las métricas que se plantearon originalmente.

❖ **Modelo original sin SMOTE (1) VS modelo original después de aplicar SMOTE (2)**

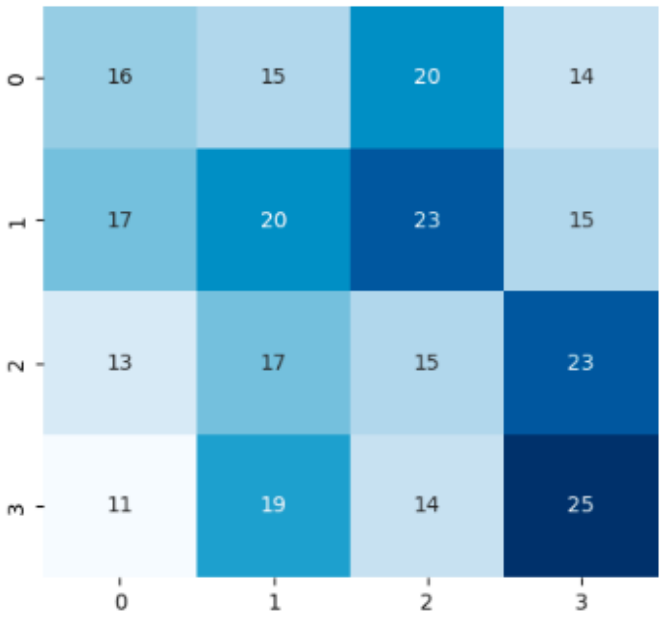


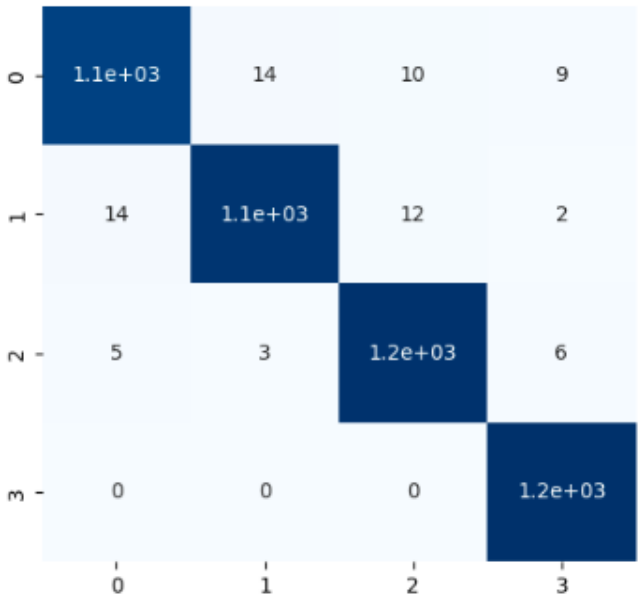


ANÁLISIS: En estos modelos se puede evidenciar que se logra mejorar los resultados del modelo haciendo uso de únicamente el algoritmo SMOTE, partiendo desde el uso de todas las variables.

2. Matrices de confusión para el algoritmo árboles de Decisión:

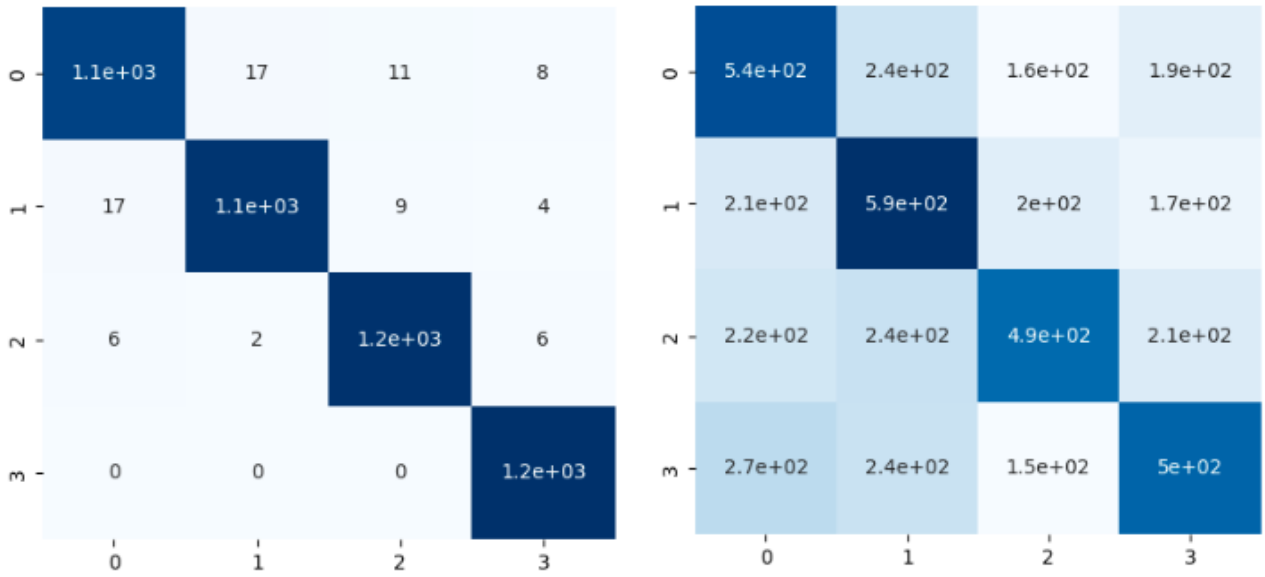
- **Modelo original sin SMOTE (1) VS modelo original después de aplicar SMOTE (2)**





ANÁLISIS: Para este modelo se puede evidenciar con el uso del algoritmo SMOTE, proporciona una mejora de en el modelo al hacer uso de este se reducen los falso positivos, y los falsos negativos, logrando así que los resultados del modelo funcionen, de manera adecuada.

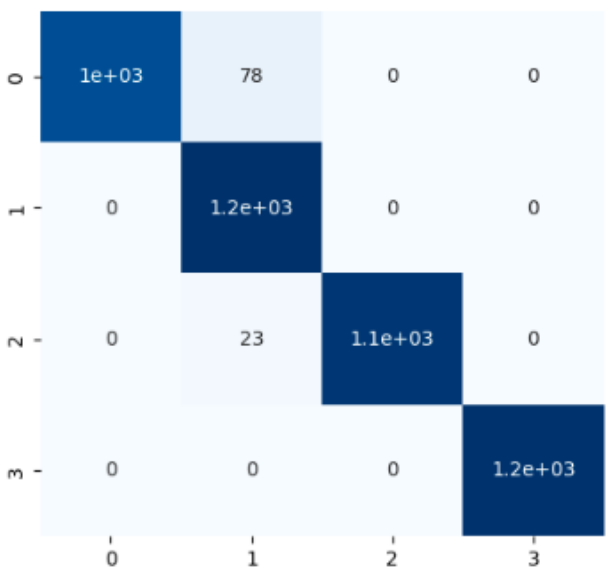
❖ **Modelo eliminando variables sin SMOTE (1) VS modelo eliminando variables después de aplicar SMOTE (2)**

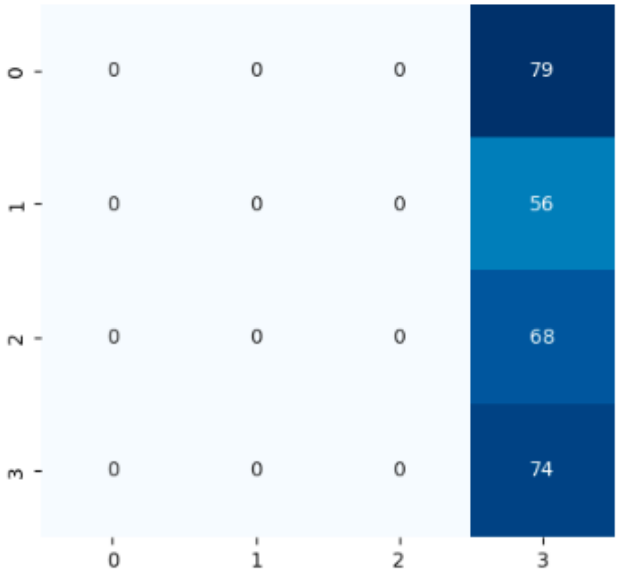


ANÁLISIS: Se tiene en cuenta que a partir de la eliminación de las variables, se logra mejorar el rendimiento del modelo para predecir el estado de enfermedad de los pacientes, sin embargo al aplicar el algoritmo SMOTE, el funcionamiento del modelo se empeora a comparación al modelo sin SMOTE, generando así que surjan falsos positivos y negativos, que no existían en el modelo sin las variables.

3. Matrices de confusión para el algoritmo de SVM:

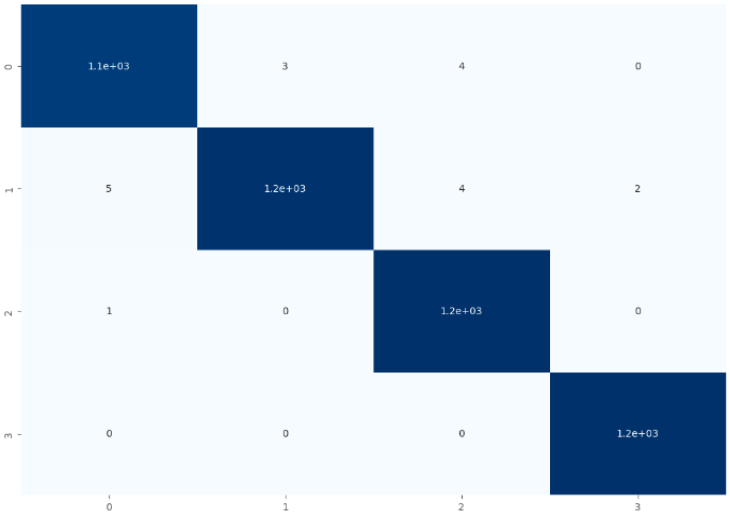
- **Modelo original con SMOTE (1) VS modelo eliminando variables y aplicando SMOTE (2)**

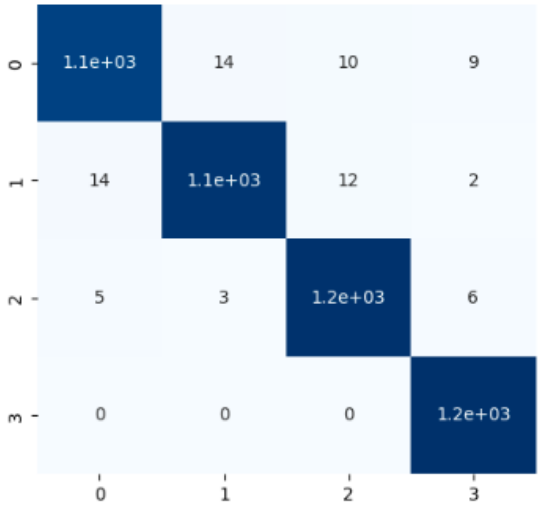


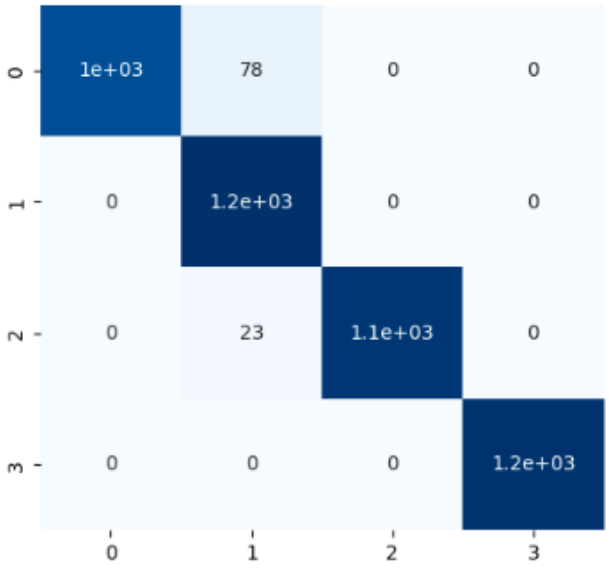


ANÁLISIS: En este modelo se tiene que al aplicar el algoritmo SMOTE, se muestra que el rendimiento logra tener una mejoría, sin embargo al contrario de lo realizado en los otros modelos, al aplicar SMOTE y eliminar variables en el SVM, se muestra que el modelo pierde toda la precisión y se convierte en un modelo equivocado, haciendo así que las predicciones solo muestre casos de pacientes con cirrosis.

4. **KNN vs DT vs SVM - Matrices originales con todas las variables y algoritmo SMOTE aplicado:**







ANÁLISIS: Para concluir se denota,que el algoritmo más eficiente como se mencionó en el artículo y se mostró en la tabla de resultados clasificadores, es el modelo de ML KNN. Como se propuso en la hipótesis, continúa siendo la mejor opción para predicción KNN en conjunto con SMOTE ya que es el modelo que presenta mejor rendimiento en la clasificación de los pacientes enfermos.

4. Recomendaciones basadas en el hallazgo del experimento

a. Conclusiones

1. Los resultados para la hipótesis no son exitosos. El algoritmo que arrojó mejores métricas como resultado en la investigación del artículo es K-nearest neighbor **(KNN).** A pesar de que después de usar **SMOTE** su métrica de especificidad es alta, se pretendía que con el uso de menos columnas en el dataset esta especificidad aumentara, sin embargo después del proceso de eliminación de columnas y la implementación del modelo, se observa que la especificidad como las demás métricas bajaron considerablemente. Se concluye que los mejores resultados del entrenamiento se dan cuando el dataset contiene más información, pero depende de un grupo de categorías menor **(grupos clasificados: F0… F4)** para los vecinos que utiliza y que a mayor vecinos los resultados también empeoran.

2. Los árboles de decisión mostraron que en el desarrollo de la hipótesis se logró un acercamiento a un incremento de las métricas eliminando las variables, sin embargo el uso del algoritmo **SMOTE**, genera en los árboles de decisión que empeore el rendimiento del modelo. Esto genera que no se logre realizar una comparación de este modelo con los otros modelos de ML, ya que al no lograr estas métricas con los demás modelos, se genera que solo sirva en árboles de decisión la eliminación de variables.

3. En máquinas de soporte vectorial se logró mostrar el fallo en la hipótesis, ya que al ser obligatorio aplicar el algoritmo **SMOTE** para este modelo como se presentó en el artículo, no se pudo asociar de manera adecuada la hipótesis planteada,pues al realizar la eliminación de las variables se disminuyeron las métricas considerablemente, dado que este algoritmo solo interpretó las entradas como pacientes con cirrosis, generando que el modelo falle y muestre resultados erróneos.

b. Posibles escenarios de fallos

En términos generales se puede determinar que los modelos pueden tener algunos fallos en la categorización de los estados de la enfermedad, sin embargo estas falencias se

encuentran en categorías cercanas lo que puede generar algunas confusiones al modelo. Adicionalmente para los modelos fue necesario eliminar las variables categóricas que se encuentran en el dataset, ya que con el uso de estas se genera un overfitting en algunos casos. Por lo tanto se decidió realizar las pruebas en todos los modelos de manera uniforme.

c. Problemas encontrados

Para KNN: En el uso de KNN, no se encontraron inconvenientes ya que como se mostró por medio de las tablas del artículo este modelo originalmente ofrecía valores bajos en las métricas, sin embargo con el uso del algoritmo **SMOTE** 8 veces como se mencionaba en el articulo, se logró mejorar las métricas sin ningún inconveniente.

Para DT: Es importante aclarar que durante la ejecución del modelo se realizó diferentes pruebas en las cuales se tiene el uso de diferentes niveles de profundidad del árbol, la cantidad de características y los criterios de la función de corte de este, por lo tanto,l se encontró que el criterio llamado "log loss" produce al modelo mejores resultados.

Para MSV: Para el uso de las máquinas de soporte vectorial, se realizó diferentes pruebas con los parámetros que permite el modelo, se probó los diferentes kernel que proporciona la herramienta, sin embargo se encontró que en el artículo se mencionó el uso del **kernel RBF**. Por otra parte se probó la función de decisión de la forma y se corroboró que el **"OVR"** obtuvo mejores resultados, esto con una **gama "auto"** y valores de aleatoriedad grandes.

5. Bibliografía

a. World. (2023, July 18). *Hepatitis C*. Who.int; World Health Organization: WHO. https://www.who.int/es/news-room/fact-sheets/detail/hepatitis-c

b. UCI Machine Learning Repository. (n.d.). Archive.ics.uci.edu. https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic

c. Prueba de hepatitis: Información en MedlinePlus sobre pruebas de laboratorio. (2016). Medlineplus.gov. https://medlineplus.gov/spanish/pruebas-de-laboratorio/prueba-de-hepatitis/

d. Kumar, S. (2022, August 2). Hepatitis C, crónica. Manual Merck Versión Para Profesionales; Manuales Merck. https://www.merckmanuals.com/es-us/professional/trastornos-hep%C3%A1ticos-y-biliares/hepatitis/hepatitis-c-cr%C3%B3nica

e. October 2020, M. D. 15. (2020, October 15). What is RNA? Livescience.com. https://www.livescience.com/what-is-RNA.html

6. Comunicación de resultados

a. Enlace repositorios

1. KNN	Parcial 1 Inteligentes 2 KNN.ipynb
2. Árboles de decisión	Parcial 1 inteligentes arboles de decision.ipynb
3. MSV	Parcial 1 inteligentes MSV.ipynb
4. Github	https://github.com/juandiegou/Parcial1SistemasInteligentes2

b. Enlace video

Link: https://youtu.be/oSy5DXotA5A