**Proyecto de Simulación: Enfermedad de Tiroides**

**Danny Francisco Hernández Godoy, Daniel Martínez Galeano**

**dfrancisco.hernandez@udea.edu.co, daniel.martinezg@udea.edu.co**

**Prof. Julián David Arias Londoño**

**julian.ariasl@udea.edu.co**

**Facultad de Ingeniería**

**Universidad de Antioquia, Colombia**

***Abstract*—La enfermedad de tiroides está dividida en dos diferentes padecimientos, puede causar muchos cambios en el cuerpo humano y es difícil de identificar quién puede padecerla o no. En este artículo se presenta diferentes aproximaciones de este problema de clasificación y finalmente se compara y se discute los resultados.**

***Keywords—modelo, selección, extractor, validación, tiroidea, clasificación, caracteristicas.***

1. **INTRODUCCIÓN**

La tiroides es una pequeña glándula con forma de mariposa ubicada en la base del cuello y es parte de una red de glándulas llamadas sistema endocrino[2]. Esta produce hormonas que regulan el metabolismo del cuerpo, cuando una persona tiene problemas con esta glándula puede presentar los siguientes síntomas: (inquietud, nerviosismo, corazón acelerado, irritabilidad, aumento de la sudoración, sacudida, ansiedad, problemas para dormir, piel delgada, cabello y uñas quebradizas, debilidad muscular, pérdida de peso, fatiga, piel seca, aumento de la sensibilidad al frío, problemas de memoria, estreñimiento, depresión, aumento de peso, debilidad, frecuencia cardíaca lenta, coma).

El campo en que se presenta es la vida, referente al área de la salud y el problema que se quiere abordar es determinar si un paciente referido a la clínica padece hipotiroidismo. Este problema corresponde a un problema de machine learning tipo clasificación, ya que sus valores son finitos y podemos categorizar las salidas.

Por tanto se utilizara para este experimento la base de datos de Thyroid Disease obtenida de la UCI [1].

1. **DESCRIPCIÓN**

Las base de datos de UCI provee la siguiente información sobre el conjunto de datos de enfermedad de tiroides:

Number of classes: 3

Number of learning examples: 3772

Number of testing examples: 3428

Variables de entrada del sistema:

Número de características 21.

15 Binarias.

6 continuas.

**Variables a predecir:**

El problema es determinar si un paciente referido a la clínica padece de hipotiroidismo por lo tanto, se construyen tres clases: normal (no hipotiroideo), hipotiroideo y funcionamiento subnormal. Porque el 92 por ciento de los pacientes No son hipotiroideos, un buen clasificador debe ser significativamente mejor que el 92%.

Nombre de los atributos[2]:

TABLE I

NOMBRE ATRIBUTOS

|  |  |
| --- | --- |
| **Nombre Atributo** | **Posible Valor** |
| age: | continuous. |
| sex: | 0, 1 |
| on thyroxine: | 0, 1 |
| query on thyroxine: | 0, 1 |
| on antithyroid medication: | 0, 1 |
| sick: | 0, 1 |
| pregnant: | 0, 1 |
| thyroid surgery: | 0, 1 |
| I131 treatment: | 0, 1 |
| query hypothyroid: | 0, 1 |
| query hyperthyroid: | 0, 1 |
| lithium: | 0, 1 |
| goitre: | 0, 1 |
| tumor: | 0, 1 |
| hypopituitary: | 0, 1 |
| psych: | 0, 1 |
| TSH: | continuous. |
| T3: | continuous. |
| TT4: | continuous. |
| T4U: | continuous. |
| FTI: | continuous. |

**Salidas nombre y valor:**

normal (no hipotiroideo) 3

hipotiroideo 2

funcionamiento subnormal. 1

Se clasifican las variables de entrada en continuas y lógicas.

La base de datos no cuenta con valores faltantes, lo que hace el desarrollo del problema más fácil de abordar.

1. **ESTADO DEL ARTE**

*A. Prediction of Thyroid Disease Using Data Mining Techniques*

El artículo [4] presenta 4 diferentes modelos de clasificación que son Naive Bayes, Decision Tree, Multi Layer Perceptron, RBF Network, para las cuales aplican 2 diferentes técnicas de aprendizaje que son Naive Bayes Learner, Decision Tree Learner.

Además de esto el nodo para validar el modelo es Scorer. El nodo informa una matriz de confusión con las medidas de calidad. También presenta una normalización de los datos para poder usarlos en el MLP y RBF Network.

Estos nodos corresponden a una red neuronal de clasificación el cual implementa para validar los modelos con WEKA; Está siendo una herramienta de software para machine learning, el cual ayuda a mejorar algunos parámetros de los modelos.

La tabla siguiente son los resultados obtenidos:

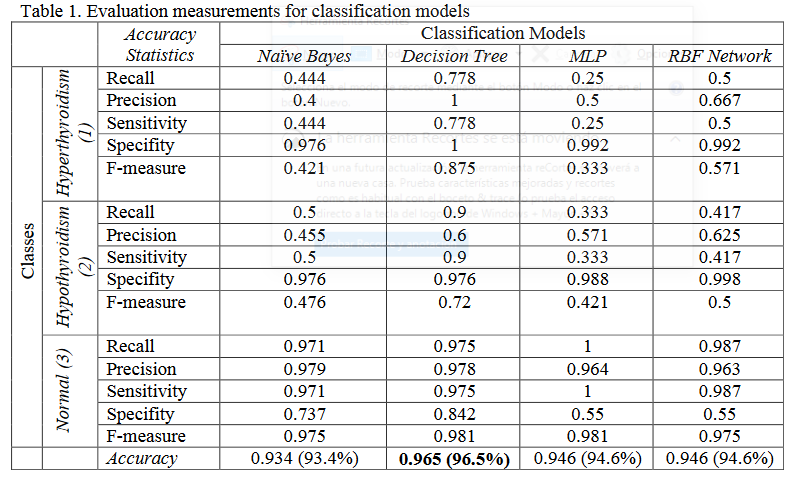


Fig. 1: Resultados artículo 1.

La mayoría de resultados superan el 90% de precisión, siendo el árbol de decisión el más alto de todos, también se puede correr de nuevo el árbol con un método diferente de podado para poder aumentar un poco más la precisión, aun así después de obtener estos resultados, se puede cambiar el porcentaje de participación a 70% para observar cómo cambia la decisión.

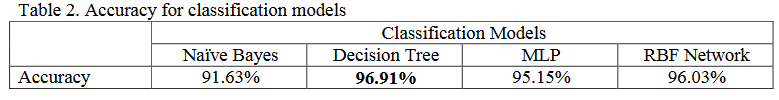


Fig. 2: Diferente técnica de podado.

Como se observa en la figura 2 el árbol de decisión aumenta su precisión demostrando que este es el mejor modelo para poder obtener el mejor calificador de las 3 clases, 1- hipertiroidismo, 2- hipotiroidismo y 3 - normal.

Como conclusión el autor expresa su aceptación por el mejor modelo árbol de decisión y proyecta su trabajo futuro a identificar no solo las características si no los factores que causan la enfermedad de tiroides.

*B. An Empirical Model for Thyroid Disease Classification using Evolutionary Multivariate Bayseian Prediction Method*

El artículo [5] tiene como objetivo usar los datos en bruto para clasificar y dar un preciso diagnóstico. Este aborda la clasificación Bayesiana para lograr una reducción de dimensión en los 7200 datos que se obtiene del repositorio UCI [1].

Para poder validar los datos se utiliza Cross-validation y la matriz de confusión, también aplica el modelo Wrapper, el cual es un algoritmo que predetermina el rendimiento para la selección de características en modelos de clasificación.

La siguiente tabla nos enseña el resultado obtenido entre hombre y mujeres:

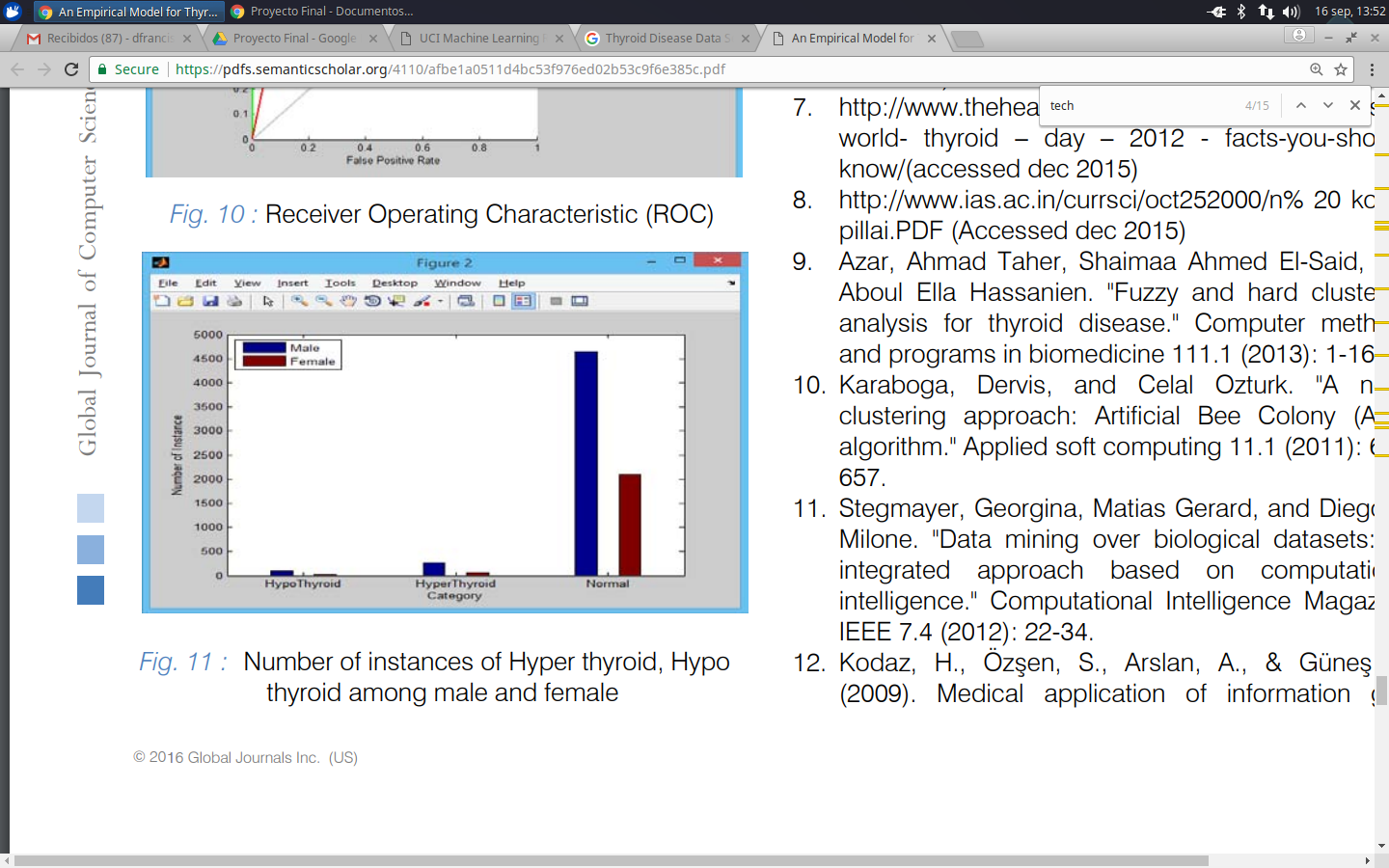


Fig. 3: Número de instancias en las tres categorías.

La decisión de clasificación supera un 97% de precisión haciendo que las métricas que se tomaron para ver si los datos están clasificados correctamente como se observa en la anterior figura.

*C. Using random forest for reliable classification and cost-sensitive learning for medical diagnosis*

El artículo [6] utiliza Random Forest Classifier para minimizar el riesgo de una clasificación errónea especificando diferentes niveles de confianza para las diferentes clases y compara con otros métodos para obtener una mejor precisión.

La técnica de validación es cross-validation y usa diferentes niveles de aceptación (Percentage of certain predictions. Percentage of uncertain predictions. Percentage of empty predictions. Percentage of corrective predictions).

En la siguiente tabla se observan los resultados corridos en diferentes modelos:

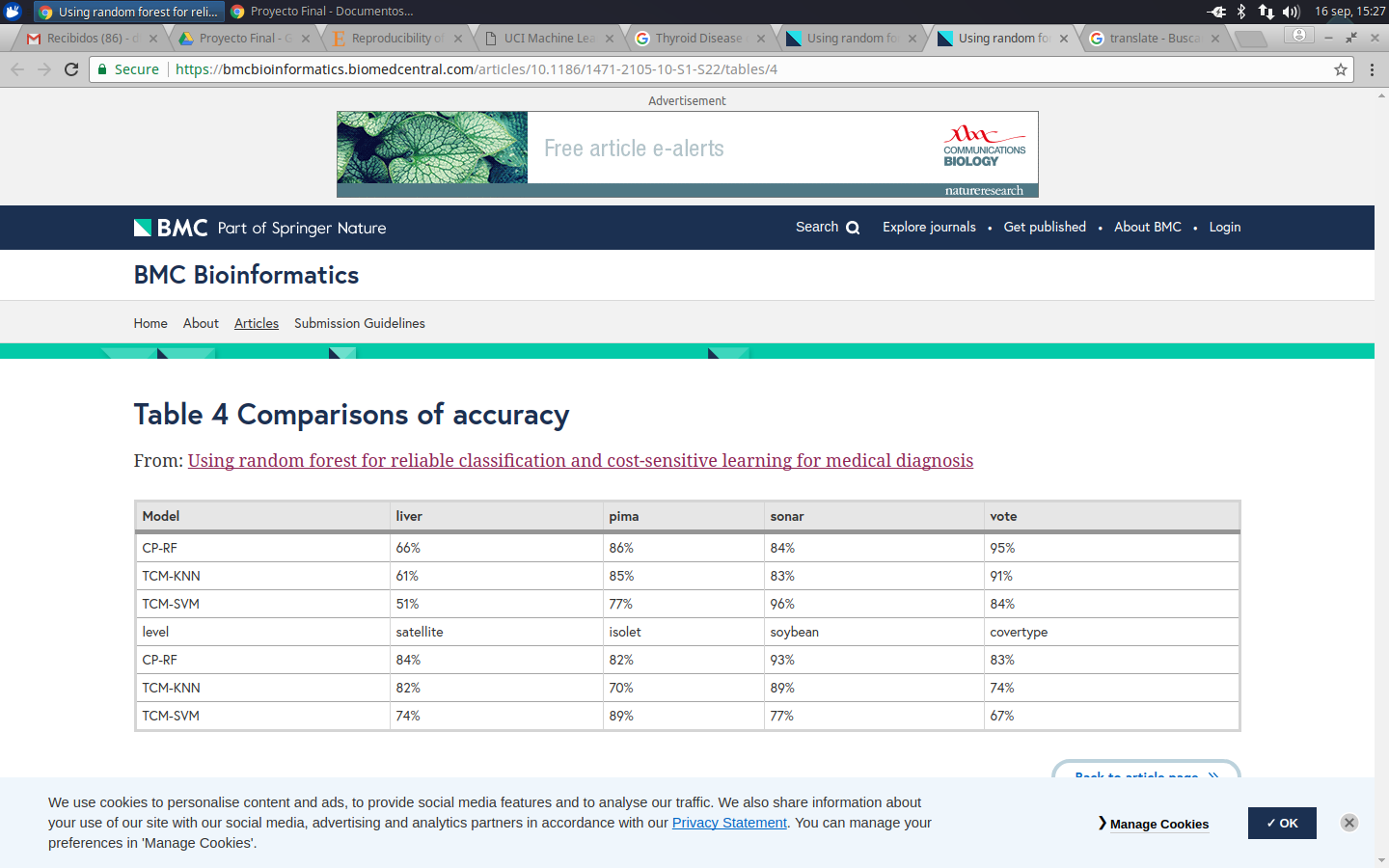


Fig. 4: Porcentajes de aceptación.

Se nota que el RF está bien modelado en la mayoría de diferentes conjuntos, sobre todo en los conjuntos de datos de variables categóricas, por tanto para la clasificación de datos de alta dimensión y con muchos datos ruidosos, este es perfecto como modelo de conjuntos de datos categóricos. En conclusión la precisión del conjunto de datos modelados en RF para el diagnóstico médico de enfermedades es muy probable que sea acertado.

*D. Performance Improvement with Combining Multiple Approaches to Diagnosis of Thyroid Cancer*

En este artículo [7] abordan la convención individual de clasificación para tener una buena comparación el los resultados de rendimiento de los diferentes modelos para poder diagnosticar de manera precisa el cáncer de tiroides.

Este aborda diferentes tipos de modelos que son BayesNet, Naïve Bayes, Sequential Minimal Optimization (SMO), IBK, Random Forest, AdaBoostM1. Los cuales usan la técnica de cross-validation para la validación de los datos los cuales son 21 muestras con 7200 sujetos tomados de la UCI [1].

Después de aplicar los diferentes algoritmos para la clasificación se obtiene los diferentes resultados y también se grafica la curva ROC para diagnosticar los efectos:

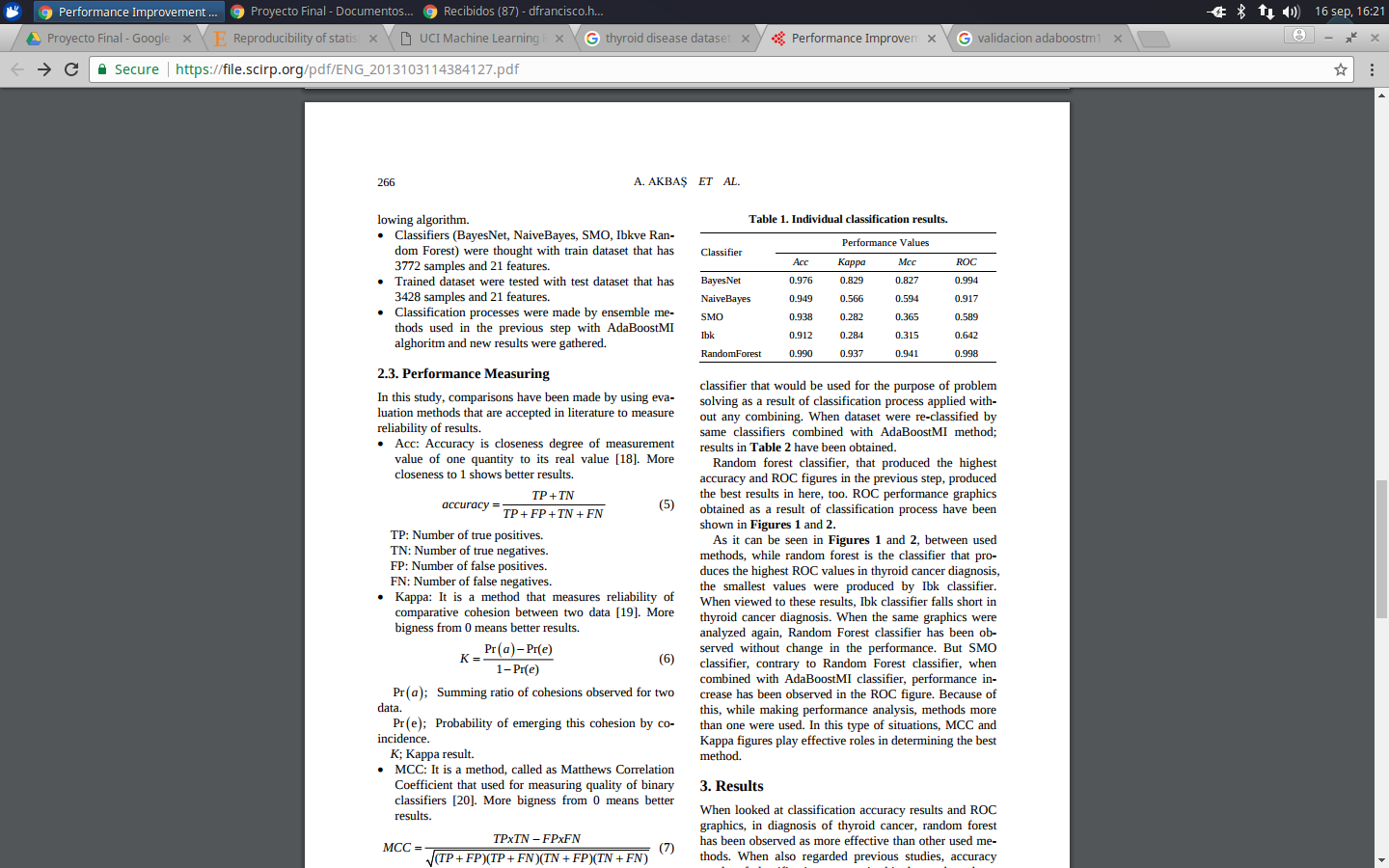


Fig. 5: Clasificación individual.

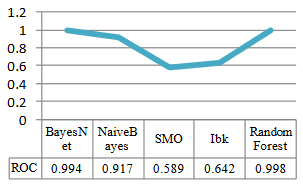


Fig. 6: Gráfica de clasificación individual.

Cuando se observa los resultado de clasificación, notamos que el RF es el que mejor diagnóstico de cáncer de tiroides, por tanto como la alta precisión de este puede afectar a la toma de decisión ya que la base de datos de la UCI [1], es antigua y se tendría que actualizar para poder modelar de nuevo estos datos y comprobar que para diagnósticos futuros el RF es el mejor modelo que se puede usar, obviamente no se descarta la posibilidad de usar otros modelados.

1. **EXPERIMENTOS**

Para este problema se aborda la base de datos “ann-train.data” del respositorio “thyroid disease data set” [1], la cual tiene 21 atributos, 3772 muestras y tiene 3 diferentes clases las cuales la clase 1 tiene 2,47% de las muestras, la clase 2 tiene 5,06% de las muestras y la clase 3 tiene 92,47% de las muestras.

Como se puede observar se tiene un problema de desbalanceo y para ello vamos a utilizar SMOTEENN [8] la cual es una librería de combinación de sobre muestreo y submuestreo inteligente para balancear los datos, creando muestras de la clase minoritaria para obtener una muestra promedio y a continuación utilizaremos la validación estratificada “Naive Bayes” utilizando diferentes porcentajes de sobremuestreo para lograr un mejor resultado.

Como se menciona en los artículos relacionados, el autor elige cuantos folds utiliza para la validación, para este vamos a utilizar 5-folds en la validación estratificada [9].

A continuación se presentan los resultados de los modelos a ejecutar.

*IV-A. Naive Bayes*

TABLE II

NB CON MUESTREO

|  |  |  |
| --- | --- | --- |
| Porcentaje de Balanceo | Error de Clasificación | Intervalo de Confianza |
| 0% | 0.2293085494665928 | 0.03077768184548598 |
| 5% | 0.18107616906144647 | 0.0214205093507078 |
| 10% | 0.1802797067101439 | 0.01583641840034585 |
| 15% | 0.18132703147953405 | 0.02910920455595323 |
| 20% | 0.17868468520535746 | 0.02253861285390327 |
| 25% | 0.18611351712879382 | 0.021841190335585424 |

Como se observa en la tabla solo hasta el 20% se obtiene un intervalo de confianza y un error de clasificación aceptables para el desbalanceo de datos sobre la clase minoritaria.

*IV-B. K vecinos más cercanos*

TABLE III

KNN CON MUESTREO

|  |  |  |
| --- | --- | --- |
| **K-Vecinos** | **Error de Clasificación** | **Intervalo de Confianza** |
| **1** | 0.06495139313798451 | 0.0021859179513763066 |
| **2** | 0.06426142677016276 | 0.0015713932872648002 |
| **3** | 0.06415704154893095 | 0.00230878419526919 |
| **4** | 0.06362653663701129 | 0.0025051004153281 |
| **5** | 0.06495139313798451 | 0.0024862993599678415 |
| **6** | 0.06495244712310225 | 0.002218473473525061 |
| **7** | 0.06468789685855199 | 0.001968414692155269 |
| **8** | 0.06389284447409563 | 0.0030858383661151747 |
| **9** | 0.06548295482703545 | 0.003419225956252193 |
| **10** | 0.06548154765220278 | 0.0031790183116519374 |
| **11** | 0.06601240575383738 | 0.003866622372120999 |
| **12** | 0.06601275615153876 | 0.003064305912252658 |
| **13** | 0.06574715190187078 | 0.003076667288654344 |
| **14** | 0.06548154765220278 | 0.0029487700956473987 |
| **15** | 0.06627730641608903 | 0.0034402849297134695 |
| **16** | 0.06601240575383738 | 0.0035840755771531442 |
| **17** | 0.06601240575383738 | 0.0035840755771531442 |
| **18** | 0.06601240575383738 | 0.003866622372120999 |
| **19** | 0.06601240575383738 | 0.003866622372120999 |

Como se observa en la tabla anterior después de correr K vecinos mayores que 11 empiezan arrojar resultados iguales en sus vecinos, con solo 10 ya se obtienen los valores más exactos de los vecinos a evaluar.

*IV-C. Redes Neuronales Artificiales*

TABLE IV

RNA CON MUESTREO

|  |  |  |  |
| --- | --- | --- | --- |
| **Épocas** | **Capas Ocultas** | **Error Clasificación** | **Intervalo de confianza** |
| 10 | **10** | 0.023672163590481618 | 0.0008595557645385432 |
| **10** | **20** | 0.02385878935407092 | 0.0054453296321465275 |
| **10** | **30** | 0.02755875548229312 | 0.013384044016925624 |
| **10** | **40** | 0.0272060217962261 | 0.012699689840352817 |
| **10** | **50** | 0.02604573560113234 | 0.0002915442473657968 |
| **100** | **10** | 0.02509675560113983 | 0.0002915442473657968 |
| **100** | **20** | 0.020854589746879308 | 0.001277903426419425 |
| **100** | **30** | 0.031094392533932386 | 0.009917802060862492 |
| **100** | **40** | 0.02253464016319878 | 0.0010100516358206506 |
| **100** | **50** | 0.022622706785481735 | 0.0015866662391695295 |
| **1000** | **10** | 0.023152977168262607 | 0.0010267350236736876 |
| **1000** | **20** | 0.021562633216855177 | 0.0012341839056585065 |
| **1000** | **30** | 0.023330281197180048 | 0.0013589674147973035 |
| **1000** | **40** | 0.26523864573010153 | 0.11983287422617692 |
| **1000** | **50** | 0.31450977104697764 | 0.0013364512658075422 |

A continuación se observan las redes neuronales que se corrieron hasta mil épocas variando las capas ocultas hasta 50, no se corren más capas ni épocas mayores ya que los otros resultados no arrojan resultados eficientes.

*IV-D. Random Forest*

TABLE V

RF CON MUESTREO

|  |  |  |
| --- | --- | --- |
| **Árboles** | **Error Clasificación** | **Intervalo de confianza** |
| 10 | 0.051430179628380215 | 0.005513568623128987 |
| 20 | 0.05461708581072187 | 0.003935999698639382 |
| 30 | 0.04904642685783025 | 0.005105214227777381 |
| 40 | 0.053557480369701684 | 0.00414069982469638 |
| 50 | 0.05010884664851578 | 0.003331951975840308 |
| 60 | 0.05037128615081703 | 0.002779226511817212 |
| 70 | 0.04825313483792152 | 0.005164826465345447 |
| 80 | 0.05143229318264277 | 0.0010229341290607123 |
| 90 | 0.05169930739915701 | 0.0039683050154970135 |
| 100 | 0.05249400938591198 | 0.0018532237894483814 |

La cantidad de árboles se toman en cuenta para poder variar los resultados, pero como se observa en la tabla no varían mucho con diferentes tamaños de árbol.

*IV-E. Máquinas de Soporte Vectorial con Kernel Lineal*

TABLE VI

SVMKL CON MUESTREO

|  |  |  |  |
| --- | --- | --- | --- |
| **C** | **Error Clasificación** | **Intervalo de confianza** | **% Vectores de Soporte** |
| **0.001** | 0.929038 | 0.0110914 | 0.142923 |
| **0.01** | 0.935657 | 0.009526 | 0.132536 |
| **0.1** | 0.94742 | 0.0116786 | 0.121417 |
| **1** | 0.9958448 | 0.011023 | 0.107906 |
| **10** | 0.960654 | 0.0111415 | 0.100278 |
| **100** | 0.95992 | 0.00977058 | 0.100094 |

Como se observa en la figura se obtiene un buen resultado con un “C” mayor que muestra cómo se clasifican, enseñando mejor resultado en una buena eficiencia.

*IV-F. Máquinas de Soporte Vectorial con Kernel RBF*

TABLE VII

SVMKRBF CON MUESTREO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C** | **GAMMA** | **Error Clasificación** | **Intervalo de confianza** | **% Vectores de Soporte** |
| **0.001** | **0.01** | **0.926833** | **0.0136474** | **0.141728** |
| **0.001** | **0.1** | **0.926833** | **0.0136474** | **0.144303** |
| **0.001** | **1** | **0.926833** | **0.0136474** | **0.153493** |
| **0.01** | **0.01** | **0.926833** | **0.0136474** | **0.150093** |
| **0.01** | **0.1** | **0.926833** | **0.0136474** | **0.211579** |
| **0.01** | **1** | **0.926833** | **0.0136474** | **0.35872** |
| **0.1** | **0.01** | **0.92904** | **0.0116774** | **0.161121** |
| **0.1** | **0.1** | **0.927567** | **0.0133098** | **0.241359** |
| **0.1** | **1** | **0.926833** | **0.0136474** | **0.467446** |
| **1** | **0.01** | **0.938599** | **0.0101901** | **0.149815** |
| **1** | **0.1** | **0.947412** | **0.0124418** | **0.237865** |
| **1** | **1** | **0.934552** | **0.0115126** | **0.478658** |
| **10** | **0.01** | **0.953665** | **0.00973111** | **0.135018** |
| **10** | **0.1** | **0.949622** | **0.00924815** | **0.214703** |
| **10** | **1** | **0.934549** | **0.0146685** | **0.467536** |
| **100** | **0.01** | **0.958078** | **0.0077083** | **0.126103** |
| **100** | **0.1** | **0.943734** | **0.0112016** | **0.164429** |
| **100** | **1** | **0.926833** | **0.0137868** | **0.457611** |

La tabla anterior muestra como el gamma influencia en la precisión de los datos, menor el gamma mayor influencia tienen los vectores de soporte, como se observa en el gamma 0.01 con C 100.

1. **ANÁLISIS INDIVIDUAL**

*V-A. Matriz de correlación*

Antes de hacer la selección de características se hace el análisis de correlación para saber cuales son las características candidatas a ser seleccionadas y posteriormente eliminar las que no aportan datos.

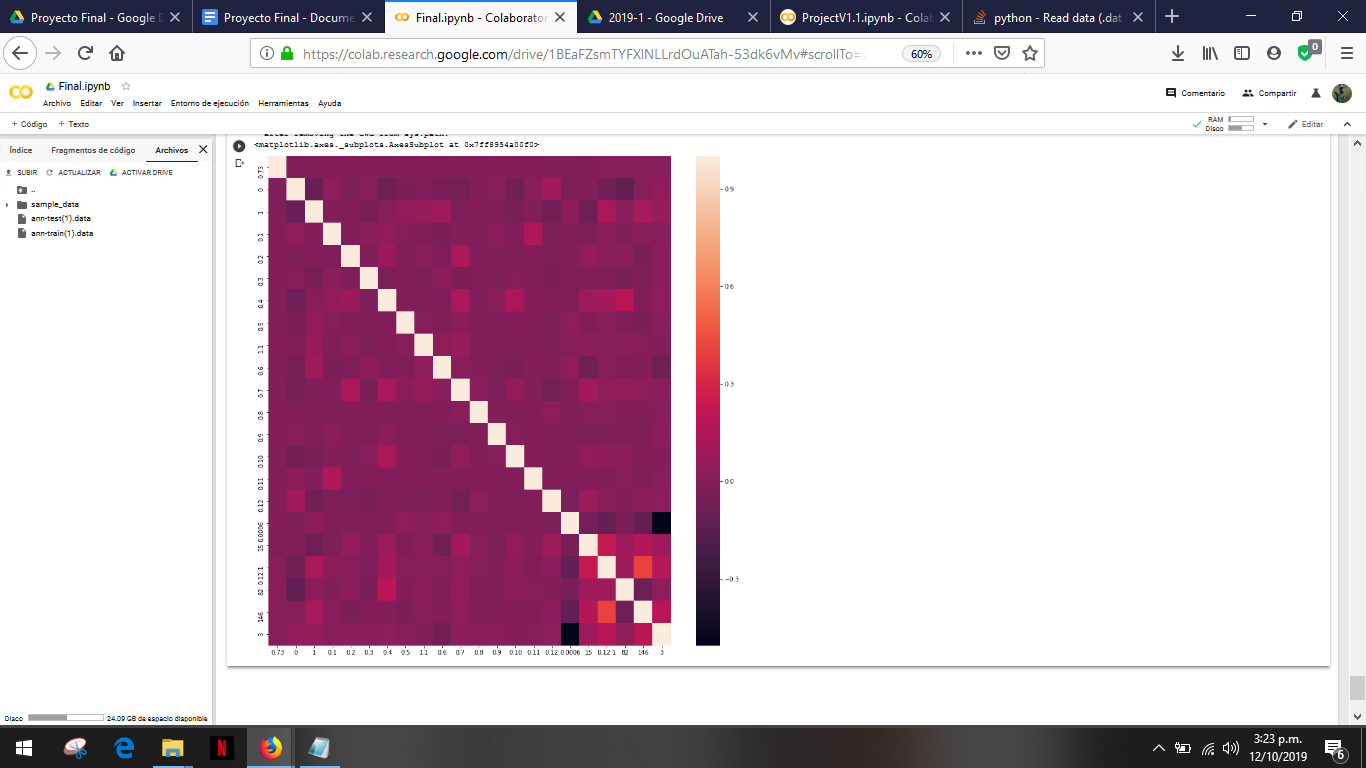
****

Fig. 7: Matriz Correlación

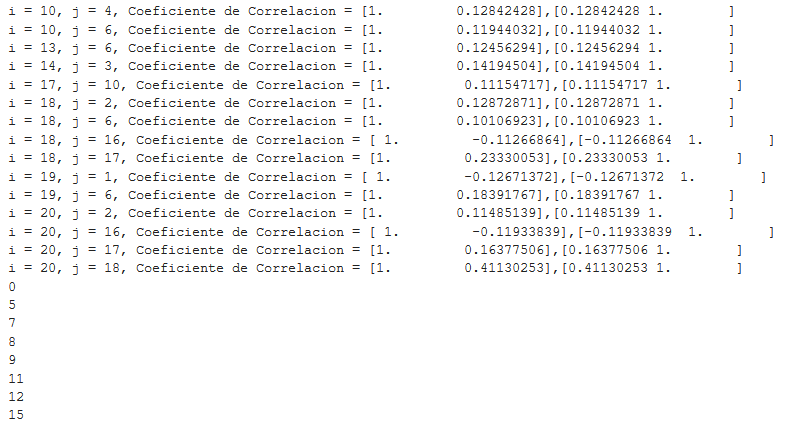
****

Fig. 8: Resultados Candidatos

Luego de analizar la matriz de correlación las características candidatas a ser eliminadas son las características a eliminar son la 0 age, 5 sick, 7 thyroid\_surgery, 8 I131\_treatment, 9 query\_hypothyroid, 11 lithium, 12 goitre, 15 psych.

1. **SELECCIÓN DE CARACTERÍSTICAS**

Para la selección de características se usa Wrapper elige la SFS (Sequential Forward Selection) la cual da como resultado que la precisión mejora con 14 características las cuales son:

(0, 2, 3, 4, 5, 6, 8, 9, 10, 16, 17, 18, 19, 20)

Ahora con la selección de estas características se procederá a evaluar nuevamente los 3 mejores modelos anteriores que son (KNN, RNA, RF):

Ahora analizamos los cambios con las características seleccionadas:

*VI-A. KNN*

TABLE VIII

KNN DESPUÉS DE SFS

|  |  |  |
| --- | --- | --- |
| **K-Vecinos** | **Error de Clasificación** | **Intervalo de Confianza** |
| **1** | 0.06893651086812216 | 0.00973422215667973 |
| **2** | 0.07264828767902566 | 0.007554271920269801 |
| **3** | 0.0697308652491893 | 0.008257849884326455 |
| **4** | 0.07026382294501868 | 0.0072051523455491295 |
| **5** | 0.06548892834004044 | 0.006597693089658821 |
| **6** | 0.06840600316418896 | 0.005032849535723948 |
| **7** | 0.07079257586843138 | 0.007467319614179129 |
| **8** | 0.06840108363630171 | 0.005695105191175573 |
| **9** | 0.0697301588697594 | 0.005086131732871021 |
| **10** | 0.06839686769583078 | 0.004680305950073526 |

*VI-B. RNA*

TABLE IX

RNA DESPUÉS DE SFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Épocas** | **Capas Ocultas** | **Error Clasificación** | **Intervalo de confianza** |
| **10** | **10** | 0.08505260549076457 | 0.11994645281886503 |
| **10** | **20** | 0.08505260549076457 | 0.11994645281886503 |
| **10** | **30** | 0.08505260549076457 | 0.11994645281886503 |
| **10** | **40** | 0.08505260549076457 | 0.11994645281886503 |
| **10** | **50** | 0.08716900760716668 | 0.11896765721543111 |
| **100** | **10** | 0.08505260549076457 | 0.11994645281886503 |
| **100** | **20** | 0.08505260549076457 | 0.11994645281886503 |
| **100** | **30** | 0.08187625032765443 | 0.12158090281273713 |
| **100** | **40** | 0.08505260549076457 | 0.11994645281886503 |
| **100** | **50** | 0.08496407074087522 | 0.11999105639446561 |
| **1000** | **10** | 0.08514114024065389 | 0.1199020941443981 |
| **1000** | **20** | 0.08505260549076457 | 0.11994645281886503 |
| **1000** | **30** | 0.08434432749164991 | 0.1203101155868251 |
| **1000** | **40** | 0.08505260549076457 | 0.11994645281886503 |
| **1000** | **50** | 0.08443286224153924 | 0.12026380540784115 |

*VI-C. RF*

TABLE X

RF DESPUÉS DE SFS

|  |  |  |
| --- | --- | --- |
| **Árboles** | **Error Clasificación** | **Intervalo de confianza** |
| **10** | 0.05222664477169635 | 0.0027267117836939493 |
| **20** | 0.05196771483043613 | 0.004900519271601527 |
| **30** | 0.05435183195875526 | 0.004409409238109414 |
| **40** | 0.049580096517116645 | 0.005043565522232653 |
| **50** | 0.048254533636713524 | 0.0031565094750441365 |
| **60** | 0.04931238150519964 | 0.002862330299469093 |
| **70** | 0.04904466928529618 | 0.0028864883135933514 |
| **80** | 0.05063583362335374 | 0.0012794830046381847 |
| **90** | 0.04984218562171651 | 0.0024814035972441956 |
| **100** | 0.05222488999117584 | 0.003488896817481885 |

Al revisar los tiempos de ejecución solo se notó gran mejora en el **RNA**, por tanto no se muestran los tiempos de ejecución ya que no dan datos diferentes a los ya aquí mencionados.

Se puede concluir que el mejor modelo con las características seleccionadas es el **RF**.

1. **EXTRACCIÓN DE CARACTERÍSTICAS**

Otra forma de mejorar la exactitud y el tiempo de procesamiento es la extracción de características, en este caso usamos Análisis de Componentes Principales (PCA) [9] para así disminuir la dimensión y aumentar el umbral de precisión para luego entrenar los e modelos con mejor aceptación.

A continuación procedemos con la extracción:

TABLE XI

EXTRACCIÓN

|  |  |  |  |
| --- | --- | --- | --- |
| **# Características** | **Error Clasificación** | **Intervalo de confianza** | **Eficiencia** |
| **2** | 0.0755631 | 0.0143574 | 92.47018110913977 |
| **3** | 0.0755631 | 0.0143574 | 92.47018110913977 |
| **4** | 0.0755631 | 0.0143574 | 92.47018110913977 |
| **5** | 0.0654881 | 0.011329 | 92.47018110913977 |
| **6** | 0.0654881 | 0.011329 | 92.47018110913977 |
| **7** | 0.0654881 | 0.011329 | 92.47018110913977 |
| **8** | 0.0654884 | 0.0116363 | 92.47018110913977 |
| **9** | 0.0657537 | 0.0115763 | 92.47018110913977 |
| **10** | 0.0657537 | 0.0116377 | 92.47018110913977 |
| **11** | 0.063898 | 0.0121049 | 92.47018110913977 |
| **12** | 0.0636327 | 0.0115882 | 92.47018110913977 |
| **13** | 0.0633675 | 0.0116903 | 92.60280710383473 |
| **14** | 0.062837 | 0.011172 | 92.6292971700599 |
| **15** | 0.062837 | 0.011172 | 92.97394909269767 |
| **16** | 0.062837 | 0.011172 | 93.00043915892283 |
| **17** | 0.0631022 | 0.0114218 | 93.026929225148 |
| **18** | 0.0628373 | 0.0115756 | 93.07997962302598 |
| **19** | 0.0625724 | 0.0117514 | 93.61016740738138 |
| **20** | 0.0625724 | 0.0117514 | 93.68974300419836 |

En este caso a pesar de que se utilizó la extracción, podemos observar que la eficiencia no varía mucho en comparación de las otras, posiblemente si se balancean los datos igualando todo con un sobremuestreo puede que se obtenga una mejor precisión, pero como se expuso anteriormente no se observa en la tabla no se disminuye bastante la extracción de características, es decir no variaría mucho en el resultado esperado de los modelos a evaluar.

1. **DISCUSIÓN**

En general, los resultados no fueron exactamente lo que se esperaban en comparación con los artículos, primero que todo, los K-folds que se usan en las referencias son 10, pero probando los modelos, no variaba en gran parte los datos con 5 o 10, por lo tanto se optó por dejar 5 K-folds en la validación. El resultado más acertado fue el Random Forest el cual con una técnica de balanceo SMOTENN [9] dando pesos a las clases se logro un error de clasificación aceptable siendo este uno de los tres modelos seleccionados por ser el mejor, aun así este modelo no se acercaba exactamente a los resultados en investigaciones anteriores.

El resultado más acertado fue el Random Forest el cual con una técnica de balanceo SMOTENN [9] dando pesos a las clases se logro un error de clasificación aceptable siendo este uno de los tres modelos seleccionados por ser el mejor, aun así este modelo no se acercaba exactamente a los resultados en investigaciones anteriores.

También la extracción de características no variaba mucho con respecto a las diferentes números de características a extraer, por lo tanto esto no ameritaba una corrida de datos ya que eficiencia y el error, no variaba demasiado. Además de esto con respecto a la selección de características, el modelo KNN no tenía cambios significativos después de la selección, pero el RNA si cambió sus datos en la dirección inesperada.

En conclusión se deja en duda si la forma en que se balancearon los datos es la correcta o si las características seleccionadas si eran las indicadas, para poder correr todos los modelos aunque estos al final muestran parte de un resultado acertado pero con los valores diferentes, pero dejando claro que el RF [7] es el mejor modelo de clasificación para el diagnóstico de la enfermedad de tiroides.

REFERENCES

[1] “UCI MAchine Learning Repository: Thyroid Disease Data Set”, Archive.ics.uci.edu, 1987. [Online]. Avaliable: <https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease>. [Accessed: 20-Ago-2019].

[2] “6 Common Thyroid Disorders & Problems”, Daniel Murrel. MD., 2017, [Online]. Avaliable: <https://www.healthline.com/health/common-thyroid-disorders>. [Accessed: 20-Ago-2019].

[3] “Datasets used for classification:

comparison of results”, is.umk.pl, 1993, [Online]. Avaliable: <http://www.is.umk.pl/~duch/projects/projects/datasets.html>. [Accessed: 20-Ago-2019].

[4] “Prediction of Thyroid Disease Using Data Mining Techniques”, researchgate.net, 2016. [Online]. Avaliable: <https://www.researchgate.net/publication/321145710_Prediction_of_Thyroid_Disease_Using_Data_Mining_Techniques>. [Accessed: 23-Ago-2019].

[5] “An Empirical Model for Thyroid Disease Classification using Evolutionary Multivariate Bayseian Prediction Method”, computerresearch.org, 2016. [Online]. Avaliable: <https://computerresearch.org/index.php/computer/article/view/1353>. [Accessed: 23-Ago-2019].

[6] “Using random forest for reliable classification and cost-sensitive learning for medical diagnosis”, bmcbioinformatics.biomedcentral.com, 2009, [Online]. Avaliable: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-S1-S22>. [Accessed: 29-Ago-2019]

[7] “Performance Improvement with Combining Multiple Approaches to Diagnosis of Thyroid Cancer”, scirp.org, 2013, [Online]. Avaliable: <http://www.scirp.org/journal/paperinformation.aspx?paperid=38870>. [Accessed: 4-Sep-2019].

[8] “imblearn.combine.SMOTEENN”, imbalanced-learn.readthedocs.io, 2016. [Online]. Avaliable: <https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.combine.SMOTEENN.html>. [Accessed: 10-Sep-2019].

[9] “scikit-learn Machine Learning in Python”, scikit-learn.org, 2007, [Online]. Avaliable: <https://scikit-learn.org/stable/documentation.html>. [Accessed: 10-Sep-2019].