Thyroid Disease

Juan David Arias
Departamento de Ingeniería de Sistemas, Facultad de Ingeniería, Universidad de Antioquia
https://github.com/juanArias8/ML-Thyroid-Issues.git
Medellín, Colombia
juan.arias8@udea.edu.co

***Resumen*— Las enfermedades tiroideas son cada vez más comunes a nivel mundial, afectando las condiciones salubres en múltiples países[1]. Este desorden se caracteriza por afectar múltiples de las funcionalidades del cuerpo humano, perjudicando directamente la condición de vida de las personas que la padecen. A nivel mundial el uso de la tecnología en el campo de la medicina ha permitido hacer seguimiento al comportamiento de diferentes desórdenes o enfermedades, y en algunos casos, ha posibilitado incluso predecir cuándo un persona podría o no padecer de alguna de ellas. Con el auge de la inteligencia artificial diferentes académicos y profesionales han enfocado sus esfuerzos en hacer de esta, una herramienta para ayudar a mejorar la condición física de la población mundial en aras de mejorar las condiciones salubres de las personas, y permitirles así una mayor calidad de vida.**

Palabras clave: Tiroides, Eutiroideo, Hormonas tiroideas, Inteligencia Artificial, Machine Learning, Clasificación, Análisis discriminante.

***Abstract*— Thyroid diseases are increasingly common worldwide, affecting health conditions in multiple countries[1]. This disorder is characterized by affecting multiple functionalities of the human body, directly damaging the living condition of people who suffer it. Globally, the use of technology in the field of medicine has made it possible to track the behavior of different disorders or diseases, and in some cases, has even made it possible to predict when a person might or might not suffer from any of them. With the rise of artificial intelligence, different academics and professionals have focused their efforts on making it a tool to help improve the physical condition of the world's population in order to improve the health conditions of people, and thus allow them a better quality of life.**

keywords: Thyroid, Euthyroid, Thyroid hormones, Artificial Intelligence, Machine Learning, Classification, Discriminant Analysis.

I. INTRODUCCIÓN

Se calcula que en Estados Unidos más del 12 por ciento de sus habitantes han desarrollado de cierta forma alguna condición hormonal de la glándula Tiroidea, además, se estima que aproximadamente 20 millones de Americanos tienen alguna forma de enfermedad Tiroidea [2]. En el caso de la India se estima que aproximadamente existen más de 42 millones de personas que padecen una situación similar [3]. Para el caso nacional, se calcula que aproximadamente el 4% de la población sufre de algún desorden hormonal [4], y se calcula que aproximadamente a nivel mundial unos 200 millones de personas padezcan algún trastorno tiroideo.

> *"La glándula tiroide es un órgano que todos tenemos en la parte inferior de la región anterior del cuello, que cumple funciones muy importantes en la regularización del metabolismo y en el correcto funcionamiento de casi todos los órganos del cuerpo. Esta glándula produce las hormonas tiroideas, la tiroxina (T4) y la triyodotironina (T3), en respuesta a la acción de la hormona estimulante del Tiroides (TSH), la cual es secretada por la glándula hipófisis que se encuentra en el cerebro"[5].*

Es importante decir que a nivel mundial las patologías de este tipo afectan mayoritariamente a mujeres, especialmente si se encuentran en edades avanzadas, según la información obtenida de THYROID AWARE las mujeres tienen entre cuatro y siete veces más posibilidad de ser afectadas por un trastorno de la tiroides que los hombres [6], según la federación internacional de Tiroides los efectos que puede causar dicho desorden en las personas van desde el cansancio excesivo, la depresión, ansiedad, irritabilidad, dificultad de concentración, dolores de cabeza y migraña, aletargamiento, intolerancia a lugares fríos o calientes, alteración de la menstruación e incluso alteración en el sistema autoinmune.

> *"Los síntomas de la enfermedad tiroidea varían según el tipo. Hay cuatro tipos generales: 1) hipotiroidismo (baja función) causado por no tener suficientes hormonas tiroideas; 2) hipertiroidismo (función alta) causado por tener demasiadas hormonas tiroideas; 3) anomalías estructurales, más comúnmente un agrandamiento de la glándula tiroides; y 4) tumores que pueden ser benignos o cancerosos. También es posible tener pruebas anormales de la función tiroidea sin ningún síntoma clínico."[7]*

Debido a las condiciones descritas anteriormente se hace necesario plantear mecanismos que puedan de cierta forma ayudar a combatir el problema, bien sea para recomendar tratamientos, descubrir la patología en una persona que desconozca su situación salubre (se estima que sólo el 50% de las personas que padecen de alteraciones en la tiroides es consciente de su situación) [8] o en el mejor de los casos para predecirla. Es por lo anterior que surge una motivación más allá de lo académico al momento de realizar este trabajo de investigación Al igual que lo han hecho gran cantidad de personas se propone hacer uso de herramientas tecnológicas con el fin de investigar sobre la tiroides, en el caso en particular, se pretende hacer uso del machine learning en su paradigma de aprendizaje supervisado con el fin de poder clasificar de manera asertiva y rápida, basados en datos obtenidos de una persona, si esta puede o no padecer de problemas tiroideos.

Para llevar a cabo la propuesta y el desarrollo del proyecto se hace necesario disponer de un conjunto de datos que lleven un histórico acerca de la enfermedad, en nuestro caso, el conjunto de datos a utilizar forma parte de la colección almacenada por la Universidad de California, Irvine (UCI) ubicada en Estados Unidos. dicha colección puede ser consultada en el archivo universitario hospedado en la siguiente URL: https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease, bajo el nombre de "Thyroid Disease Data Set", en la cual se pueden encontrar diferentes conjuntos de datos sobre la tiroides. La base de datos fue donada por el ingeniero informático Ross Quinlan en el año de 1987 durante la visita que realizó para el "1987 Machine Learning Workshop"[9]. En el proyecto a desarrollar se hará uso del dataset que recibe por nombre "sick-euthyroid", la cual se encuentra disponible desde el 26 de febrero de 1990.

II. DATASET

La base de datos está distribuida en dos archivos, sick-euthyroid.names el cual contiene el metadata de la base de datos, las clases de las cuales se dispone, el nombre de los atributos y los posibles datos que esta puede poseer. En la tabla 1 se muestra en detalle la información general del dataset.

atribute	value	value	faltantes
age	continuous		?
sex	M	F	?
on_thyroxine	f	t	
query_on_thyroxine	f	t	
on_antithyroid_medication	f	t	
thyroid_surgery	f	t	
query_hypothyroid	f	t	
query_hyperthyroid	f	t	
pregnant	f	t	
sick	f	t	
tumor	f	t	
lithium	f	t	
goitre	f	t	
TSH_measured	n	y	
TSH	continuous		?
T3_measured	n	y	
T3	continuous		?
TT4_measured	n	y	
TT4	continuous		?
T4U_measured	n	y	
T4U	continuous		?
FTI_measured	n	y	
FTI	continuous		?
TBG_measured	n	y	
TBG	continuous		?

Tabla 1: variables

En total cuenta con 25 características, que se distribuyen como sigue:

- 7 variables continuas
- 18 variables categóricas

Además de lo anterior la clase representa un problema de clasificación biclase, las clases se enumeran a continuación:

- Clase 0: negative - 2870 muestras
- Clase 1: sick-euthyroid - 293 muestras

En la tabla se puede apreciar claramente las variables que poseen datos faltantes, los cuales son representados con el signo de interrogación. En general la base de datos no cuenta con gran cantidad de datos faltantes, a excepción de, la variable TGB, la cual presentó un índice del 91% de "?", por lo cual se decide eliminar la columna correspondiente y esta y a su compañera TGB measure. Al resto de variables que contenían datos faltantes se les aplicó la estrategia de imputación por la media. Es importante resaltar que en las demás variables con pérdida de datos el índice de pérdida era inferior al 20%.

III. HISTORIA DEL ARTE

En el proceso de recolección de información académica para revisar el estado del arte del trabajo, se evidenció que numerosas personas han trabajado con el mismo problema, haciendo uso de diferentes arquitecturas de machine learning, a continuación se listan los 5 artículos académicos más representativos que se encontraron.

1. **Prediction of Thyroid Disease Using Data Mining Techniques[9]**

El trabajo realizado por Irina y Liviu fue desarrollado basado en la herramienta KNIME Analytics Platform 2.12, la cual "es una herramienta de minería de datos que permite el desarrollo de modelos en un entorno virtual" [9.1], dicha herramienta se encuentra escrita en Java y está disponible bajo la licencia GPL. Los resultados obtenidos por el equipo se muestran en la siguiente tabla.

		Accuracy Statistics	Classification Models			
			Naïve Bayes	*Decision Tree*	*MLP*	*RBF Network*
Classes	*Hyperthyroidism (1)*	Recall	0.444	0.778	0.25	0.5
		Precision	0.4	1	0.5	0.667
		Sensitivity	0.444	0.778	0.25	0.5
		Specifity	0.976	1	0.992	0.992
		F-measure	0.421	0.875	0.333	0.571
	Hypothyroidism (2)	Recall	0.5	0.9	0.333	0.417
		Precision	0.455	0.6	0.571	0.625
		Sensitivity	0.5	0.9	0.333	0.417
		Specifity	0.976	0.976	0.988	0.998
		F-measure	0.476	0.72	0.421	0.5
	Normal (3)	Recall	0.971	0.975	1	0.987
		Precision	0.979	0.978	0.964	0.963
		Sensitivity	0.971	0.975	1	0.987
		Specifity	0.737	0.842	0.55	0.55
		F-measure	0.975	0.981	0.981	0.975
		Accuracy	0.934 (93.4%)	**0.965 (96.5%)**	0.946 (94.6%)	0.946 (94.6%)

2. Thyroid Data Prediction using data Classification Prediction[10]

Los hallazgos encontrados por Ammunu y Venegopal se lograron haciendo uso de la herramienta Weka, la cual es una herramienta open source diseñada para realizar actividades de minería de datos, fue desarrollada por la Universidad de Waikato en Nueva Zelanda y está escrita en el lenguaje de programación Java. El trabajo fue realizado haciendo uso del paradigma Random Forest. Los resultados obtenidos por el equipo se muestran a continuación:

Confusion matrix Result for Different K value

K=n	*10*	*8*	*6*	*3*
Accuracy	*70.519 %*	*71.086 %*	*71.160 %*	*71.162 %*
TP rate	*0.705*	*0.711*	*0.712*	*0.712*
FP rate	*0.318*	*0.312*	*0.312*	*0.318*
Precision	*0.698*	*0.701*	*0.701*	*0.705*
Recall	*0.705*	*0.711*	*0.712*	*0.712*
F-Measure	*0.690*	*0.696*	*0.696*	*0.695*

3. Efficient Thyroid Disease Classification Using Differential Evolution Systems.[11]

El trabajo desarrollado por K. Geetha y S. Santosh está centrado en el uso de un algoritmo híbrido denominado Differential Evolution, es una metodología perteneciente a la familia de algoritmos evolutivos, el cual es usado para crear subconjuntos de hijos de los registros padres. El equipo logró una exactitud en la clasificación del 99.89%, métrica que fue tomada haciendo uso del test T, usado para medir la probabilidad de clasificación errónea.

4. Predicting Thyroid Disease Using Linear Discriminant Analysis (LDA) Data Mining Technique. [12]

G. Rasitha Banu realizó su trabajo haciendo uso del Análisis Discriminante Lineal (LDA) en la herramienta Weka, el proceso anterior le propició una exactitud en la tarea de clasificación del 99.62% usando validación cruzada con k=6. Los resultados obtenidos se muestran en la siguiente tabla de exactitud.

K=n	Accuracy	TP-rate	FP-rate	Precision	Recall	ROC area
K=10	99.49	0.995	0.013	0.995	0.995	0.997
K=8	99.57	0.996	0.013	0.995	0.996	0.996
K=n	**Accuracy**	**TP-rate**	**FP-rate**	**Precision**	**Recall**	**ROC area**
K=6	99.62	0.996	0.013	0.996	0.996	0.996
K=4	99.60	0.996	0.006	0.996	0.996	0.996
K=2	99.39	0.994	0.016	0.993	0.994	0.996

IV. METODOLOGÍA

El desarrollo del proyecto se basó en una metodología escalonada, los pasos desarrollados se dieron como sigue:

A. Preprocesamiento

Debido al estado original de la base de datos, se hizo necesario aplicar ciertas técnicas que permitieran adecuarla con el fin de luego ser procesada por los modelos seleccionados, como se mencionó con anterioridad, el primer paso llevado a cabo fue la imputación de características faltantes y la eliminación de dos columnas, una de ellas porque tenía índices de pérdida de datos del 91% y la segunda porque era directamente dependiente de la primera.

El segundo paso a realizar fue cambiar la codificación de las variables categóricas, las cuales venían representadas por las letras ‘F’, ‘M’, ‘y’, ‘n’, ‘t’, ‘f’. Dicho procedimiento fue necesario realizarlo, dado que sin esto los modelos de las librerías utilizadas no funcionaban de manera adecuada.

El tercer paso realizado consistió en hacer over y under sampling sobre las muestras de las dos clases.
Tal como se indicó anteriormente, la base de datos cuenta con dos clases, una de ellas posee 2870 muestras mientras que la otra cuenta con apenas 293, lo cual nos da aproximadamente una relación de 10:1. Lo anterior resulta en un problema sobretodo porque la mayoría de algoritmos de clasificación son sensitivos al desbalance en la predicción de las clases. Un modelo categórico que sea entrenado y testeado con una base de datos desbalanceada podría dar niveles de exactitud muy altos aún cuando esté realizando la clasificación de manera errónea, lo anterior se debe a que el set de datos hará que el BIAS tienda siempre a la clase más común.
Dada la situación anterior, se hizo necesario aplicar técnicas de sobremuestreo para la clase minoritaria y submuestreo para la clase mayoritaria, al finalizar esta etapa, cada clase terminó con un número igual de características, cuyo número oscila entre las 300 y 400 muestras.

El paso final para la etapa de preprocesamiento e basó en la selección de 10 muestras de manera aleatoria, cuyo propósito sería netamente

probar los modelos desarrollados y así evitar tener niveles de exactitud que realmente no representaban la capacidad del modelo al momento de predecir. Las muestras fueron aisladas del set de entrenamiento y validación.

B. Exploración de modelos

Esta etapa es la parte central del proyecto, pues es en esta donde se entrenan, se validan y se prueban las capacidades de los diferentes modelos utilizados de clasificar de manera correcta o errónea. Es importante recalcar que cada uno de los modelos seleccionados se entrenó y validó tanto con la base de datos desbalanceada como la base de datos balanceada, lo anterior se realiza, primero con el objetivo de verificar la veracidad del problema del desbalanceo y finalmente con el fin de comprobar cuáles de los modelos se comportan mejor en casos de desbalance.

Todos los modelos trabajados se hicieron bajo el lenguaje de programación python haciendo uso de las diferentes librerías que este lenguaje ofrece para el análisis de datos y el cálculo científico.

Los modelos que se trabajaron en el desarrollo del proyecto serán descritos a continuación:

- **Naive Bayes**

También conocido como el clasificador ingenuo. Es un clasificador probabilístico fundamentado en la teoría de Bayes y algunas hipótesis simplificadoras adicionales. El modelo trabajado hace parte de la librería sklearn y recibe el nombre de ComplementNB, se decide hacer uso de este dado que en la documentación oficial se indica que ha sido diseñado para corregir las suposiciones severas hechas por el clasificador Multinomial Naive Bayes estándar. El modelo se prueba con 350 muestras de cada clase y haciendo uso de las estrategia de validación estratificada. Los resultados obtenidos se pueden apreciar en la siguiente tabla:

Folds	Train	Test	Sensibility	Specicity
5	75	86	84	52
10	74	7	85	54
15	73	84	91	69
20	73	82	88	76
30	73	86	90	81
50	73	85	85	85

.

- **KNN**

K vecinos más cercanos es un método de clasificación no paramétrico, el cual estima la función de densidad de probabilidad o directamente la probabilidad, hace uso de la medida de distancia entre vectores para clasificar las muestras, lo anterior basado en el supuesto de que las los cambios de la función modelo describe cambios suaves, por tanto la clasificación se puede hacer midiendo la moda de los k vecinos cercanos. Para este modelo se hace uso de la librería sklearn haciendo uso de KNeighborsClassifier, se realizan dos pruebas con dicho modelo, la primera consiste en entrenar y validar haciendo uso de la base de datos desbalanceada y realizando una partición de los datos, tomando 40% para validación y 60% para entrenamiento, los resultados obtenidos bajo esta configuración se muestran a continuación:

k	Train	Test	Sensibility	Specicity
2	0.93	0.89	0.05	0.98
5	0.92	0,89	0.15	0.97
10	0.91	0.90	0.10	0.99
25	0.91	0.90	0.09	0.99
20	0.91	0.90	0.07	0.99

Si observamos los datos contenidos en la tabla, podemos observar que se trata de un modelo entrenado con una base de datos desbalanceada, aún cuando los valores de exactitud en la etapa de entrenamiento y validación son significativos, al hallar la sensibilidad del modelo podemos ver que este está clasificando mal los datos.

El segundo enfoque que se realiza con KNN es a partir de un conjunto de datos que ha sido alterado haciendo sobre y submuestreo y adempas haciendo uso de validación estratificada. Los resultados obtenidos bajo este enfoque se aprecian a continuación:

K	Folds	Train	Test	Sensibi	Specicit
2	2	92	74	66	80
5	5	84	74	.8	68
10	10	82	79	8	8
15	15	80	74	82	56
20	20	78	77	76	47

donde claramente se puede apreciar una similitud y constancia en los datos. Haciendo uso del set de datos de test, se puede observar a simple vista que si bien falla en algunas clasificaciones, la mayoría de ellas las acierta

- **MLP**

El perceptrón multicapa es una red neuronal compuesta por un conjunto de perceptrones simples, lo cual lo potencializa a hacer, cada vez que hayan más perceptrones, capacidad de cálculos más complejos sin la limitante de que los problemas a resolver pueden no ser linealmente separables. esta arquitectura consiste de por lo menos tres capas, la capa de entrada de datos, una capa oculta y finalmente una capa de salida. Al igual que para el caso anterior, el MPL se entrenó tanto con los datos balanceados como los desbalanceados, claramente la diferencia en la capacidad del MPL de clasificar en el segundo caso es mucho mejor. Para nuestro caso se hace uso de la serie de algoritmos definidos por la librería sklearn haciendo principal uso de MLPClassifier.

La tabla de resultados se presenta a continuación:

	Neuron	Train	Test	Sensibil	Specicit
logistic	(20, 20)	95	91	92	89
	(35, 35, 20)	96	91	90	92
tanh	(20, 20)	94	86	91	89
	(35, 35, 20)	96	88	89	82

por lo cual podemos comprobar que ha sido el mejor modelo hasta el momento.

- **Random Forest**

Random Forest es una técnica ampliamente utilizada y aceptada debido a las altas prestaciones que ofrece, esta técnica consiste en la creación de un bosque combinando múltiples árboles de decisión. Esta técnica hace uso de dicho conjunto de árboles, a cada uno de los cuales les asigna una porción de los datos de entrenamiento, para al final combinar los esfuerzos realizados por todo el grupo y tomar una decisión. En el desarrollo de este proyecto se hace uso de RandomForestClassifier, el cual es un clasificador que se encuentra disponible en la librería de python sklearn. La estrategia utilizada con este paradigma de clasificación se divide en tres partes, la primera de ella es utilizar la base de datos desbalanceada, una transformación de los datos haciendo uso del escalador estándar, el cual se encarga de estandarizar el conjunto de características removiendo la media y escalando la varianza a la unidad, además se divide el conjunto de datos dedicando el 70% para entrenamiento y el 30% para validación, se utilizan 20 estimadores en el bosque. Los resultados obtenidos de este proceso son:

RF Estima	Train	Test	Sensibility	Specicity
20	99	99	82	98

Sin embargo, al evaluar el conjunto de pruebas en nuestro modelo, podemos observar que todas las muestras de la clase 1 las clasificó como clase 0, por tanto podríamos decir que el modelo es víctima de a base de datos desbalanceada.

El segundo enfoque que realizamos con el RandomForest consiste en crear un sistema de validación estratificado pero aún con la base de datos desbalanceada, los resultados se presentan a continuación:

Trees	Variable	Train	Test	Sensibil	Specici
5	5	99	97	91	98
10	10	99	97	87	98
15	15	99	97	93	99
20	20	99	97	87	99

Los resultados anteriores posicionan a este modelo como el mejor hasta el momento.

El enfoque final que se realiza consiste en balancear la base de datos y utilizar validación estratificada , de nuevo con los mismos valores de número de árboles y características de la versión anterior, los resultados fueron los siguientes:

Trees	Variable	Train	Test	Sensibil	Specici
5	5	99	95	93	93
10	10	99	96	93	95
15	15	99	96	93	93
20	20	99	99	92	93

Resultados en los cuales podemos observar una disminución en los campos de validación, en la sensibilidad y en la especificidad respecto al modelo anterior.

- **SVM**

Las máquinas de soporte vectorial son modelos que representan a los puntos de muestra en el espacio separando dos clases en 2 espacios lo más amplio que pueda mediante un hiperplano de separación definido[13]. En esta sección se ha realizado el mismo experimento de entrenamiento y validación con la base de datos desbalanceada y la balanceada, es de importancia recalcar que los valores obtenidos en el primero son en su mayoría erróneos. Mientras que con la base de datos balanceada el comportamiento es diferente y realiza muy buenas predicciones. Para trabajar con SVM usamos los algoritmos definidos en la librería sklearn para la versión de clasificación, esto es SVC. A continuación se detallan los valores obtenidos del proceso.

Kernel	Gamm	C	Train	Test	Sensib	Specic
rbf	0.1	1	99	83	68	98
	1	10	1	83	65	78
linear	0.1	0,01	86	86	93	77
	1	1	92	91	93	90

En esta sección podemos observar buenos resultados si utilizamos 380 muestras por cada clase, un kernel lineal, un valor de gamma de 1 y un valor de c de 1.

C. Extracción de características y reevaluación

Con el objetivo de verificar cuáles son las variables más representativas de la base de datos, es decir, cuáles son las variables que aportan más información al sistema, se hace uso de diferentes técnicas tanto de selección como de extracción de características, como lo son FSF, PCA, y LDA.

A continuación se pretende dar una somera explicación del procedimiento llevado a cabo para lograr este propósito, y además se

pretende enseñar cuáles fueron los resultados obtenidos en el proceso de entrenamiento y validación con los nuevos conjuntos de datos.

- **Índice de Fisher**

El análisis discriminante de Fisher busca realizar una reducción de dimensionalidad haciendo uso de direcciones que son eficientes para discriminación, es decir, que permitan una mejor separación de las clases en el espacio de menor dimensión[14]. En el proceso desarrollado con el fin de implementar la definición anterior, lo primero que se hace con nuestro conjunto de datos es aplicar un sobre y submuestreo con el fin de que ambas clases queden balanceadas, dejando el número de atributos para cada clase en un valor de 380 muestras. Seguidamente, se emplea la técnica de análisis discriminante lineal (LDA) la cual busca un nuevo espacio de características para proyectar los datos con el fin de maximizar la separabilidad de las clases. Se hace uso de dos modelos LDA con el fin de comparar resultados, el primero de ellos se define haciendo uso de un solucionador eigen y un recogimiento automático, el segundo de ellos se define por defecto, se hace el respectivo entrenamiento y transformación y los resultados que se obtienen al finalizar es una varianza de 0.22 para la primer definición y de 1 para la segunda. Finalmente realizamos una prueba haciendo uso de RandomForestClassifier y obtenemos los siguientes valores:

Folds	RF esti	Train	Test	Sensibil	Specici
6	20	99	96	90	95

Valores que resultan ser muy buenos comparando los resultados obtenidos en puntos anteriores, pero que siguen estando por debajo del promedio que ha obtenido RandomForest anteriormente.

- **SFS**

SFS es una estrategia de búsqueda, conocida como Selección Secuencial hacia Adelante, se emplea para realizar selección de características y consiste en que el proceso inicia con un conjunto de característica vacío, pero que una a una se van agregando al modelo hasta alcanzar el punto deseado. La característica agregada es aquella con la cual el criterio de selección se maximice. [15]
La versión utilizada en el proyecto hace parte de la librería MLxtend, la cual se encuentra disponible para el lenguaje de programación Python. De nuevo el desarrollo es similar a pasos anteriores, se balancea el conjunto de datos dejando 380 muestras por clase, e define un modelo RandomForest, el cual se encargará de ayudar a el algoritmo SFS en su proceso de definición y entrenamiento y finalmente se recogen las métricas arrojadas por el algoritmo, en nuestro caso este concluye que las características más significativas para el modelo son [0, 4, 7, 11, 12, 15, 19, 20, 22]

Creando una gráfica número de características vs media de Cv obtenemos la figura 1, de nuevo evaluamos los mejores modelos obtenidos en el proceso con la nueva configuración de características, para nuestro caso RandomForest, ANN, SVM. Los resultados se muestra a continuación.

Model	Train	Test	Sensibil	Specici
RandomF	99	96	90	95
ANN	94	90	92	88
SVM	86	86	93	70

De nuevo RandomForest obtiene los mejores porcentajes de clasificación.

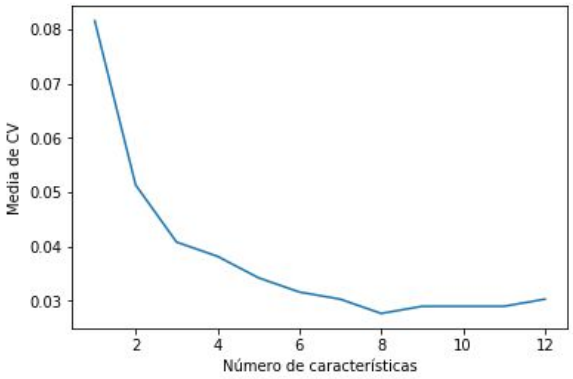

Figura 1. SFS resultado

- **PCA**

El Análisis de Componentes Principales (PCA) es una técnica utilizada para realizar reducción de dimensionalidad apoyados en la teoría del álgebra lineal sobre los vectores propios. El objetivo de esta técnica es disminuir la dimensionalidad del espacio de características pero conservando la mayor cantidad de información posible. Nuestro procedimiento para esta técnica es exactamente igual al llevado en el punto anterior, balanceo, transformación, obtención de nuevo conjunto de variables y pruebas sobre los modelos más eficientes, esto es RandomForest, ANN y SVM. Los resultados obtenidos se muestran a continuación.

Model	Compo	Train	Test	Sensibil	Specici
Random	1	99	96	90	95
ANN	1	94	90	92	88
SVM	1	86	86	93	77

Situación bajo la cual de nuevo RandomForest se reafirma como el mejor clasificador para el proyecto en particular. Es importante recalcar que los resultados anteriores se obtuvieron a partir de una sóla componentes, es decir, la variable que PCA seleccionó como la mejor.

V. DISCUSIÓN

Tal como lo expresa Rocío Erandi et al. en su artículo Árboles de decisión como herramienta en el diagnóstico médico [16], el margen de error que deja este paradigma de clasificación es mínimo en cuanto a asuntos de salud se refiere. Los resultados obtenidos en el desarrollo de del proyecto me dejan como individuo bastante satisfecho, el modelo de RandomForest alcanzó valores del 99.92% de eficiencia en la etapa de entrenamiento, 97.43% en la etapa de

validación, 99.16% de especificidad y 95.23% de sensibilidad, valores que se pueden considerar buenos dado el problema y que además se encuentran muy cerca de los valores obtenidos por los expertos citados en el segundo punto de este artículo.

NOTA: Los scripts desarrollados para llevar a cabo el proyecto pueden ser consultados en el siguiente repositorio de github: https://github.com/juanArias8/ML-Thyroid-Issues.git

VI. .AGRADECIMIENTOS

Agradezco al profesor Julián David Arias, por el acompañamiento que nos ha ofrecido en el margen del semestre 2018-2 a pesar de las dificultades académicas que atraviesa el país, además, por su disposición para compartir sus conocimientos con sus alumnos.

VII. REFERENCIAS

[1]http://www.eluniversal.com.co/salud/el-4-de-los-colombianos-sufren-problemas-de-tiroides-278828-HBEU394560

[1] https://www.thyroid.org/media-main/press-room/

[2] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3 169866/

[4]http://www.eluniversal.com.co/salud/el-4-de-los-colombianos-sufren-problemas-de-tiroides-278828-HBEU394560

[5] http://www.elhospitalblog.com/enfermedades-tiroideas/

[6] Thyroid Foundation of Canada. Thyroid disease: know the facts. Disponible en: ttp://www.thyroid.ca/know_the_facts.php.

[7] http://www.ijirst.org/articles/IJIRSTV4I2054.pdf

[8]https://archive.ics.uci.edu/ml/machine-learning-databases/thyroid-disease/HELLO

[9] Ionita, Irina. (2016). Prediction of Thyroid Disease Using Data Mining Techniques. BRAIN. Broad Research in Artificial Intelligence and Neuroscience. Vol.7. pp.115-124. URL: https://www.researchgate.net/publication/321145710_Prediction_of_Thyroid_Disease_Using_Data_Mining_Techniques

[9.1] https://es.wikipedia.org/wiki/KNIME

[10] Ammulu K., Venugopal. (2017). Thyroid Data Prediction using Data Classification Algorithm. IJIRST –International Journal for Innovative Research in Science & Technology. Vol.4. Issue 2. July 2017. ISSN (online): 2349-6010

URL: http://www.ijirst.org/articles/IJIRSTV4I2054.pdf

[11] Banu, Gulmohamed. (2016). Predicting Thyroid Disease using Linear Discriminant Analysis (LDA) Data Mining Technique. Communications on Applied Electronics. 4. 4-6. 10.5120/cae2016651990. URL: https://www.caeaccess.org/research/volume4/number1/banu-2016-cae-651990.pdf

[12] Geetha K., Santosh S. Eficient Thyroid Disease Classification Using Differential Evolution with SVM. Journal of Theoretical and Applied Information Technology. Vol.88. No.3. E-ISSN: 1817-3195 URL: http://www.jatit.org/volumes/Vol88No3/4Vol88No3.pdf

[13]https://es.wikipedia.org/wiki/M%C3%A1quinas_de_vectores_de_soporte

[14]https://github.com/jdariasl/ML_IntroductoryCourse/blob/master/Clase%20-%20An%C3%A1lisis%20Discriminante%20de%20Fisher.ipynb

[15]https://github.com/jdariasl/ML_IntroductoryCourse/blob/master/Clase%20-%20Selecci%C3%B3n%20de%20Caracter%C3%ADsticas.ipynb