

Proyecto de Aprendizaje de Máquina: Progresión de la Enfermedad de Parkinson

Machine Learning Project: Progression of Parkinson Disease

Kevin Martinez Gallego, Andrés Mauricio Álvarez Ortiz

Departamento de Ingeniería de Sistemas

Universidad de Antioquia

Medellín, Colombia

Junio de 2020

kevin.martinez@udea.edu.co, amauricio.alvarez@udea.edu.co

Resumen: A través de la escala UPDRS (Unified Parkinson's Disease Rating Scale) se puede hacer seguimiento a la progresión de la enfermedad de Parkinson en los pacientes; el valor *motor*-UPDRS oscila entre 0-108 y se asocia a tareas como el habla. Este artículo presenta los resultados obtenidos en la predicción del valor *motor*-UPDRS empleando modelos predictivos para abordar un problema de regresión: Regresión Múltiple, Ventana de Parzen, Redes Neuronales Artificiales, Random Forest y Regresión por Vectores de Soporte. Estos modelos se entrenan y validan usando la base de datos Parkinson's Telemonitoring Dataset, la cual está disponible en línea en el UC Irvine Machine Learning Repository. Los resultados de este estudio indican que es posible hacer predicciones del valor *motor*-UPDRS con un margen de error de aproximadamente 7 puntos, lo cual son predicciones muy cercanas a las observaciones de los médicos.

Palabras clave: Aprendizaje de Máquina, Aprendizaje Automático, Regresión, Modelamiento Fenomenológico, Enfermedad de Parkinson.

I. DESCRIPCIÓN DEL PROBLEMA

Las observaciones de pruebas físicas para pacientes con la enfermedad de Parkinson son mapeadas a una métrica específicamente diseñada para hacer seguimiento a la progresión de la enfermedad; típicamente se utiliza la escala UPDRS (*Unified Parkinson's Disease Rating Scale*) que refleja la presencia y severidad de los síntomas [1]. Esta escala está compuesta de tres secciones: actividad mental, comportamiento y estado de ánimo; actividades de la vida diaria; y funciones motoras. La escala UPDRS para las funciones motoras (*motor*-UPDRS) oscila entre 0-108, donde 0 denota ausencia de síntomas y 108 denota un severo deterioro; *motor*-UPDRS abarca tareas tales como el habla, la expresión facial, el temblor y la rigidez [1]. Mayores valores de la escala UPDRS indican una enfermedad más severa.

La base de datos a utilizar está compuesta por un conjunto de características biomédicas de la voz obtenidas de 42 pacientes con la

enfermedad de Parkinson en etapa inicial, quienes participaron de una prueba de telemonitorización para hacer seguimiento durante seis meses al estado y avance de su enfermedad. Las grabaciones fueron capturadas en las casas de los pacientes.

Estas características (16 en total) establecen una relación entre las propiedades de señal de la voz y el valor UPDRS [1]. Por lo tanto, esta base de datos nos permite abordar un problema de regresión para predecir el valor de la escala UPDRS para las funciones motoras (*motor*-UPDRS) a partir de estas características, y así estimar el estado de la enfermedad para un paciente en un momento dado.

II. VARIABLES

En total, las columnas del conjunto de datos contienen el número del sujeto, la edad, el género, el intervalo de tiempo desde la fecha de reclutamiento inicial, el valor *motor*-UPDRS, el valor *total*-UPDRS, y las medidas de la voz. Sin embargo, el objetivo principal de este estudio es predecir el valor de la escala UPDRS para las funciones motoras (*motor*-UPDRS) a partir de las 16 medidas biomédicas de la voz. En la Tabla I se muestra la descripción de las características a utilizar y la variable a predecir, cuyos valores corresponden a números reales.

TABLA I. DESCRIPCIÓN DE CARACTERÍSTICAS Y VARIABLE A PREDECIR

Descripción	Atributo	#
valor clínico de la escala UPDRS para las funciones motoras (variable a predecir).	motor-UPDRS	16
varias medidas de variación en frecuencia.	Jitter(%)	0
	Jitter(Abs)	1
	Jitter:RAP	2
	Jitter:PPQ5	3
	Jitter:DDP	4
varias medidas de variación en amplitud.	Shimmer	5

	Shimmer(dB)	6
	Shimmer:APQ3	7
	Shimmer:APQ5	8
	Shimmer:APQ11	9
	Shimmer:DDA	10
dos medidas de relación de ruido a componentes tonales en la voz.	NHR	11
	HNR	12
una medida de complejidad dinámica no lineal.	RPDE	13
exponente de escala fractal de señal.	DFA	14
una medida no lineal de variación de frecuencia fundamental.	PPE	15

La base de datos no contiene valores faltantes.

III. TRABAJOS RELACIONADOS

A continuación resumimos el análisis de cuatro artículos que abordan el mismo problema de aprendizaje que nosotros enfrentamos y utilizan la misma base de datos que nosotros empleamos: *Parkinson's Telemonitoring Data Set*.

1. *Accurate Telemonitoring of Parkinson's Disease Progression by Noninvasive Speech Tests* [1].

Este artículo es la referencia principal de la base de datos en cuestión. Los métodos de predicción usados son: *Least Absolute Shrinkage and Selection Operator* (LASSO) y *Classification and Regression Trees* (CART). La metodología de validación usada fue validación cruzada con 10 subconjuntos, en la cual se ejecutaron 1000 iteraciones. Para la evaluación del error al predecir el valor *motor-UPDRS* se utilizó la métrica MAE (*Mean Absolute Error*) y los resultados obtenidos fueron los siguientes: Para LASSO el MAE fue de 6.57 con una desviación estándar de 0.17 y para CART el MAE fue de 5.95 con una desviación estándar de 0.19.

2. *A comparison of regression methods for remote tracking of Parkinson's disease progression* [2].

En este artículo se evalúan los siguientes modelos de regresión: máquinas de soporte vectorial (SVM), máquinas de soporte vectorial de mínimos cuadrados (LS-SVM), redes neuronales perceptrón multicapa (MLPNN) y redes neuronales de regresión generalizadas (GRNN). Como metodología de validación para evaluar el desempeño de los modelos se emplea validación cruzada con 10 subconjuntos y 100 iteraciones. Como medidas de error para la evaluación de los métodos de regresión se utilizan las siguientes métricas: error absoluto medio (MAE), error cuadrático medio (MSE) y el coeficiente de correlación (R).

En la Tabla II se muestran los resultados obtenidos en la validación de los cuatro métodos al predecir el valor *motor-UPDRS* y considerando las 16 variables biomédicas de la voz. Se infiere que el método LS-SVM provee mejores resultados que los demás.

TABLA II. RESULTADOS DE VALIDACIÓN EN LA PREDICCIÓN DE LOS MODELOS PARA *motor-UPDRS* [2]

	MAE	MSE	R
LS-SVM	4.96 (0.14)	39.47 (2.09)	0.63
MLPNN	5.61 (0.16)	48.59 (3.29)	0.51
SVM	5.46 (0.14)	46.73 (2.09)	0.54
GRNN	6.36 (0.14)	59.85 (2.39)	0.30

3. *An analytical method for measuring the Parkinson's disease progression: A case on a Parkinson's telemonitoring dataset* [3].

En este artículo se propone un método híbrido para abordar el problema, en el cual se combinan técnicas de aprendizaje supervisado y no supervisado. El método consiste en aplicar SVD (*Singular Value Decomposition*) para reducción de dimensionalidad, SOM y EM para clustering y un ensamble de modelos ANFIS para la predicción del valor UPDRS; *Adaptive Neuro-Fuzzy Inference System* (ANFIS) es una técnica que incluye lógica difusa y un enfoque de redes neuronales. Para evaluar los modelos ANFIS, los datos fueron divididos en tres grupos: *test set* (20%), *checking test* (20%) y *training set* (60%). Como medidas de error para la evaluación del método se utilizan las siguientes métricas: coeficiente de determinación (R^2), MAE y RMSE (*Root Mean Square Error*). Para comparar el método propuesto se llevaron a cabo diferentes experimentos sobre el mismo conjunto de datos utilizando otros métodos: ANFIS, redes neuronales (NN), regresión por vectores de soporte (SVR) y regresión lineal múltiple (MLR). La Tabla III presenta estos resultados, de los que se infiere que el método propuesto que usa clustering (SOM, EM), SVD y el ensamble de ANFIS tienen un mejor desempeño en relación a los demás modelos.

TABLA III. RESULTADOS EN LA PREDICCIÓN DEL VALOR *motor-UPDRS* [3]

	RMSE	MAE	R^2
MLR	2.4142	0.997	0.697
NN	2.3836	0.977	0.719
ANFIS	1.7047	0.771	0.785
SVR	1.4942	0.721	0.814
SOM-SVD-ANFIS ensemble	0.6921	0.496	0.951
EM-SVD-ANFIS ensemble	0.6867	0.491	0.956

4. *Accuracy Improvement for Predicting Parkinson's Disease Progression* [4].

En este trabajo se utilizan los siguientes modelos para predecir la progresión de la enfermedad de Parkinson a través del valor *motor-UPDRS*: *Adaptive Neuro-Fuzzy Inference System* (ANFIS), redes neuronales (NN) y regresión por vectores de soporte (SVR); se emplea PCA como técnica de reducción de dimensión, y además, se añade una aproximación en la que se utiliza un proceso previo de clustering usando el método de esperanza y maximización (EM)

antes de realizar la predicción. Como metodología de validación se emplea validación cruzada con 10 subconjuntos. Los modelos son evaluados a través de las siguientes métricas: error absoluto medio (MAE) y coeficiente de determinación (R^2). En la Tabla IV se muestran los resultados obtenidos en la cual se evidencia que, los modelos en los que se aplicó un proceso previo de clustering mediante la técnica EM obtienen los mejores resultados.

TABLA IV. RESULTADOS EN LA PREDICCIÓN DEL VALOR motor-UPDRS [4]

	MAE	R^2
PCA-NN	0.861	0.721
PCA-ANFIS	0.662	0.791
PCA-SVR	0.611	0.825
EM-PCA-ANFIS	0.585	0.887
EM-PCA-SVR	0.4721	0.977

IV. EXPERIMENTOS

Para el entrenamiento y validación de los modelos de aprendizaje automático que se usan en este estudio, empleamos la base de datos *Parkinson's Telemonitoring Data Set*, la cual está disponible en línea en el *UC Irvine Machine Learning Repository*.¹ Este conjunto de datos fue construido por Athanasios Tsanas y Max Little de la Universidad de Oxford en colaboración con 10 centros médicos en Estados Unidos y la corporación Intel [1]. La base de datos está compuesta por 5875 muestras correspondientes a 42 pacientes en estado inicial de la enfermedad de Parkinson, quienes atendieron un programa de telemonitoreo de la enfermedad durante seis meses. Hay aproximadamente 200 muestras por paciente, cada una de las cuales está compuesta por 22 variables que incluyen algunos datos básicos del paciente, los valores clínicos *total-UPDRS* y *motor-UPDRS*, y 16 variables asociadas a grabaciones de la voz del paciente. No obstante, el objetivo principal de este estudio es predecir el valor de la escala UPDRS para las funciones motoras (*motor-UPDRS*) a partir de las 16 medidas biomédicas de la voz.

La metodología de validación usada en este estudio es *GroupSuffleSplit*; esta es una variante de la metodología de validación *Bootstrap*, tal que se realiza la división del conjunto de datos teniendo en cuenta una agrupación del mismo [5]. En este caso, las 5875 muestras están divididas en 42 grupos que corresponden a los pacientes. Aplicando *GroupSuffleSplit* garantizamos que muestras de un mismo paciente no queden combinadas en los conjuntos de entrenamiento y validación; combinar tales muestras en estos subconjuntos puede provocar que el sistema adquiera un sesgo perdiendo la capacidad de generalización, es decir, la capacidad del sistema de predecir correctamente nuevas muestras. Sin importar la muestra que se le presente al modelo, este

debería comportarse adecuadamente independientemente del sujeto asociado a la muestra.

Teniendo en cuenta lo anterior, para cada modelo se aplica *GroupSuffleSplit* con 10 iteraciones, dividiendo aleatoriamente el conjunto de datos en una proporción 70% para entrenamiento y 30% para validación, lo que corresponde aproximadamente a 29 y 13 grupos (pacientes) respectivamente.

Para normalizar los datos se utiliza el tipo de normalización *zscore*, tal que todas las variables quedan con media 0 y varianza 1:

$$x_n = \frac{x - \text{mean}(X)}{\text{std}(X)}$$

Las medidas de error empleadas para evaluar los modelos son: MAE (*Mean Absolute Error*), MAPE (*Mean Absolute Percentage Error*) y RMSE (*Root Mean Squared Error*):

$$\text{Error(MAE)} = \frac{1}{N} \sum_{i=1}^N |f(x_i) - y_i|$$

$$\text{Error(MAPE)} = \frac{1}{N} \sum_{i=1}^N \left| \frac{f(x_i) - y_i}{y_i} \right|$$

$$\text{Error(RMSE)} = \left(\frac{1}{N} \sum_{i=1}^N (f(x_i) - y_i)^2 \right)^{\frac{1}{2}}$$

En las ecuaciones anteriores $f(x_i)$ es la i -ésima salida predicha por el modelo y y_i es la i -ésima salida real (salida objetivo), donde N es el número de predicciones.

V. RESULTADOS DE LOS MODELOS DE PREDICCIÓN

En este estudio se emplean los siguientes modelos predictivos para abordar el problema de regresión: Regresión Múltiple, Ventana de Parzen, Redes Neuronales Artificiales, Random Forest y Regresión por Vectores de Soporte (con kernel lineal y kernel RBF). Inicialmente, se entrenan estos modelos sin realizar ningún análisis previo de las variables del problema, es decir, que en este punto se emplean las 16 variables asociadas a las grabaciones de la voz del paciente. Las métricas de error empleadas en cada modelo fueron:

- Regresión múltiple → MAE, MAPE, RMSE.
- Ventana de Parzen → MAE, MAPE.
- Redes Neuronales Artificiales → MAE, MAPE, RMSE.
- Random Forest → MAE, MAPE.
- Regresión por Vectores de Soporte (SVR) → MAE, MAPE, RMSE.

Los resultados obtenidos de la búsqueda en malla fueron los siguientes:

TABLA V. RESULTADOS MODELO DE REGRESIÓN MÚLTIPLE

params	mean_test_mae	std_test_mae	mean_test_mape	std_test_mape	mean_test_rmse	std_test_rmse
{'degree'	7.113	0.258	0.424	0.080	8.577	0.4485

¹ <https://archive.ics.uci.edu/ml/datasets/Parkinsons+Telemonitoring>

{ 1}						
{'degree': 2}	8.240	1.386	0.469	0.104	1,29E+01	7.00
{'degree': 3}	42.478	51.502	1.875	1.945	5,51E+02	8.42e+02
{'degree': 4}	96705.6	112886.2	5125.89	4922.5	1,70E+06	1.35e+06

De la Tabla V se puede inferir que el mejor resultado para el modelo de regresión múltiple se obtuvo con un polinomio de grado 1. Además, es importante resaltar que a medida que aumentaba el grado del polinomio se notaba claramente como se sobreajustaba el modelo. Por ejemplo, los peores resultados de *test* fueron con el polinomio de grado 4, sin embargo, éste obtuvo los mejores resultados de entrenamiento con un MAE de 0.0002, un MAPE de 0.00001 y un RMSE de 0.0004.

TABLA VI. RESULTADOS MODELO VENTANA DE PARZEN

	MAE	std_MAE	MAPE	std_MAPE
Ancho del kernel				
0.5	10.183	1.581	0.431	0.046
1	7.461	0.912	0.391	0.055
2	7.045	0.678	0.415	0.084
5	6.993	0.617	0.425	0.090
15	6.981	0.598	0.427	0.091

Según los resultados de la Tabla VI, el mejor hiperparámetro para el modelo no paramétrico ventana de parzen es un ancho de ventana (o de kernel) igual a 15. Es importante resaltar que con un ancho de kernel de 0.01, al ser un valor muy pequeño, el algoritmo falló; esto sucede debido a que con este valor de ventana las distancias calculadas serán muy grandes, resultando en que la función kernel (que corresponde a una exponencial negativa) retorna valores que tienden a cero, creando así un vector de valores NaN.

TABLA VII. RESULTADOS MODELO REDES NEURONALES ARTIFICIALES

params	mean_test_mae	std_test_mae	mean_test_mape	std_test_mape	mean_test_rmse	std_test_rmse
{'hidden_layer_sizes': 8, 'max_iter': 1000}	7.610	0.557048	0.448084	0.095644	9.190	0.627202
{'hidden_layer_sizes': 8, 'max_iter': 1500}	7.636	0.571398	0.450647	0.099410	9.227	0.617105
{'hidden_layer_sizes': 16, 'max_iter': 1000}	7.698	0.532208	0.453039	0.099645	9.314	0.582912

{'hidden_layer_sizes': 16, 'max_iter': 1500}	7.803	0.466114	0.458128	0.098914	9.433	0.503035
{'hidden_layer_sizes': 40, 'max_iter': 1000}	7.922	0.513800	0.464910	0.100005	9.577	0.609392
{'hidden_layer_sizes': 40, 'max_iter': 1500}	7.915	0.519743	0.461990	0.093813	9.582	0.601719
{'hidden_layer_sizes': (8, 8), 'max_iter': 1000}	7.756	0.457339	0.455148	0.095945	9.432	0.441799
{'hidden_layer_sizes': (8, 8), 'max_iter': 1500}	7.963	0.503253	0.468960	0.101116	9.650	0.556203
{'hidden_layer_sizes': (16, 16), 'max_iter': 1000}	8.112	0.502662	0.476627	0.098279	9.824	0.524561
{'hidden_layer_sizes': (16, 16), 'max_iter': 1500}	8.111	0.511142	0.473273	0.095218	9.861	0.583598
{'hidden_layer_sizes': (40, 40), 'max_iter': 1000}	8.203	0.535117	0.475539	0.099968	9.987	0.594247
{'hidden_layer_sizes': (40, 40), 'max_iter': 1500}	8.320	0.542926	0.482316	0.093952	10.176	0.610761
{'hidden_layer_sizes': (8, 8, 8), 'max_iter': 1000}	7.858	0.516494	0.462476	0.099287	9.524	0.563871
{'hidden_layer_sizes': (8, 8, 8), 'max_iter': 1500}	7.781	0.568115	0.458611	0.101903	9.478	0.597111
{'hidden_layer_sizes': (16, 16, 16), 'max_iter': 1000}	8.055	0.508514	0.471980	0.099158	9.836	0.514696
{'hidden_layer_sizes': (16, 16, 16), 'max_iter': 1500}	8.220	0.601432	0.479815	0.105475	10.012	0.635736
{'hidden_layer_sizes': (40, 40, 40), 'max_iter': 1000}	8.468	0.552332	0.492953	0.098952	10.421	0.586546
{'hidden_layer_sizes': (40, 40, 40), 'max_iter': 1500}	8.426	0.441438	0.490709	0.090939	10.405	0.421211

Tal como se muestra en la Tabla VII, el entrenamiento de los modelos de redes neuronales artificiales se realizó teniendo en cuenta las siguientes combinaciones de hiperparámetros:

- Capas ocultas: 1, 2, 3.
- Neuronas en cada capa oculta: 8, 16, 40.
- Número de épocas: 1000, 1500.

Para todas estas combinaciones se usó la misma función de activación en las capas ocultas: tangente hiperbólica.

A partir de la Tabla VII se puede inferir que la mejor combinación de hiperparámetros para este modelo fue: red neuronal con una capa oculta, 8 neuronas en esta capa y 1000 épocas. Sin embargo, se intentó reducir la complejidad del modelo que presentó los mejores resultados sobre el conjunto de entrenamiento (y por lo tanto el modelo más propenso al sobreajuste), a través de la estrategia de parada anticipada y del método de regularización con norma L2. Dicho modelo corresponde al que tenía la siguiente configuración: red neuronal con 3 capas ocultas de 40 neuronas cada una y 1500 épocas. Con esta configuración se obtuvieron los siguientes valores de error en el entrenamiento: MAE = 2.5, MAPE = 0.15 y RMSE = 3.417.

Así, aplicando la estrategia de parada anticipada con un valor de tolerancia de 0.1 y un valor de paciencia de 10 iteraciones, se obtuvo una mejora en los resultados de *test* (MAE = 6.868 ± 1.04 , MAPE = 0.384 y RMSE = 8.192), superando incluso al mejor modelo de la búsqueda en malla. Sin embargo, se debe tener en cuenta que la estrategia de parada anticipada implementada en la librería *Sklearn*, separa automáticamente y de manera aleatoria, el 10% del subconjunto de entrenamiento como subconjunto de validación; por lo tanto, esto presenta una limitante debido a la necesidad de realizar las particiones de subconjuntos considerando los grupos con que cuenta el conjunto de datos.

Teniendo en cuenta lo anterior, se aplicó también regularización sobre este mismo modelo empleando la norma L2, con un valor del parámetro de regularización $\alpha = 100$. En este caso se obtuvieron resultados similares a la aproximación anterior, pero sin la limitante que ésta presentaba: MAE = 6.90 ± 0.713 , MAPE = 0.378 y RMSE = 8.246.

TABLA VIII. RESULTADOS MODELO RANDOM FOREST

params	mean_test_mae	std_test_mae	mean_test_mape	std_test_mape
{'max_features': 5, 'n_estimators': 5}	7.736	0.342465	0.458462	0.083408
{'max_features': 5, 'n_estimators': 10}	7.567	0.367710	0.450929	0.086070
{'max_features': 5, 'n_estimators': 20}	7.403	0.367118	0.443255	0.080878
{'max_features': 5, 'n_estimators': 50}	7.324	0.330162	0.438989	0.084031
{'max_features': 5, 'n_estimators': 100}	7.320	0.334875	0.439408	0.084469
{'max_features': 10, 'n_estimators': 5}	7.809	0.303777	0.465035	0.090379
{'max_features': 10, 'n_estimators': 10}	7.631	0.313510	0.455357	0.086818

{'max_features': 10, 'n_estimators': 20}	7.465	0.329436	0.446287	0.085697
{'max_features': 10, 'n_estimators': 50}	7.410	0.360195	0.444222	0.087167
{'max_features': 10, 'n_estimators': 100}	7.413	0.351831	0.444205	0.086081
{'max_features': 16, 'n_estimators': 5}	7.804	0.395679	0.462756	0.090221
{'max_features': 16, 'n_estimators': 10}	7.607	0.360495	0.452266	0.085135
{'max_features': 16, 'n_estimators': 20}	7.561	0.393079	0.451953	0.086875
{'max_features': 16, 'n_estimators': 50}	7.466	0.378086	0.446382	0.086330
{'max_features': 16, 'n_estimators': 100}	7.453	0.370984	0.446006	0.086652

Como se evidencia en la Tabla VIII, para el entrenamiento del modelo Random Forest se emplearon las siguientes combinaciones de hiperparámetros:

- Número de estimadores (árboles): 5, 10, 20, 50, 100.
- Máximo número de características a evaluar para las particiones: 5, 10, 16.

TABLA IX. RESULTADOS MODELO DE REGRESIÓN POR VECTORES DE SOPORTE

params	mean_test_mae	std_test_mae	mean_test_mape	std_test_mape	mean_test_rmse	std_test_rmse
{'C': 1000.0, 'gamma': 0.1, 'kernel': 'linear'}	7.500	0.481315	0.4207 20	0.06642 5	9.180	0.7588 17
{'C': 1000.0, 'gamma': 0.1, 'kernel': 'rbf'}	8.488	0.490063	0.4936 30	0.08424 4	10.322	0.5988 86
{'C': 1000.0, 'gamma': 1, 'kernel': 'linear'}	7.500	0.481315	0.4207 20	0.06642 5	9.180	0.7588 17
{'C': 1000.0, 'gamma': 1, 'kernel': 'rbf'}	8.949	0.431197	0.5323 04	0.09481 3	11.207	0.5665 69
{'C': 1, 'gamma': 0.1, 'kernel': 'linear'}	7.487	0.467114	0.4200 68	0.06723 7	9.123	0.7057 11
{'C': 1, 'gamma': 0.1, 'kernel': 'rbf'}	7.455	0.452823	0.4318 73	0.08068 4	8.953	0.5355 30
{'C': 1, 'gamma': 1, 'kernel': 'linear'}	7.487	0.467114	0.4200 68	0.06723 7	9.123	0.7057 11
{'C': 1, 'gamma': 1, 'kernel': 'rbf'}	7.403	0.395171	0.4405 45	0.08093 0	8.862	0.4789 66
{'C': 0.001, 'gamma': 0.1, 'kernel': 'linear'}	7.159	0.709951	0.4128 01	0.07703 3	8.446	0.8091 90
{'C': 0.001, 'gamma': 0.1, 'kernel': 'rbf'}	7.173	0.740053	0.4232 26	0.08625 5	8.454	0.7772 21

'kernel': 'rbf'}						
{'C': 0.001, 'gamma': 1, 'kernel': 'linear'}	7.159	0.709951	0.4128 01	0.07703 3	8.446	0.8091 90
{'C': 0.001, 'gamma': 1, 'kernel': 'rbf'}	7.169	0.745879	0.4231 84	0.08648 7	8.452	0.7816 32
{'C': 3e-05, 'gamma': 0.1, 'kernel': 'linear'}	7.160	0.746203	0.4217 01	0.08523 8	8.442	0.7844 80
{'C': 3e-05, 'gamma': 0.1, 'kernel': 'rbf'}	7.169	0.747791	0.4232 07	0.08646 1	8.453	0.7832 22
{'C': 3e-05, 'gamma': 1, 'kernel': 'linear'}	7.160	0.746203	0.4217 01	0.08523 8	8.442	0.7844 80
{'C': 3e-05, 'gamma': 1, 'kernel': 'rbf'}	7.169	0.747914	0.4232 06	0.08646 2	8.452	0.7832 53

Como fue mencionado previamente en esta sección, el modelo de regresión por vectores de soporte se entrenaría usando un kernel lineal y un kernel RBF. Adicionalmente, se evaluaron diferentes valores del parámetro de regularización 'C' y del ancho de la ventana 'gamma' para el kernel RBF así:

- Parámetro de regularización 'C': 1e3, 1, 1e-3, 3e-5.
- Ancho de ventana 'gamma': 0.1, 1.

Teniendo en cuenta todos los resultados anteriores, los tres mejores modelos son:

1. Red Neuronal Artificial con 3 capas ocultas de 40 neuronas cada una, 1500 épocas y regularización con normal L2 ($\alpha = 100$) \rightarrow MAE = 6.90 ± 0.713
2. Ventana de Parzen con un ancho del kernel $h = 15 \rightarrow$ MAE = 6.981 ± 0.598
3. Máquina de Vectores de Soporte con kernel lineal y parámetro de regularización $C = 1e-3 = 0.001 \rightarrow$ MAE = 7.159 ± 0.709

VI. ANÁLISIS INDIVIDUAL DE CARACTERÍSTICAS

Teniendo en cuenta que se está abordando un problema de regresión, para realizar el análisis individual de características se emplea el coeficiente de correlación de Pearson. Este es un índice que mide la relación lineal entre dos variables aleatorias, es independiente de la unidad de medida y toma valores entre -1 y 1, donde [6]:

- $\rho = 1 \rightarrow$ existe correlación positiva perfecta.
- $0 < \rho < 1 \rightarrow$ existe correlación positiva.
- $\rho = 0 \rightarrow$ no existe relación lineal.
- $-1 < \rho < 0 \rightarrow$ existe correlación negativa.
- $\rho = -1 \rightarrow$ existe correlación negativa perfecta.

No obstante, se debe tener en cuenta que este índice tiene una limitante, y es que mide exclusivamente la relación **lineal** entre variables.

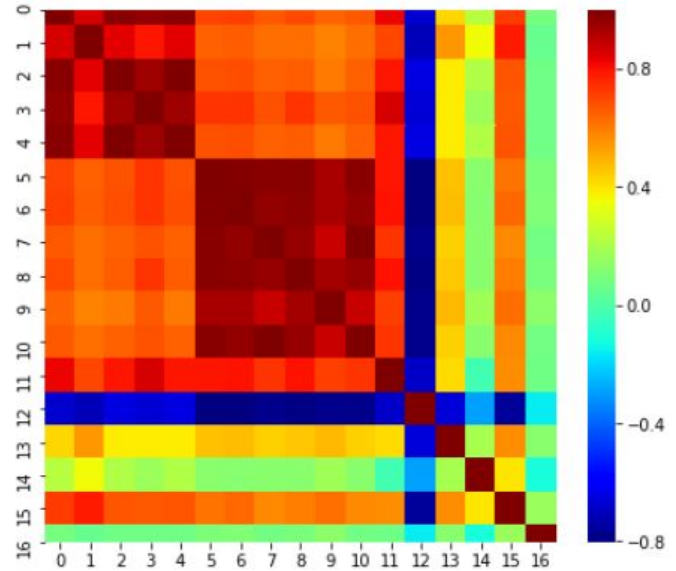


Fig. 1. Correlación lineal entre variables

Los siguientes análisis se hacen teniendo en cuenta la numeración asignada a los atributos y la variable de salida que se muestran en la Tabla I (sección II: VARIABLES).

La situación esperada es que las variables de entrada deben tener una baja correlación entre sí, y una alta correlación con la variable de salida. Sin embargo, lo que se observa en la Figura 1 es justamente lo contrario. Más específicamente, las variables 0, 1, 2, 3 y 4 presentan una alta correlación positiva entre ellas, al igual que las variables 5, 6, 7, 8, 9 y 10; lo cual deriva en una situación de redundancia. Esto no carece de sentido, ya que el primer grupo de variables se agrupan como medidas de variación en frecuencia, mientras que el segundo, se agrupan como medidas de variación en amplitud. También, se evidencia que la variable 12 presenta una correlación negativa (llegando incluso a ser alta) con las restantes 15 variables; la variable 13 presenta una correlación positiva moderada con las demás variables; y la variable 14 no presenta demasiada correlación (ya sea positiva o negativa) con las demás variables. Además, es importante resaltar que no se evidencia una correlación lineal alta entre los atributos y la variable de salida; en general se observa una leve correlación positiva, y en dos casos una leve correlación negativa.

En síntesis, varias de las 16 medidas biomédicas de la voz están altamente correlacionadas entre sí, lo que sugiere prescindir de varias de estas variables que son redundantes y no aportan de manera significativa a la predicción del valor *motor*-UPDRS. Por ejemplo, del grupo de variables que se agrupan como medidas de variación en frecuencia se podrían descartar cuatro de ellas, y dejar la que está más correlacionada con la variable de salida; análogamente, para el grupo de variables que se agrupan como medidas de variación en amplitud se podrían excluir cinco de ellas.

VII. SELECCIÓN DE CARACTERÍSTICAS

Para abordar la reducción de dimensionalidad del conjunto de datos a través de la estrategia de selección de variables, es necesario definir dos componentes: una estrategia de búsqueda y un criterio de selección.

Como estrategia de búsqueda se utiliza selección secuencial hacia adelante o SFS (*Sequential Forward Selection*) por sus siglas en Inglés. Se decide elegir esta estrategia (comparada con SBS - *Sequential Backward Selection* -) debido a que presenta mejor desempeño cuando el conjunto óptimo tiene un número de características bajo, debido a que esta estrategia empieza con un subconjunto vacío y procede a iterar añadiendo variables; a partir del análisis individual de características presentado en la sección anterior, donde se evidenció en general una alta correlación positiva entre varias variables, se podría intuir que el subconjunto óptimo tendría pocos elementos. Sin embargo, esta estrategia presenta una limitante: es poco flexible porque una vez que una característica es adicionada al subconjunto, no se podrá remover en caso de que esta variable llegase a quedar obsoleta.

Como criterio de selección se utiliza un criterio tipo Wrapper: la función objetivo es un modelo de aprendizaje. Se decide emplear este criterio debido a que presenta una ventaja en exactitud sobre los criterios tipo filtro, es decir, permite obtener mejores tasas de predicción debido a que la función objetivo se ajusta específicamente para reducir el error de validación. No obstante, sus principales desventajas son los tiempos de ejecución (en general lentos) y la falta de generalidad de los subconjuntos obtenidos.

En este caso, los modelos a evaluar son los tres mejores modelos presentados en la sección V (RESULTADOS DE LOS MODELOS DE PREDICCIÓN); para cada modelo se aplica una metodología de validación apropiada usando *GroupSuffleSplit* con 10 iteraciones, y se evalúan subconjuntos de características compuestos con una (1) única variable, hasta el total de variables (16).

Los resultados obtenidos fueron los siguientes:

TABLA X. RESULTADOS SELECCIÓN DE CARACTERÍSTICAS
MODELO DE REDES NEURONALES ARTIFICIALES

	feature_idx	avg_score (MAE)	std_dev
1	(14,)	7.111	0.454989
2	(10, 14)	7.301	0.340495
3	(4, 10, 14)	7.363	0.36626
4	(4, 10, 11, 14)	7.453	0.320988
5	(1, 4, 10, 11, 14)	7.557	0.371331
6	(1, 2, 4, 10, 11, 14)	7.578	0.352208
7	(1, 2, 4, 10, 11, 13, 14)	7.634	0.479677
8	(1, 2, 4, 10, 11, 12, 13, 14)	7.651	0.478337
9	(1, 2, 4, 10, 11, 12, 13, 14, 15)	7.656	0.419925

10	(1, 2, 4, 7, 10, 11, 12, 13, 14, 15)	7.690	0.418561
11	(1, 2, 4, 7, 8, 10, 11, 12, 13, 14, 15)	7.698	0.439545
12	(0, 1, 2, 4, 7, 8, 10, 11, 12, 13, 14, 15)	7.725	0.542257
13	(0, 1, 2, 4, 6, 7, 8, 10, 11, 12, 13, 14, 15)	7.717	0.517568
14	(0, 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 13, 14, 15)	7.685	0.480189
15	(0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, ...)	7.707	0.549089
16	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...)	7.583	0.556207

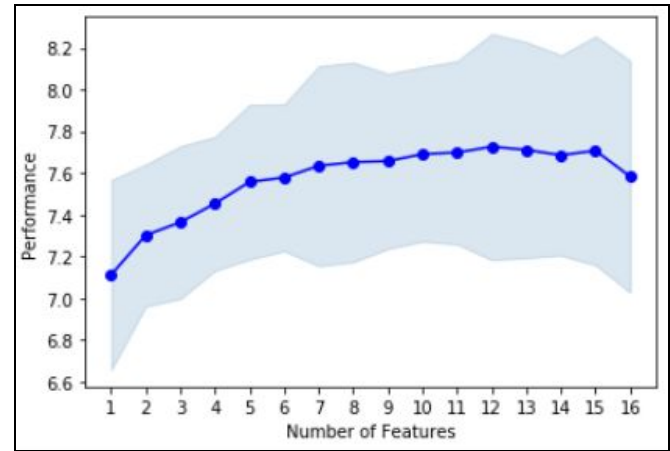


Fig. 2. Selección de características modelo redes neuronales artificiales

Resultado de este modelo en la búsqueda en malla: MAE = 7.610 ± 0.557

TABLA XI. RESULTADOS SELECCIÓN DE CARACTERÍSTICAS
MODELO VENTANA DE PARZEN

	feature_idx	avg_score (MAE)	std_dev
1	(7,)	8.136	0.4516
2	(7, 10)	7.915	0.45341
3	(7, 8, 10)	7.034	0.574067
4	(4, 7, 8, 10)	7.017	0.601296
5	(2, 4, 7, 8, 10)	7.020	0.605058
6	(2, 3, 4, 7, 8, 10)	7.020	0.60792
7	(1, 2, 3, 4, 7, 8, 10)	7.019	0.603792
8	(0, 1, 2, 3, 4, 7, 8, 10)	7.019	0.606976
9	(0, 1, 2, 3, 4, 7, 8, 10, 11)	7.015	0.6075
10	(0, 1, 2, 3, 4, 5, 7, 8, 10, 11)	7.011	0.60593
11	(0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11)	7.007	0.605004
12	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)	7.001	0.605719
13	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)	6.993	0.595254

14	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)	6.990	0.591656
15	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...)	6.986	0.598946
16	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...)	6.973	0.563214

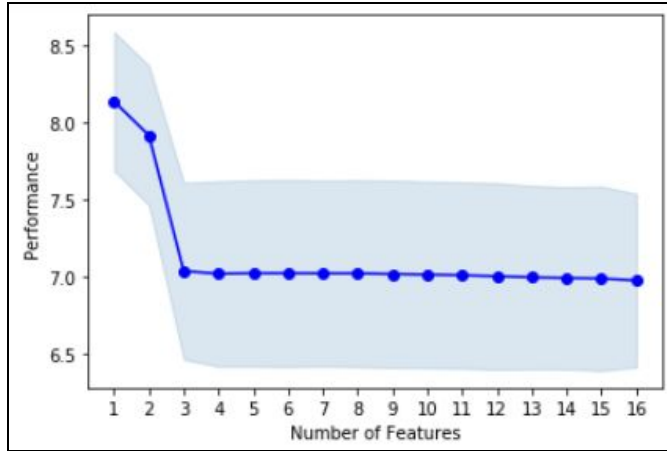


Fig. 3. Selección de características modelo ventana de Parzen

Resultado de este modelo en la búsqueda en malla: $MAE = 6.981 \pm 0.598$

TABLA XII. RESULTADOS SELECCIÓN DE CARACTERÍSTICAS MODELO DE REGRESIÓN POR VECTORES DE SOPORTE

	feature_idx	avg_score (MAE)	std_dev
1	(14,)	7.184	0.674606
2	(1, 14)	7.194	0.69049
3	(1, 4, 14)	7.194	0.695393
4	(1, 2, 4, 14)	7.196	0.699874
5	(1, 2, 3, 4, 14)	7.197	0.705911
6	(1, 2, 3, 4, 11, 14)	7.197	0.703584
7	(1, 2, 3, 4, 10, 11, 14)	7.196	0.698643
8	(1, 2, 3, 4, 7, 10, 11, 14)	7.197	0.696788
9	(0, 1, 2, 3, 4, 7, 10, 11, 14)	7.196	0.698433
10	(0, 1, 2, 3, 4, 7, 8, 10, 11, 14)	7.192	0.694135
11	(0, 1, 2, 3, 4, 5, 7, 8, 10, 11, 14)	7.187	0.694253
12	(0, 1, 2, 3, 4, 5, 7, 8, 10, 11, 13, 14)	7.182	0.70222
13	(0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14)	7.178	0.699169
14	(0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14)	7.173	0.699621
15	(0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14,...)	7.170	0.71058
16	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...)	7.159	0.709951

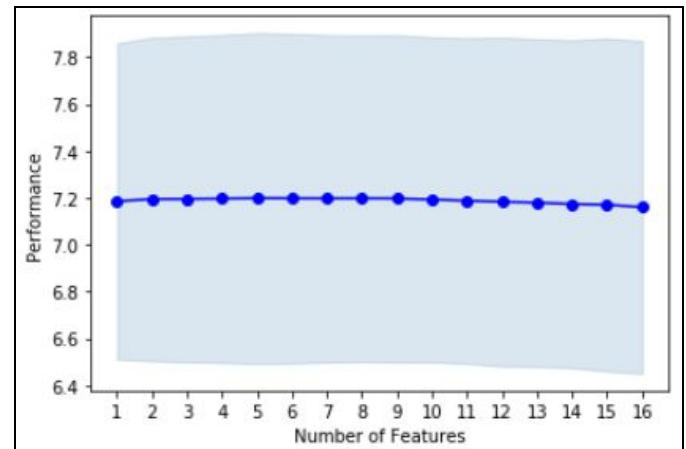


Fig. 4. Selección de características modelo de regresión por vectores de soporte

Resultado de este modelo en la búsqueda en malla: $MAE = 7.159 \pm 0.709$

Es importante clarificar algunos aspectos que se deben tener en cuenta en la interpretación de estos resultados:

- La medida de desempeño utilizada fue exclusivamente MAE (*Mean Absolute Error*), a pesar de que en la búsqueda en malla se mostraron también otras métricas (MAPE y RMSE).
- Los *feature_idx* de las características corresponden a la numeración asignada a los atributos que se muestran en la Tabla I (sección II: VARIABLES).
- Los resultados de la Tabla X, se obtuvieron utilizando el mejor modelo de la búsqueda en malla para el modelo de redes neuronales artificiales; no se utilizó el modelo regularizado con normal L2 que se presentó como el primero de los mejores tres modelos al final de la sección V (RESULTADOS DE LOS MODELOS DE PREDICCIÓN). Lo anterior debido a que este modelo regularizado, a causa de su complejidad (y por lo tanto su número de parámetros), presentaba problemas en tiempos de cómputo.

De la Tabla X y la Figura 2, se puede observar que para el modelo de redes neuronales artificiales, el mejor resultado en promedio se obtuvo con un subconjunto con una única característica: la característica número 14; este resultado presenta una mejora con respecto al obtenido en la búsqueda en malla. La Tabla XI y la Figura 3 evidencian que, para el modelo de ventana de Parzen, los subconjuntos con 1 y 2 variables presentan valores de error más altos que con los demás subconjuntos, para los cuales no se nota casi ninguna diferencia en sus valores de error promedio, ni tampoco se evidencia una mejora con respecto al resultado del modelo con un subconjunto completo con las 16 variables. En la Tabla XII y la Figura 4, se muestran los resultados para el mejor modelo de regresión por vectores de soporte, en los cuales se evidencia que el

desempeño es casi el mismo con cualquiera de los subconjuntos, desde una característica hasta el total de características.

Es importante resaltar la preponderancia de la característica número 14 (DFA - exponente de escala fractal de señal -), con la cual se obtienen resultados buenos siendo esta la única variable de un subconjunto con el cual se entrena un modelo de redes neuronales y de regresión por vectores de soporte. Esta variable también se destacó en la sección anterior, en la que se realizó el análisis individual de características, como aquella menos correlacionada con las demás variables del conjunto de datos.

Adicionalmente, en aras de confirmar el argumento expuesto en el párrafo anterior, se llevó a cabo un experimento extra en el que se entrenó el mejor modelo resultante de la búsqueda en malla para el modelo Random Forest (5 características y 100 estimadores), con el objetivo de verificar la importancia que este modelo da a las características en su proceso de entrenamiento:

```
rf = RandomForestRegressor(n_estimators=100, max_features=5)
rf.fit(X_norm, Y)
rf.feature_importances_

array([0.04882928, 0.08969498, 0.04137943, 0.05152607, 0.04289603,
       0.04003284, 0.0400631 , 0.04209095, 0.04336507, 0.06133337,
       0.04467615, 0.06253199, 0.09363687, 0.07882606, 0.13319844,
       0.08591936])
```

Fig. 5. Importancia de características modelo Random Forest

En la Figura 5 se puede observar como la variable número 14 (recorriendo correctamente el arreglo desde la posición 0), es aquella a la que el modelo asigna más importancia, con un valor del 13.3% dentro del conjunto de 16 características.

VIII. EXTRACCIÓN DE CARACTERÍSTICAS

Para realizar la reducción de dimensionalidad del conjunto de datos a través de la estrategia de extracción de variables, se usa en este caso el método PCA (*Principal Component Analysis*). Este método tiene como objetivo realizar una transformación a un nuevo conjunto de variables conservando la mayor cantidad posible de información de las variables iniciales, entendiendo información en términos de variabilidad (una variable constante no aporta información al modelo).

Para aplicar este método, es necesario conocer el número de componentes principales sobre los cuales se efectuará la proyección de los datos. Se decide utilizar la curva de varianza acumulada para determinar el número de componentes, en lugar de un criterio tipo Wrapper (como el que se usó en la sección anterior de selección de características), que se muestra más costoso computacionalmente. La curva de varianza acumulada es un gráfico que nos permite evaluar cómo se va acumulando la variabilidad que aporta cada componente principal, tal que, se pueda seleccionar un número de componentes que cubran una determinada proporción significativa de varianza.

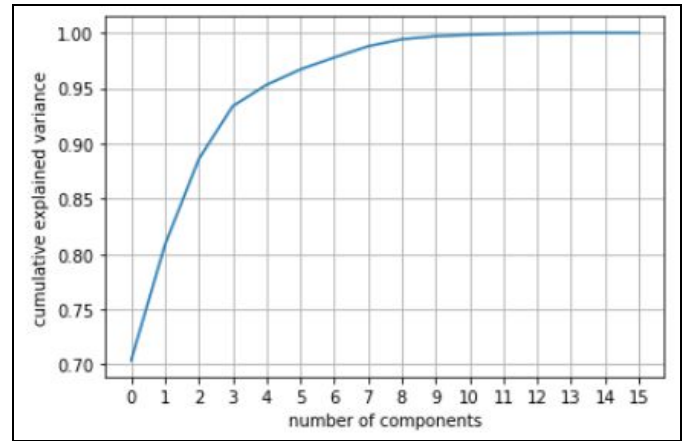


Fig. 6. Curva de varianza acumulada (PCA)

Teniendo en cuenta la Figura 6, se seleccionan 2 valores de número de componentes para evaluar con los tres mejores modelos encontrados en la sección V (RESULTADOS DE LOS MODELOS DE PREDICCIÓN). Estos valores son 4 y 8 componentes, los cuales explican un 93.3% y 98.7% (respectivamente) de la variación presente en el conjunto original de variables. Los resultados obtenidos después de realizar las transformaciones, y aplicar una metodología de validación con *GroupShuffleSplit* y 10 iteraciones, fueron los siguientes:

TABLA XIII. RESULTADOS EXTRACCIÓN DE CARACTERÍSTICAS

		MAE	std_MAE	MAPE	std_MAPE	RMSE	std_RMSE
model	n_comp onents						
MLPR + PCA	4	7.004	1.245	0.4057	0.1257	8.431	1.465
	8	6.973	0.561	0.427	0.0884	8.243	0.5736
Parzen Window + PCA	4	7.168	1.093	0.4419	0.1107	8.333	1.280
	8	7.212	1.076	0.4227	0.1322	8.610	1.039
SVR + PCA	4	6.972	0.5581	0.4273	0.0882	8.244	0.571
	8	7.174	1.097	0.4417	0.11	8.336	1.285

Una de las principales ventajas de este método de reducción (PCA) es que las nuevas variables que se obtienen son no correlacionadas. Sin embargo, de acuerdo con los resultados de la Tabla XIII, no se evidencia una mejora significativa en los resultados de predicción de los 3 modelos evaluados; únicamente se presentó una muy sutil mejoría en el resultado del modelo de regresión por vectores de soporte usando 4 componentes principales.

IX. DISCUSIÓN

Se puede argumentar que los resultados obtenidos en este estudio son aceptables. Los tres mejores modelos resultantes de la búsqueda en malla, presentaron tasas de error entre 6.9 y 7.1 puntos teniendo en cuenta la medida de error MAE (*Mean Absolute Error*). Esta medida nos permite calcular la desviación promedio absoluta del valor predicho por los modelos respecto al valor real; esto quiere

decir que del rango de valores de la variable de salida *motor*-UPDRS (0-108), se pueden hacer predicciones con un margen de error de aproximadamente 7 puntos, lo cual, como indican Tsanas, et al., en [1] son predicciones muy cercanas a las observaciones de los médicos.

La medida de error MAE fue la medida preponderante a lo largo del estudio, fue la base de comparación y sobre la que se tomaron las diferentes decisiones. Sin embargo, a pesar de que la medida MAPE (*Mean Absolute Percentage Error*) podría pensarse más adecuada como función de validación (ya que nos presenta en términos porcentuales la desviación de la salida predicha respecto a la salida esperada), se razonó que en este estudio esta medida no era muy pertinente, razonamiento que surgió a partir de evidenciar que ésta presentaba tasas relativamente altas de error. Esta tendencia de MAPE a entregar valores altos se debe a la metodología de validación empleada en este estudio: aplicando *GroupShuffleSplit*, el conjunto de *test* estará conformado exclusivamente por las muestras de algunos de los pacientes (13 pacientes en este caso, debido a los porcentajes asignados al aplicar la metodología de validación). Esto implica que en validación, esta medida, al calcular el porcentaje de desviación sobre el conjunto de *test*, solo tiene en cuenta los valores de estos 13 pacientes; y no todos los pacientes tienen un valor de la variable *motor*-UPDRS que alcance valores altos (hasta máximo 108) debido a que la mayoría de los pacientes que conforman la base de datos eran pacientes en estado inicial de la enfermedad; a raíz de que algunos pocos pacientes sí presentaron valores altos de *motor*-UPDRS en las etapas finales del proceso de telemonitorización, se podía presentar una discrepancia entre el ajuste del modelo en el entrenamiento y los valores esperados en validación, lo cual se reflejaría en un porcentaje de desviación incongruente con la desviación absoluta.

En el análisis anterior se mencionó el uso de *GroupShuffleSplit* como metodología de validación en este estudio. Es evidente que los resultados obtenidos están muy alejados de los mejores resultados presentados en el estado del arte (sección III: TRABAJOS RELACIONADOS); sin embargo, es pertinente aclarar que estos autores no usaron metodologías de validación que tuvieran en cuenta la división por grupos (pacientes) del conjunto de datos, lo cual se traduce en un sesgo del modelo que reduce su fiabilidad, pero que a su vez permite obtener tasas de error más bajas (han sido ampliamente criticados por este aspecto). Además, es importante considerar que algunos de estos autores emplearon modelos mucho más complejos que los presentados en este estudio, incluyendo incluso etapas de aprendizaje no supervisado para llevar a cabo tareas de *Clustering* con el conjunto de datos; por ejemplo, Nilashi, et al., en [3] argumentan que el objetivo del agrupamiento es encontrar grupos similares de datos para mejorar la precisión en la predicción.

Es destacable que después de realizar el proceso de reducción de dimensionalidad, empleando selección y extracción de características, no se evidenció una mejora significativa en los resultados de los modelos de predicción. En lo que respecta a la

selección de característica, sin embargo, Eskidere et al., exponen una situación similar en [2]: evaluando cuatro subconjuntos de variables diferentes con 6, 13, 3 y 16 elementos, obtienen resultados muy similares, resultando incluso en que el mejor es el que se alcanza con el subconjunto compuesto de la totalidad de variables (16). Respecto a la extracción de características, como se mencionó, no se obtuvo una mejoría en los resultados; no obstante, es importante tener en cuenta la pugna existente entre interpretabilidad y capacidad predictiva. Nilashi, et al., en [3] y [4] aplicaron PCA como técnica de extracción de características, obteniendo un muy buen desempeño en sus modelos predictivos (teniendo en mente también las demás consideraciones ya analizadas); la cuestión que esta situación plantea estriba en el contexto del problema abordado, ya que se intenta construir una herramienta de apoyo al diagnóstico médico: las nuevas variables no tienen interpretación humana específica porque las unidades se distorsionan como resultado de la transformación de los datos.

REFERENCIAS

- [1] A. Tsanas, M. A. Little, P. E. McSharry and L. O. Ramig, "Accurate Telemonitoring of Parkinson's Disease Progression by Noninvasive Speech Tests," in *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 884-893, April 2010.
- [2] Ö. Eskidere, F. Ertaş and C. Haniçi, "A comparison of regression methods for remote tracking of Parkinson's disease progression", *Expert Systems with Applications*, vol. 39, no. 5, 2012.
- [3] M. Nilashi, O. Ibrahim, S. Samad, H. Ahmadi, L. Shahmoradi and E. Akbari, "An analytical method for measuring the Parkinson's disease progression: A case on a Parkinson's telemonitoring dataset", *Measurement*, vol. 136, pp. 545-557, 2019.
- [4] M. Nilashi, O. Ibrahim and A. Ahani, "Accuracy Improvement for Predicting Parkinson's Disease Progression", *Scientific Reports*, vol. 6, no. 1, 2016.
- [5] Scikit-learn developers, "sklearn.model_selection.GroupShuffleSplit", *Scikit-learn: Machine Learning in Python*, 2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupShuffleSplit.html
- [6] J. Magiya, "Pearson Coefficient of Correlation Explained.", *Medium*, 2019. [Online]. Available: <https://towardsdatascience.com/pearson-coefficient-of-correlation-explained-369991d93404>