Informe 1: Aprendizaje de Máquina Implementado en Servidor Remoto. Sensores 1*

Julio Jimmy Cuadros Acosta
juliocuadros230835@correo.itm.edu.co
Instituto Tecnológico Metropolitano

***Abstract*— This work includes the implementation of algorithms for signal quality evaluation based on segmentation and characterization of signals in swallowing patterns. According to the state of the art, the most used and recommended characteristics that allow describing the quality of the signal include average, variance, inclination, kurtosis, and root mean square. Running codes on the local computer have a high energy cost and, in some cases, the processing capacity is not enough. In this report, was evaluated the ability to implement models on a server by comparing their performance, resulting in an efficient outline to follow for project development.**

keywords-Query, Trace.

***Resumen*— Este trabajo incluye la implementación de algoritmos para la evaluación de la calidad de la señal basada en la segmentación y caracterización de las señales en los patrones de deglución. Según el estado del arte, las características más utilizadas y recomendadas que permiten describir la calidad de la señal incluyen promedio, varianza, skewness, curtosis y raíz cuadrática media. Ejecutar códigos en la computadora local tiene un alto costo de energía y, en algunos casos, la capacidad de procesamiento no es suficiente. En este informe, se evaluó la capacidad de implementar modelos en un servidor mediante la comparación de su rendimiento, lo que resultó en un esquema eficiente a seguir para el desarrollo de un proyecto.**

Palabras Clave-Query, Trace.

I. INTRODUCCIÓN

La disfagia es la alteración en el proceso del tragar que dificulta el movimiento del bolo alimenticio de manera segura desde la cavidad oral al estómago, y es una condición potencialmente peligrosa que puede causar infecciones respiratorias a repetición hasta neumonía, aumentado la demanda en recursos de salud.En el ámbito de la salud es frecuente su presencia, pero es uno de los trastornos más descuidados en diagnóstico, tratamiento y seguimiento, requiriendo propuestas de investigación dirigidas a su reconocimiento precoz, diferenciación oportuna de su etiología neurogénica o neuromuscular, pronóstico y evolución en el tiempo[1].

Este informe está en el marco del proyecto: "Diagnóstico y seguimiento de pacientes con disfagia neuromuscular y neurogénica mediante la integración de señales no invasivas y variables clínicas"[2].El alcance de este trabajo incluye el desarrollo de algoritmos para la evaluación de la calidad de la señal y su evaluación a partir de la segmentación y caracterización de señales en patrones deglutorios[3].Según el estado del arte, se evalúa de forma preliminar las características más ampliamente utilizadas y recomendadas que permiten describir la calidad de la señal incluye promedio, varianza, Skewness, Kurtosis y Raíz Cuadrática Media.

El método para el análisis del comportamiento incluye la integración en el esquema de trabajo de las herramientas Structured Query Language (SQL),lenguaje de consulta estructurado que se utiliza para comunicarse con una base de datos[4];SQL Server Management Studio (SSMS),entorno integrado para administrar cualquier infraestructura SQL[5]; Visual Studio Code (VSC), editor de código redefinido y optimizado para crear y depurar aplicaciones web y en la nube[6];Spyder, entorno de desarrollo integrado para programación científica en Python[7]; y finalmente, Microsoft Azure, servicio de computación en la nube para construir, probar, implementar y administrar aplicaciones y servicios a través de centros de datos[8].

Ejecutar códigos en el ordenador local tiene un alto costo energético y para algunos casos la capacidad de procesamiento no es suficiente[9].Esto adicional a la facilidad de acceso global de la información en la nube. Es por esta razón que el objetivo de este informe es entender e implementar el manejo de flujo de la información de una base de datos desde y hacia la nube en un servidor con la finalidad de correr algoritmos de alto consumo energético y requerimiento de procesamiento mediante una comparación puntual de 3 algoritmos de Aprendizaje de Maquina K-vecino más cercano (KNN); Maquina de Soportes Vectorial(SVM) y Naive Bayes(NB).

II. METODOLOGÍA

El desarrollo del proyecto está dividido en dos secciones principales. En primer lugar, se migró la base de datos en formato .mat al lenguaje python, se procesó las señales sEMG obteniendo la matriz de características en un numpy array(Código). Finalmente, esta matriz se convirtió a tipo dataframe, se hizo la conexión al server y fue insertada a una tabla SQL Azure con el nombre de FeaturesTest (Código).

*Facultad de Ciencias Exactas y Aplicadas.

A continuación, se presenta la linea de código que permite realizarlo.

```
Xdf.to_sql('Featurestest', index=False, con=
    engine_azure)
```

En segundo lugar, se usó estos datos en SQL desde el VSC, se implementó los algoritmos de KNN, SVM y NB en la Función Azure desplegada en la nube y finalmente se ejecutó mediante la herramienta postman que permite realizar las peticiones HTTP. Se condicionó a 3 posibles casos, según el valor de variable1 que puede ser menor a 10, 29 y validación al error sucesivamente;en el primer caso se replica con los resultados para comparar el desempeño a cada modelo aplicado; El segundo que entrega la información de la tabla de características con la consulta "SELECT * FROM Featurestest" y el tercero informa que se recibió la petición pero no genera un retorno específico. En el bloque Siguiente se muestra la sección del código que indica las condiciones y la información de la respuesta según la petición HTTP. Se resalta en la linea 6,la instrucción que realiza el traceo al ejecutar el query "INSERT INTO [dbo].[logs]" adjuntándole la información definida de hora, fecha e información de errores y procedimiento.

```
comparacion_df = pd.DataFrame ([comparacion],
    columns=['KNN','SVM','NB'])
diccionario = comparacion_df.to_dict('dict')
json_response = json.dumps(diccionario,indent=2)
diccionario2 = df_datos.to_dict('dict')
json_response2 = json.dumps(diccionario2,indent=2)
traceDB(cnxnAzure,ID,'Enviando respuesta, Funcionó.
    ')

if variable1 < 10:
    return func.HttpResponse(json_response)
elif variable1 == 29:
    return func.HttpResponse(json_response2)
else:
    return func.HttpResponse("LISTO EL POLLO-maybe
    we should writing in English too. by the way,
    the postman flawlessly worked",status_code=200)
```

La clasificación se hizo biclase a partir de la librería de aprendizaje de máquina Scikit-learn, separando señales etiquetadas de mala calidad(valor=1) de señales etiquetadas de buena calidad(valor=0). Debido a que el informe es de carácter preliminar al proyecto en cuestión, no se normalizó la base de datos, no se realizó una optimización de los algoritmos clasificadores.Respecto a la validación, la base de datos se dividió de forma aleatoria en dos grupos, uno que contiene aproximadamente el 70 de los datos, con el cual se realiza el entrenamiento y el 30 restante de los datos se emplea para validar el desempeño del algoritmo al momento de clasificar, teniendo en cuenta el etiquetado que entrega la base de datos.

Se resalta que el KNN estima la densidad de probabilidad de acuerdo a la distancia respecto a una cantidad definida de datos en el entrenamiento y agrupa según las clases definidas[10]. Se tomó una distancia de 3 vecinos con la métrica de Minkowski por defecto, Euclideana. En segundo lugar, La SVM separa las clases lo más amplio posible mediante un hiper-plano definido como el vector de soporte, que es el que permite separar a las dos clases bajo el criterio de la función Kernel, que es una medida de similitud entre dos datos, específicamente esta función retorna el resultado del producto punto entre dos vectores en un espacio de mayor dimensionalidad sin mapear los datos específicamente a ese espacio[11]. Se implementó sobre las características con una función kernel de base radial. En tercer lugar NB se presenta como un algoritmo que en su etapa de entrenamiento busca estimar una función de densidad de probabilidad por cada una de las clases.Durante la etapa de clasificación, este algoritmo evalúa los datos a clasificar en cada una de las funciones de densidad de probabilidad estimadas y finalmente el dato es asignado a la clase en donde su probabilidad de pertenecer a esta es mayor[12].

Para finalizar esta sección metodológica se adiciona la lista de videos requeridos en el reporte:

1. Evidencia de funcionamiento preliminar en la nube, tener en cuenta que no se realizó video del resultado final debido a falta de crédito Azure, no obstante si fue completamente desarrollado y verificado por el docente en cargo.
2. Migración de la matriz de características al servidor remoto.
3. Explicación de Función Azure definitiva desplegada.
4. Control de versión y subida en Repositorio Git-Hub.

III. RESULTADOS Y DISCUSIÓN

A. *Insertar tabla en SQL Azure.*

En esta sección se presenta el resultado de subir a la nube la matriz de características.

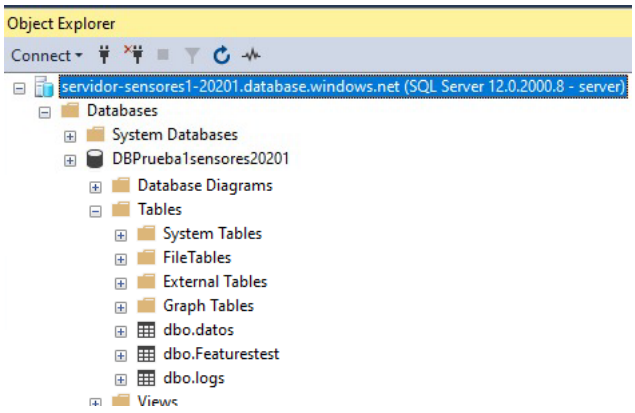

Fig. 1. Reporte de tabla en la Nube

Se resalta la tabla Featurestest a la cual se le hizo el Query request, y la tabla logs que incluye la información del traceo.

B. *Punto 2 del Taller.*

En esta sección se presenta las gráficas obtenidas a partir del reporte de los 3 modelos.

KNN

	precision	recall	f1-score	support
0	0.64	0.88	0.74	8
1	0.75	0.43	0.55	7
accuracy			0.67	15
macro avg	0.69	0.65	0.64	15
weighted avg	0.69	0.67	0.65	15

Fig. 2. Reporte de clasificación KNN

SVM

	precision	recall	f1-score	support
0	0.00	0.00	0.00	8
1	0.47	1.00	0.64	7
accuracy			0.47	15
macro avg	0.23	0.50	0.32	15
weighted avg	0.22	0.47	0.30	15

Fig. 3. Reporte de clasificación SVM

NB

	precision	recall	f1-score	support
0	0.67	1.00	0.80	8
1	1.00	0.43	0.60	7
accuracy			0.73	15
macro avg	0.83	0.71	0.70	15
weighted avg	0.82	0.73	0.71	15

Fig. 4. Reporte de clasificación NB

El parámetro precision indica que en el caso del NB clasificó correctamente todas las señales malas. por otro lado el SVM presentó el menor rendimiento clasificando correctamente solo 47% de esa clase. Para la clase de señales buenas, también obtuvo el rendimiento más bajo con un valor de cero, mientras que en este caso el KNN fue el más alto porque correctamente clasificó el 75% de las señales buenas como buenas. Como valor de recall que indica el porcentaje de clasificación correcta de verdaderos positivos y falsos negativos, sin embargo el SVM aquí tuvo mayor desempeño entre los modelos el valor máximo al identificar correctamente la clase de señal de mala calidad pero también cuenta con el menor valor en cuanto a la clase de buena calidad. Estos anteriores valores son apreciados más claramente con el F1-score donde se afianza la tendencia de los anteriores parámetros. En general NB tiene el mejor rendimiento con valores más cercanos a 1, luego, el KNN y finalmente la SVM tiene el peor rendimiento. Esto es observado en que esta tiene la menor accuracy que es una medida globlal de rendimiento según la librería sklearn, con valores de 0.73;0.67 y 47 sucesivamente. NB con accuracy de 73% indica que se debe hacer un ajuste. porque es preciso el clasificador al separar clases correctamente, mas no es tan robusto es porque pierde un número significativo de clases a clasificar. De igual forma, la optimización debe ser replicada a todos los modelos y se esperaría una mejora en el desempeño de todos los modelos. Analizando estos valores se sugiere que puede ser debido a que la SVM suele ofrecer mayor rendimiento a una cantidad mayor de características, como solo hay 5. Lo esperado es efectivamente un bajo rendimiento,por este motivo a una mayor cantidad se esperaría una mejora sustancial de este modelo, finalmente NB debido a la suposición de independencia condicional de las clases genera el mayor desempeño, esto se sugiere puede ser debido a que las características seleccionadas le permiten brindar una correcta separabilidad de las clases por su descripción acorde del tipo de señal.

En ultimo lugar es menester resaltar la importancia de aumentar la cantidad de características, y los modos de evaluación del desempeño, incluyendo tests más robustos que indiquen la diferencia significativa entre los modelos como el test ANNOVA y métodos gráficos tal como los boxplot.

IV. CONCLUSIÓN

Basado en la discusión de los resultados y el objetivo planteado, se confirma el entendimiento del flujo de la información desde y hacia la nube permitiendo implementar un análisis de modelos ejecutados totalmente sobre una base de datos alojada en un Server remoto proporcionado por la licencia de Azure. De esta manera es se realizaron procesos sin gasto local energético y se aplicó nuevas competencias adquiridas durante el curso de sensores 1 sobre un proyecto con altos requerimientos de computo ahorrando gasto energético local.En el cual Se logra un mayor rendimiento del modelo NB con Accuracy de 73%. Como trabajo futuro es menester realizar una optimización de los parámetros para los 3 modelos de clasificación KNN, SVM y NB tal como la evaluación de los desempeños de forma más robusta con métodos como ANNOVA y Boxplot. Así como finalmente se propone aumentar tanto la base de datos como el numero de características aplicadas sobre la base de datos sEMG.

REFERENCES

[1] J. C. S. Escudero, "Diagnóstico y seguimiento de pacientes con disfagia neuromuscular y neurogénica mediante la integración de señales no invasivas y variables clínicas," pp. 1–48, 2017.

[2] Instituto Tecnológico Metropolitano and Universidad Pontificia Bolivariana, "Diagnóstico y seguimiento de pacientes con disfagia neuromuscular y neurogénica mediante la integración de señales no invasivas y variables clínicas.," 2017.

[3] A. F. Orozco Duque and J. Cuadros Acosta, " Plan de formación Jóven investigador e Innovador ITM-2020," tech. rep., Instituto Tecnológico Metropolitano, 2020.

[4] IT Business Edge, "SQLCourse - Lesson 1: What is SQL?," 2015.

[5] Microsoft, "SQL Server Management Studio (SSMS)," 2018.

[6] Microsoft, "Visual Studio Code - Code Editing. Redefined," 2016.

[7] Python Software Foundation, "What is Python? Executive Summary— Python.org.," 2017.

[8] Microsoft Azure, "¿Qué es Azure? El mejor servicio en la nube de Microsoft — Microsoft Azure," 2016.

[9] L. J. Stockmeyer, "Computational Complexity," jan 1992.

[10] Z. Zhang, "Introduction to machine learning: k-nearest neighbors," *Ann. Transl. Med.*, vol. 4, no. 11, pp. 218–218, 2016.

[11] Scikit-Learn, "1.4. Support Vector Machines — scikit-learn 0.22.2 documentation," 2019.

[12] Scikit-Learn, "1.9. Naive Bayes — scikit-learn 0.22.2 documentation," 2019.