Tecnológico de Monterrey

Campus Guadalajara

Análisis de datos para aplicaciones médicas

Por

Jorge Eduardo Guijarro Márquez | A01563113

Frado García Palacios | A01352112

Claudio José González Arriaga | A00232276

Modelación del aprendizaje con Inteligencia Artificial

Prof. Omar Mendoza Montoya

10 de Junio del 2024

1. **Descripción de la aplicación implementada junto con los datos recolectados**

   La aplicación consiste en un sistema clasificador en línea por reconocimiento de voz, el cual, se espera que sea capaz de constantemente recolectar datos y conforme se haga la recolección de datos, esos mismos sean clasificados.

   Los usuarios pueden activar el micrófono y grabar su voz, posteriormente la aplicación convierte esos datos en una matriz de características y en base en el modelo de clasificación con el que cuenta la aplicación, la aplicación es capaz de clasificar cada una de las observaciones en las clasificaciones preestablecidas.

   Actualmente, existen algunos softwares que utilizan el reconocimiento de voz y modelos de clasificación para realizar tareas específicas. Un ejemplo de ello es Invox Medical, un software de IA para el dictado y transcripción en tiempo real de informes médicos, que permite que los usuarios empiecen a dictar y el audio se transcribe a texto de manera automática donde haya situado el cursor. De igual manera, es posible realizar otro tipo de acciones cómo desplazarse en la pantalla, corregir y cargar plantillas mediante comandos de voz.

   La aplicación trabaja con audio, que posteriormente es convertido en matrices de características para ser trabajada.

2. **Descripción de las características extraídas de los datos.**

   Para poder realizar la clasificación en línea, primero se necesitó de la recolección de datos para entrenar el propio modelo de clasificación. Los datos se conforman de 540 observaciones de las palabras *perro, gato, jirafa, tortuga, avestruz, elefante*. La composición de los datos es de las siguiente manera:

   - Perro: 90 palabras
     - 30 palabras pronunciadas por Jorge
     - 30 palabras pronunciadas por Claudio
     - 30 palabras pronunciadas por Frado
   - Gato: 90 palabras
     - 30 palabras pronunciadas por Jorge
     - 30 palabras pronunciadas por Claudio
     - 30 palabras pronunciadas por Frado
   - Jirafa: 90 palabras
     - 30 palabras pronunciadas por Jorge
     - 30 palabras pronunciadas por Claudio
     - 30 palabras pronunciadas por Frado
   - Tortuga: 90 palabras
     - 30 palabras pronunciadas por Jorge
     - 30 palabras pronunciadas por Claudio

        - 30 palabras pronunciadas por Frado
    - Avestruz: 90 palabras
        - 30 palabras pronunciadas por Jorge
        - 30 palabras pronunciadas por Claudio
        - 30 palabras pronunciadas por Frado
    - Elefante: 90 palabras
        - 30 palabras pronunciadas por Jorge
        - 30 palabras pronunciadas por Claudio
        - 30 palabras pronunciadas por Frado

Después de la recolección de audio, siguió el procesamiento del mismo. Para esto se utilizó la técnica de extracción de características *Mel-frequency cepstral coefficients (MFCC),* una técnica muy usada en el procesamiento de audio y voz.

MFCC crea conjuntos de coeficientes que capturan la forma del espectro de la potencia de una señal de sonido.

Una de las principales razones para usar MFCC es que es particularmente útil a la hora de capturar características importantes para la percepción del habla humana.

Al final, la parte de procesamiento arrojó una matriz donde la primera columna correspondía a la asignación de las clases 1 - 6 (correspondiente a cada animal) mientras que las demás columnas son justamente las características obtenidas por medio de la técnica MFCC.

Se obtuvo una matriz de 540 observaciones (filas) y 5186 características (columnas) más otra columna para las clases (columna 0).

3. Resultados obtenidos en la evaluación de los clasificadores probados, y optimización de modelos

Previo a una obtención de hiperparámetros y selección de características, se probaron 10 modelos de clasificación distintos, entre los cuales se incluyen: SVC kernel lineal, SVC kernel radial, KNN y otros modelos como Random Forest y Adaboost, entre los cuales, el clasificador que obtuvo mejores resultados fue el SVM con Kernel Lineal con un accuracy 0.91 en la mayoría de los casos.
También fue notable la diferencia de algunos modelos que demostraron un mayor rendimiento en términos de velocidad de ejecución en comparación con otros como SVC Lineal con un tiempo de 3 seg aproximadamente, contra un modelo AdaBoost que tardaba aproximadamente 2 min.

SVC LINEAL

	precision	recall	f1-score	support
1.0	0.88	0.88	0.88	90
2.0	0.83	0.81	0.82	90
3.0	0.95	0.96	0.95	90
4.0	0.90	0.92	0.91	90
5.0	0.99	0.99	0.99	90
6.0	0.92	0.91	0.92	90
accuracy			0.91	540
macro avg	0.91	0.91	0.91	540
weighted avg	0.91	0.91	0.91	540

SVM BASE RADIAL

	precision	recall	f1-score	support
1.0	0.70	0.81	0.75	90
2.0	0.74	0.63	0.68	90
3.0	0.91	0.91	0.91	90
4.0	0.90	0.92	0.91	90
5.0	0.99	0.92	0.95	90
6.0	0.84	0.87	0.85	90
accuracy			0.84	540
macro avg	0.85	0.84	0.84	540
weighted avg	0.85	0.84	0.84	540

LDA

	precision	recall	f1-score	support
1.0	0.82	0.84	0.83	90
2.0	0.73	0.81	0.77	90
3.0	0.97	0.92	0.94	90
4.0	0.91	0.90	0.91	90
5.0	0.99	0.96	0.97	90
6.0	0.93	0.88	0.90	90
accuracy			0.89	540
macro avg	0.89	0.89	0.89	540
weighted avg	0.89	0.89	0.89	540

KNN

	precision	recall	f1-score	support
1.0	0.79	0.90	0.84	90
2.0	0.83	0.71	0.77	90
3.0	0.88	0.91	0.90	90
4.0	0.88	0.90	0.89	90
5.0	0.99	0.97	0.98	90
6.0	0.87	0.84	0.86	90
accuracy			0.87	540
macro avg	0.87	0.87	0.87	540
weighted avg	0.87	0.87	0.87	540

RED PERCEPTRÓN MULTICAPA

	precision	recall	f1-score	support
1.0	0.17	0.20	0.18	90
2.0	0.17	0.31	0.22	90
3.0	0.24	0.36	0.28	90
4.0	0.84	0.18	0.29	90
5.0	0.00	0.00	0.00	90
6.0	0.17	0.21	0.19	90
accuracy			0.21	540
macro avg	0.26	0.21	0.19	540
weighted avg	0.26	0.21	0.19	540

DECISION TREE CLASSIFIER

	precision	recall	f1-score	support
1.0	0.43	0.46	0.44	90
2.0	0.46	0.48	0.47	90
3.0	0.75	0.71	0.73	90
4.0	0.59	0.60	0.60	90
5.0	0.74	0.66	0.69	90
6.0	0.52	0.54	0.53	90
accuracy			0.57	540
macro avg	0.58	0.57	0.58	540
weighted avg	0.58	0.57	0.58	540

LOGISTIC REGRESSION CLASSIFIER

	precision	recall	f1-score	support
1.0	0.87	0.82	0.85	90
2.0	0.75	0.78	0.77	90
3.0	0.92	0.93	0.93	90
4.0	0.89	0.90	0.90	90
5.0	0.98	0.97	0.97	90
6.0	0.89	0.90	0.90	90
accuracy			0.88	540
macro avg	0.88	0.88	0.88	540
weighted avg	0.88	0.88	0.88	540

RANDOM FOREST CLASSIFIER

	precision	recall	f1-score	support
1.0	0.79	0.88	0.83	90
2.0	0.84	0.77	0.80	90
3.0	0.99	0.91	0.95	90
4.0	0.84	0.90	0.87	90
5.0	0.98	0.96	0.97	90
6.0	0.88	0.88	0.88	90
accuracy			0.88	540
macro avg	0.88	0.88	0.88	540
weighted avg	0.88	0.88	0.88	540

SVM KERNEL POLINÓMICO

	precision	recall	f1-score	support
1.0	0.83	0.87	0.85	90
2.0	0.80	0.80	0.80	90
3.0	0.96	0.90	0.93	90
4.0	0.91	0.89	0.90	90
5.0	0.99	0.90	0.94	90
6.0	0.82	0.93	0.88	90
accuracy			0.88	540
macro avg	0.89	0.88	0.88	540
weighted avg	0.89	0.88	0.88	540

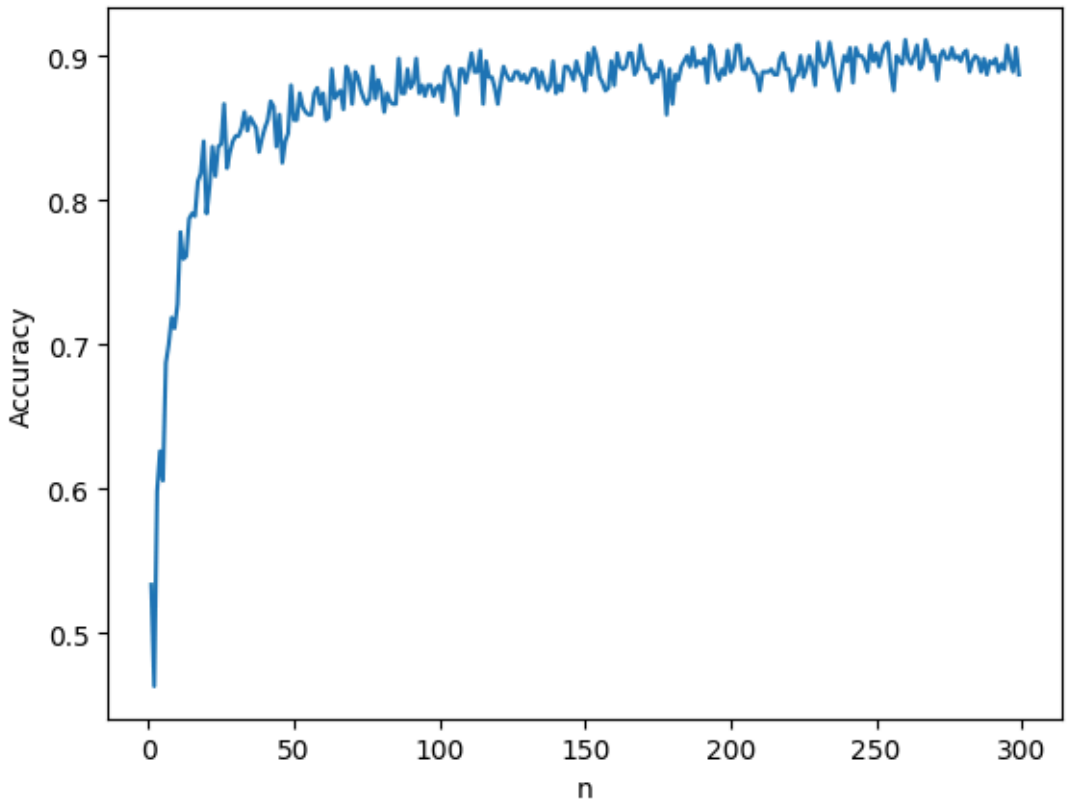
ADABOOST CLASSIFIER

	precision	recall	f1-score	support
1.0	0.32	0.47	0.38	90
2.0	0.26	0.16	0.19	90
3.0	0.39	0.57	0.46	90
4.0	0.29	0.29	0.29	90
5.0	1.00	0.71	0.83	90
6.0	0.40	0.31	0.35	90
accuracy			0.42	540
macro avg	0.44	0.42	0.42	540
weighted avg	0.44	0.42	0.42	540

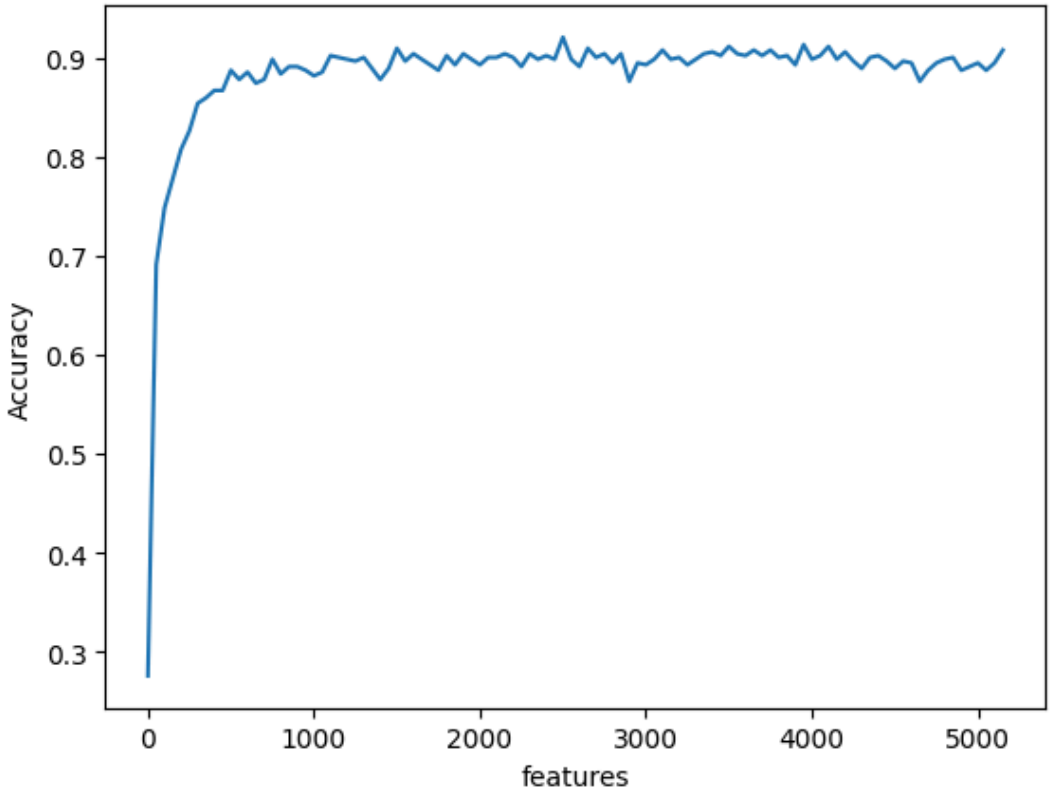
Para realizar una obtención de hiperparámetros y selección de características seleccionamos los modelos Random Forest y KNN:
Para la selección de hiperparámetros se probó una serie de posibles valores para cada hiperparámetro donde por cada valor de estos, se evaluó el modelo usando validación cruzada.
Para la selección de características, se utilizó el método Filter debido a que su complejidad computacional no es tan alta y toma considerablemente menos tiempo que otros métodos.

- Modelo de clasificación Random Forest:

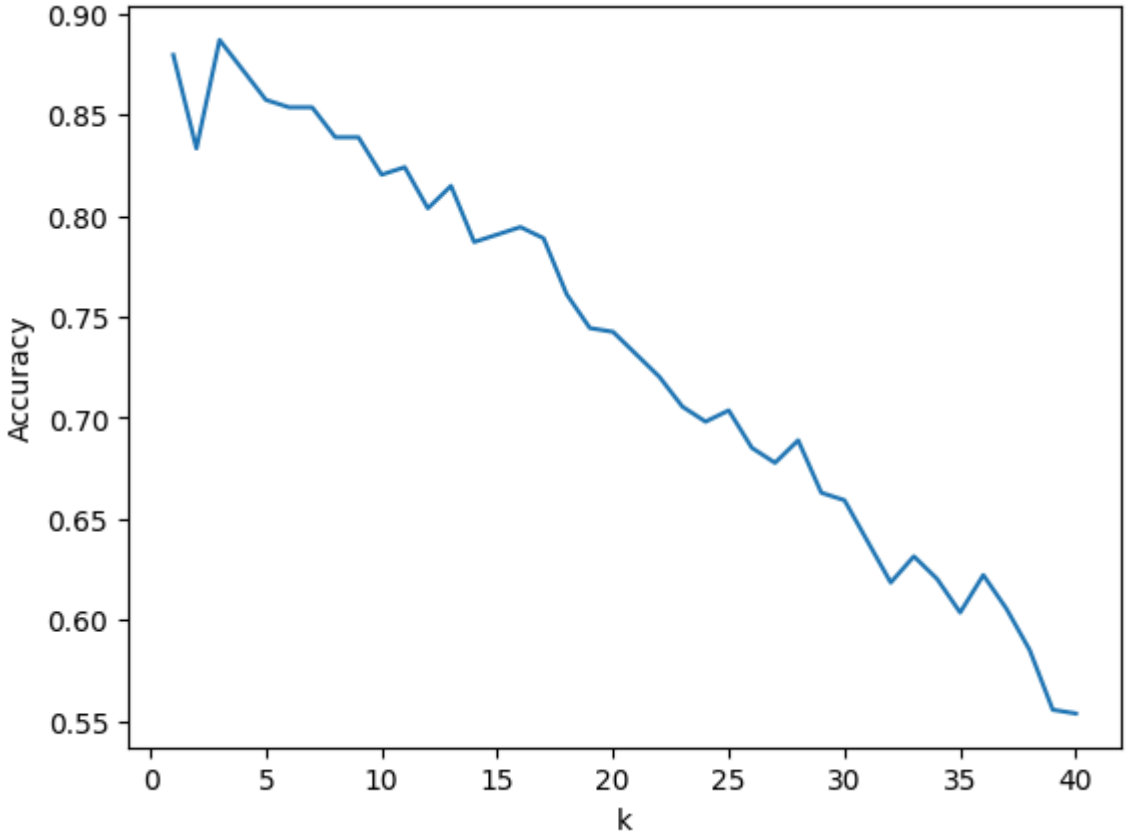
- Hiperparámetro a optimizar: Número de árboles (*n_estimators*)
- Número óptimo de árboles: 260



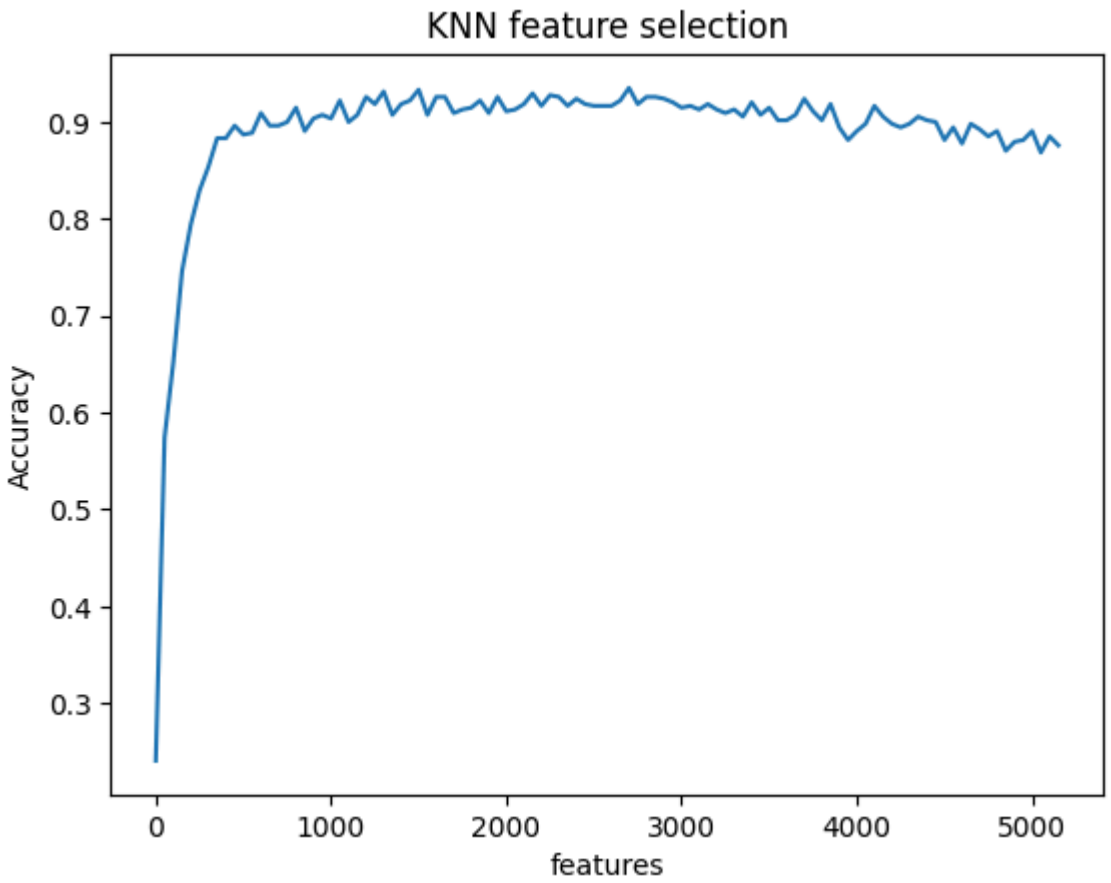
- Número óptimo de características: 2501
- Exactitud alcanzada con 2501 características: 0.9204



- Modelo de clasificación KNN
    - Hiperparámetro a optimizar: Número de vecinos más cercanos (*n_neighbors*)
    - Número óptimo de vecinos más cercanos: 3



    - Número óptimo de características: 2701
    - Exactitud alcanzada con 2501 características: 0.9352



4. **Resultados de la aplicación en línea y respuesta a las siguientes preguntas:**
    - **¿Funciona igual con todos los miembros del equipo?**
    - **¿El rendimiento de la aplicación en línea es el esperado de acuerdo a los resultados de validación cruzada?**

    No, la aplicación no funciona igual con todos los miembros del equipo. Aunque las grabaciones se realizaron en una misma computadora y los resultados fueron consistentes para todos los miembros del equipo en ese entorno, al intentar hacer las pruebas en una computadora diferente, específicamente en una computadora que tiene MacOS, el formato de la grabación resultante fue diferente. Esta discrepancia en el formato de grabación resultó en resultados inconsistentes al procesar el audio.

    El rendimiento de la aplicación en línea es mayormente el esperado de acuerdo a los resultados de validación cruzada, pero con algunas excepciones. En casi todas las pruebas realizadas en la misma computadora donde se entrenó y validó el modelo, los resultados fueron congruentes con la validación del modelo. Esto indica que el modelo funciona correctamente en condiciones controladas y con el mismo hardware y configuración de grabación.

Sin embargo, al realizar pruebas en mi computadora personal (Mac), el formato de grabación difería, lo que afectaba el procesamiento del audio y, consecuentemente, los resultados de clasificación del modelo. Esta diferencia sugiere que el modelo y la aplicación están altamente dependientes del formato y la configuración del hardware de grabación.

5. Conclusiones individuales}

Conclusión Jorge:

El desarrollo de un aplicación de análisis de datos para aplicaciones médicas mediante el reconocimiento de voz ha sido un proceso revelador. Durante el desarrollo de este proyecto además de poner a prueba una gran cantidad de modelos de clasificación, también tuvimos la oportunidad de participar en el proceso de recolección y procesamiento de los datos, algo que previamente no habíamos realizado. Sin embargo,en el transcurso del mismo, nos enfrentamos a distintos desafíos, como lograr que el clasificador seleccionado para la aplicación, fuera lo más efectivo posible, pues fue necesario un proceso bastante largo y computacionalmente lento, pero a pesar de ello, notamos lo efectivos que algunos métodos como *Mel-frequency cepstral coefficients (MFCC)* pueden ser al momento de trabajar con señales de audio y voz.Además tenemos que tomar en cuenta que los datos de voz son inevitablemente variables debido a factores externos como el acento de cada persona, la pronunciación y el ruido, por lo que fue de suma importancia tratar de controlar cada una de estas variables para lograr que el clasificador pudiera tener resultados efectivos y similares para cada miembro del equipo.A pesar de estos desafíos, el desarrollo de un sistema de este tipo representa un gran avance del aprendizaje supervisado dentro del campo médico.

Conclusión Claudio:

Este proyecto fue un reto significativo, especialmente en la parte de la captura de audio debido a diferencias en los sistemas operativos. Como uso macOS, tuve que investigar cómo otorgar permisos a las aplicaciones para usar los micrófonos. Una vez resuelto este problema, descubrí que el formato de grabación en mi computadora era diferente al de mis compañeros. Esto nos llevó a grabar todos los audios en la misma computadora para mantener la consistencia.

El entrenamiento del modelo se realizó con datos grabados en otra computadora, y cuando intenté montar el modelo en mi Mac, las clasificaciones eran incorrectas. Esto nos llevó a realizar varios cambios en el código y la implementación. Finalmente, al probar el código en la computadora original, la clasificación funcionó perfectamente,

esto fue especialmente valioso ya que notamos de primera mano la importancia de la consistencia en el hardware y el software utilizados.

Este proyecto destacó la importancia de desarrollar aplicaciones multiplataforma y nos permitió evaluar diferentes modelos de clasificación de audio, comprendiendo por qué algunos funcionan mejor en ciertas situaciones. Fue un muy buen proyecto ya que fue un reto en el cual nos encontramos con diversos desafíos técnicos así como de implementación, lo cual nos llevó a soluciones diferentes en las cuales fue vital la colaboración entre los miembros del equipo.

Conclusión Frado:

Debo decir que aunque el proyecto fue bastante interesante y hasta entretenido, no se exentó de problemas tediosos que limitaron la capacidad para evaluar el modelo o reinventar mecánicas y hacer el clasificador más interesante. Una de las limitaciones más evidentes fue el tiempo de ejecución del código donde había caso en los que si uno quería ser muy preciso con las evaluaciones o selección de de características, el tiempo ascendía a varias horas. Esto se solucionó simplificando los métodos y tomando ciertos “atajos” para que el tiempo de ejecución fuera menor. Por ejemplo, el equipo quería usar el método de selección de características Filter-Wrapper ya que fue el que mejores resultados arrojó en una actividad pasada; por desgracia este método, de ser aplicado, hubiese tardado varias horas para ejecutarse por completo; es por esto que se decidió usar el método Filter.

Me hubiera gustado poder agregar algunas otras características a la parte de la clasificación en línea como una interfaz o la existencia de una séptima clase donde entraran todas aquellas palabras que no fueran cualquiera de las 6, sin embargo, justamente el tiempo fue nuestro peor enemigo para la realización de este proyecto.

Al final, el modelo resultó ser bastante exacto tanto en las métricas de evaluación como en la clasificación en línea

Referencias

INVOX Medical: la solución definitiva para dictar informes médicos. (2021, abril 17). INVOX Medical. https://invoxmedical.com/

Kiran, U. (2021, June 13). *MFCC technique for speech recognition*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/mfcc-technique-for-speech-recognition/

Máquinas de Vector Soporte (SVM) con Python. (s. f.). https://cienciadedatos.net/documentos/py24-svm-python

Allaire, A., Alexis, A., & van den Oord, J. (2019). Streamlit: Democratizing Data Science. [GitHub Repository]. Recuperado de https://github.com/streamlit/streamlit

Github. (s. f.). *VSCode terminal doesn't allow/request permissions to access media devices · Issue #95062 · microsoft/vscode*. GitHub. https://github.com/microsoft/vscode/issues/95062#issuecomment-614044993