

APP DE VISUALIZACIÓN DE BASES DE DATOS

MEMORIA TÉCNICA

Por Samuel José Bulnes Caicoya, María Pérez Caballero, David Tzafrir, Hugo Pardo Vellisca



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

ÍNDICE

1- Introducción: Justificación, Aspectos generales, Objetivo, Funcionalidad e Información complementaria. (pág.3)

2- Materiales. (pág.5)

2.1- Datos. (pág.8)

2.2- Preproceso anterior a experimentos. (pág.8)

3- Métodos. (pág.10)

3.1- Metodologías de desarrollo de clasificación. (pág.13)

3.3- Experimentación: en forma de tablas y explicaciones. (pág.18)

3.4- Evaluación. (pág.20)

4- Resultados. (pág.21)

5- Discusión. (pág.22)

6- Conclusiones. (pág.23)

7- Referencias. (pág.24)

I-INTRODUCCIÓN

Justificación

El análisis y la visualización de datos médicos, especialmente aquellos relacionados con imágenes cerebrales, son de suma importancia en el campo de la medicina y la investigación biomédica. Estos datos pueden proporcionar información crucial para comprender enfermedades, diagnosticar trastornos y desarrollar tratamientos efectivos. Sin embargo, el análisis de conjuntos de datos complejos puede ser desafiante debido a la gran cantidad de datos y la necesidad de herramientas especializadas. Por lo tanto, la creación de un sistema de software que facilite este proceso puede tener un impacto significativo en la comunidad médica y científica.

Objetivo Principal

Desarrollo de una aplicación de escritorio para el análisis y visualización de datos clínicos, utilizando Pycharm como programa de desarrollo y Python como lenguaje de programación.

Aspectos Generales

El sistema propuesto en esta memoria técnica tiene como objetivo abordar los desafíos asociados con el análisis y la visualización de datos médicos, centrándose en datos relacionados con imágenes cerebrales. El sistema proporcionará una interfaz intuitiva y herramientas avanzadas para cargar, filtrar, visualizar y analizar estos datos de manera eficiente.

Funcionalidades de la Aplicación

La carga de datos mediante archivos CSV. En cuanto a los filtros de datos se realizan por medio de parámetros como rangos de edad, sexo, variables de interés en función del archivo subido y calidad. La visualización de estos datos es posible gracias a un intuitivo sistema de selección de tipo de gráfico (histograma, bar chart, line chart, scatter). Su posterior análisis y regresiones se lleva a cabo por medio de modelos de regresión como polinomial, lineal, robust (Huber y RANSAC). Existe la función de exportar gráficos y datos interesantes con las funciones de **Export** (graph/outliers/filtered_data). Por último, los datos anómalos pueden ser eliminados para la obtención de mejores resultados.

Desarrollo programa:

El programa se desarrolla en Pycharm, un entorno de desarrollo integrado, muy popular para Python, que ofrece funcionalidades como las antes comentadas, además de control de versiones y depuración.

Librerías utilizadas:

Tkinter → IGU.

Pandas → Carga y manipulación datos.

Matplotlib → Creación gráficos.

Seaborn → Visualización gráficos.

Scikit-learn → Modelos regresión y métricas.

Statsmodels → Métodos regresión avanzada y análisis estadístico.

Numpy → Operaciones matemáticas.

Re → Validar formatos rangos de entrada.

2-MATERIALES

Para contextualizar la puesta en práctica del proyecto, procedemos a describir con exactitud el formato de la base de datos utilizada, de forma que, para posteriores utilizaciones de la aplicación, su compatibilidad para la ejecución sea fácil y sencilla. La base de datos utilizada en este proyecto contiene una amplia gama de variables. A continuación, se presenta un resumen de todas las variables incluidas:

Patient ID	Sex	Age	Report Date	Scale factor	SNR	mSNR	QC	Tissue WM cm	Tissue WM %	Tissue GM cm
job458067	Male	64	20-mar-23	0.75465	211.319	221.612	0.95591	4.755.257	340.955	6.972.817

Ilustración 1. Fragmento de la Base de Datos empleada

Patient ID: Identificador único del paciente.

Sex: Sexo del paciente (Masculino, Femenino, Desconocido).

Age: Edad del paciente en años.

Report Date: Fecha de emisión del informe clínico.

Scale factor: Factor de escala utilizado en las mediciones.

SNR: Relación señal-ruido.

mSNR: Relación señal-ruido modificada.

QC: Control de calidad de los datos.

Tissue WM cm3: Volumen de tejido blanco en centímetros cúbicos.

Tissue WM %: Porcentaje de tejido blanco.

Tissue GM cm3: Volumen de tejido gris en centímetros cúbicos.

Tissue GM %: Porcentaje de tejido gris.

Tissue CSF cm3: Volumen de líquido cefalorraquídeo en centímetros cúbicos.

Tissue CSF %: Porcentaje de líquido cefalorraquídeo.

Tissue Brain cm3: Volumen total del cerebro en centímetros cúbicos.

Tissue Brain %: Porcentaje del volumen total del cerebro.

Tissue IC cm3: Volumen de estructuras intracraneales en centímetros cúbicos.

Tissue IC %: Porcentaje de estructuras intracraneales.

Cerebrum Total cm3: Volumen total del cerebro en centímetros cúbicos.

Cerebrum Total %: Porcentaje del volumen total del cerebro.

Cerebrum T GM cm3: Volumen total de la materia gris del cerebro.

Cerebrum T GM %: Porcentaje del volumen total de la materia gris del cerebro.

Cerebrum T WM cm3: Volumen total de la materia blanca del cerebro.

Cerebrum T WM %: Porcentaje del volumen total de la materia blanca del cerebro.

Cerebrum Right cm3: Volumen del hemisferio derecho del cerebro en centímetros cúbicos.

Cerebrum Right %: Porcentaje del volumen del hemisferio derecho del cerebro.

Cerebrum R GM cm3: Volumen de la materia gris del hemisferio derecho del cerebro.

Cerebrum R GM %: Porcentaje del volumen de la materia gris del hemisferio derecho del cerebro.

Cerebrum R WM cm3: Volumen de la materia blanca del hemisferio derecho del cerebro.

Cerebrum R WM %: Porcentaje del volumen de la materia blanca del hemisferio derecho del cerebro.

Cerebrum Left cm3: Volumen del hemisferio izquierdo del cerebro en centímetros cúbicos.

Cerebrum Left %: Porcentaje del volumen del hemisferio izquierdo del cerebro.

Cerebrum L GM cm3: Volumen de la materia gris del hemisferio izquierdo del cerebro.

Cerebrum L GM %: Porcentaje del volumen de la materia gris del hemisferio izquierdo del cerebro.

Cerebrum L WM cm3: Volumen de la materia blanca del hemisferio izquierdo del cerebro.

Cerebrum L WM %: Porcentaje del volumen de la materia blanca del hemisferio izquierdo del cerebro.

Cerebrum Assymetry: Asimetría entre los hemisferios cerebrales.

Cerebelum Total cm3: Volumen total del cerebelo en centímetros cúbicos.

Cerebelum Total %: Porcentaje del volumen total del cerebelo.

Cerebelum T GM cm3: Volumen total de la materia gris del cerebelo.

Cerebelum T GM %: Porcentaje del volumen total de la materia gris del cerebelo.

Cerebelum T WM cm3: Volumen total de la materia blanca del cerebelo.

Cerebelum T WM %: Porcentaje del volumen total de la materia blanca del cerebelo.

Cerebelum Right cm3: Volumen del hemisferio derecho del cerebelo en centímetros cúbicos.

Cerebelum Right %: Porcentaje del volumen del hemisferio derecho del cerebelo.

Cerebelum R GM cm3: Volumen de la materia gris del hemisferio derecho del cerebelo.

Cerebelum R GM %: Porcentaje del volumen de la materia gris del hemisferio derecho del cerebelo.

Cerebelum R WM cm3: Volumen de la materia blanca del hemisferio derecho del cerebelo.

Cerebelum R WM %: Porcentaje del volumen de la materia blanca del hemisferio derecho del cerebelo.

Cerebelum Left cm3: Volumen del hemisferio izquierdo del cerebelo en centímetros cúbicos.

Cerebelum Left %: Porcentaje del volumen del hemisferio izquierdo del cerebelo.

Cerebelum L GM cm3: Volumen de la materia gris del hemisferio izquierdo del cerebelo.

Cerebelum L GM %: Porcentaje del volumen de la materia gris del hemisferio izquierdo del cerebelo.

Cerebelum L WM cm3: Volumen de la materia blanca del hemisferio izquierdo del cerebelo.

Cerebelum L WM %: Porcentaje del volumen de la materia blanca del hemisferio izquierdo del cerebelo.

Cerebelum Assymetry: Asimetría entre los hemisferios del cerebelo.

Brainstem cm3: Volumen del tronco encefálico en centímetros cúbicos.

Brainstem %: Porcentaje del volumen del tronco encefálico.

Lateral ventricles Total cm3: Volumen total de los ventrículos laterales en centímetros

2.1-DATOS

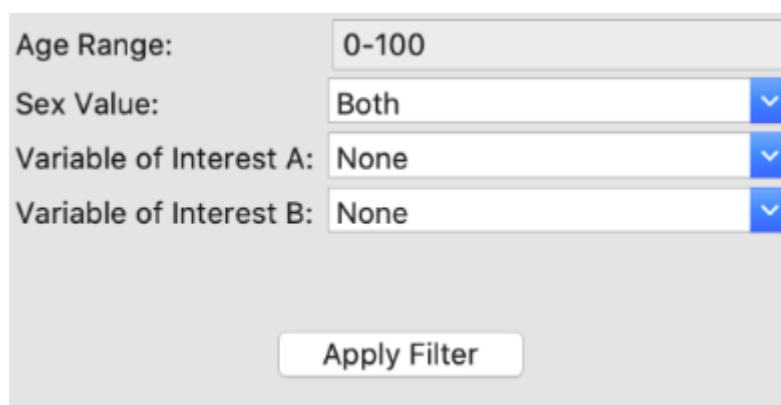
La base de datos proporcionada contiene información biomédica detallada de pacientes, estructurada en un formato tabular. Cada fila representa un paciente único y cada columna corresponde a una característica específica. Entre las características incluidas se encuentran datos demográficos (sexo, edad), medidas de tejido cerebral (volumen de tejido gris, volumen de tejido blanco), así como información detallada sobre otras variables clínicas relevantes.

Los datos abarcan una amplia variedad de pacientes, incluyendo tanto casos de control como pacientes con diversas condiciones neurológicas. Esta diversidad en la población de pacientes proporciona un conjunto de datos robusto y completo para el análisis y la construcción de modelos predictivos.

2.2.-PREPROCESO ANTERIOR A EXPERIMENTOS

Antes de realizar los experimentos de clasificación, se llevó a cabo un preproceso de los datos para garantizar su calidad y prepararlos adecuadamente para su análisis, seleccionando las características más relevantes para la tarea de clasificación.

En nuestra interfaz, hemos desarrollado una intuitiva selección de filtros para especificar cuáles son los datos que se muestran en las gráficas que seleccionemos. Para ello, hemos incluido el rango de edad (en formato int-int), selección de sexo (en el cual se incluye M vs F y both, para su comparación y agregación respectivamente) y la selección de las dos variables de interés a medir para ser comparadas, que se adaptan al fichero que seleccionemos. Por último, también podremos seleccionar un rango de calidad en formato float (0.1-1).

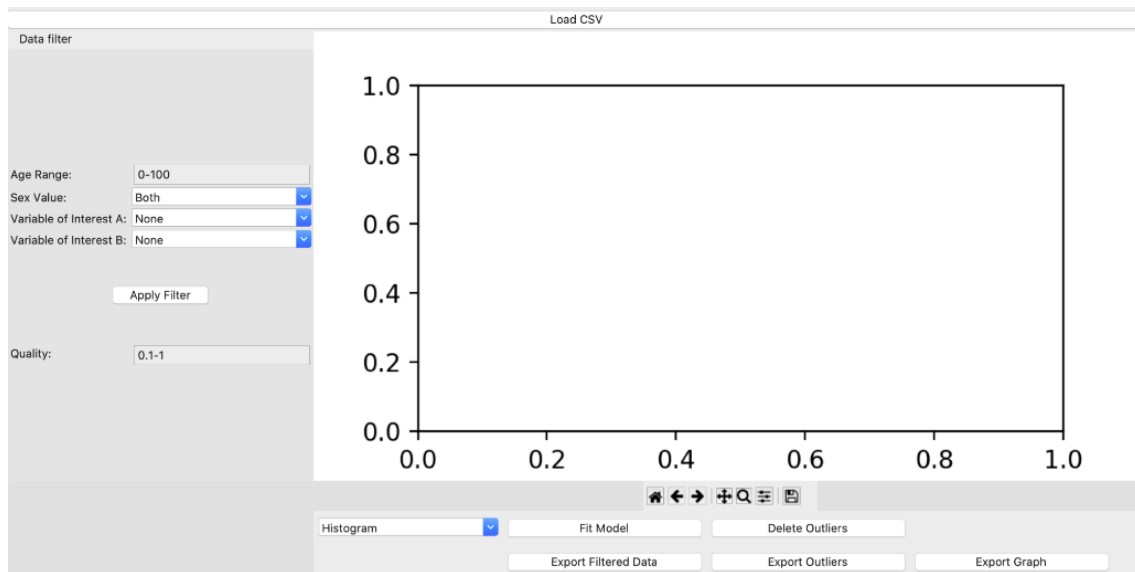


The image shows a user interface for applying filters. It consists of four rows of input fields, each with a label on the left and a text input or dropdown on the right. The first row is 'Age Range' with a text input containing '0-100'. The second row is 'Sex Value' with a dropdown menu showing 'Both'. The third row is 'Variable of Interest A' with a dropdown menu showing 'None'. The fourth row is 'Variable of Interest B' with a dropdown menu showing 'None'. Each dropdown menu has a blue arrow icon on the right. Below these fields is a button labeled 'Apply Filter'.

Age Range:	0-100
Sex Value:	Both
Variable of Interest A:	None
Variable of Interest B:	None

Apply Filter

La interfaz propuesta tiene el siguiente aspecto:

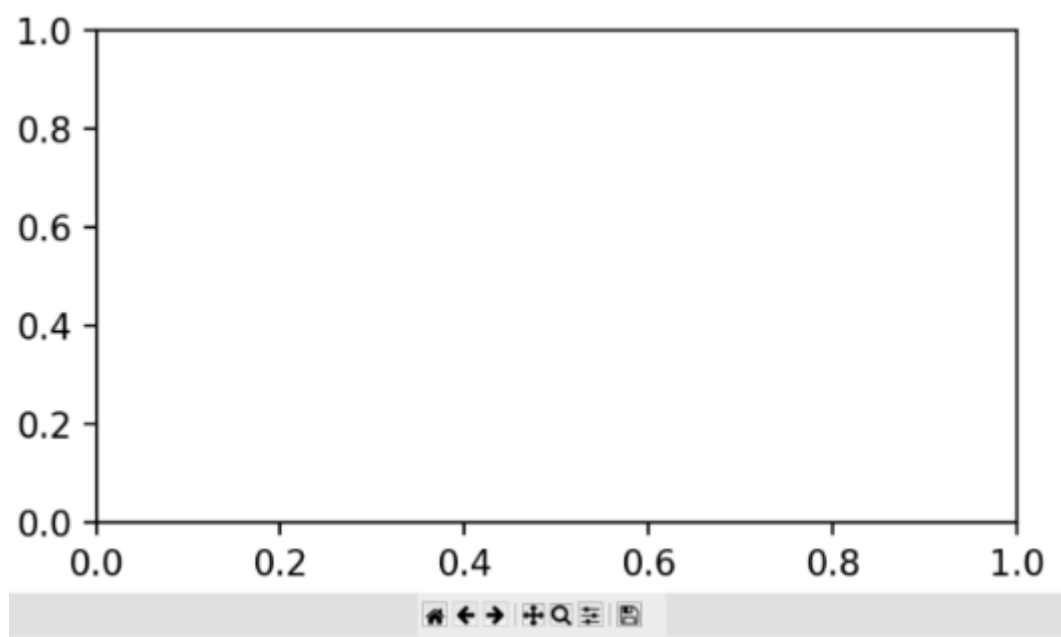


Podemos observar en la parte superior un gran botón para darle visibilidad mediante el cual se cargan los ficheros de tipo *parquet*.

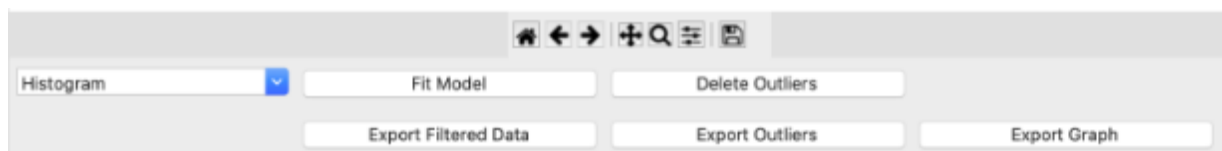
3.-MÉTODOS

En esta sección se describen los métodos utilizados para el análisis y clasificación de los datos médicos contenidos en la base de datos proporcionada, con el objetivo de extraer información relevante sobre los escaneos cerebrales de los pacientes.

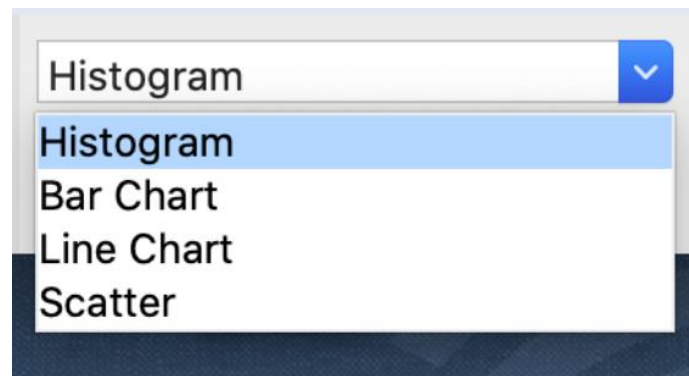
En nuestra interfaz propuesta hemos ubicado un toolbar estándar para que la manipulación de las gráficas sea más intuitiva y sencilla de comprender.



Bajo la toolbar, creamos un menú de selección de gráficas con todas las opciones disponibles para su visualización, junto a una serie de botones destinados a la exportación y regresión de los datos. Veamos este apartado en profundidad:



Los tipos de gráfica seleccionados han sido los siguientes:



Al pulsar sobre el botón Fit Model, aparecerá una ventana emergente donde podremos seleccionar la función de regresión que más se adapte a nuestras necesidades en cada momento:

Exponential Regression: Utilizado para observar crecimientos o decrecimientos exponenciales de variables en función del tiempo o de la variable deseada.

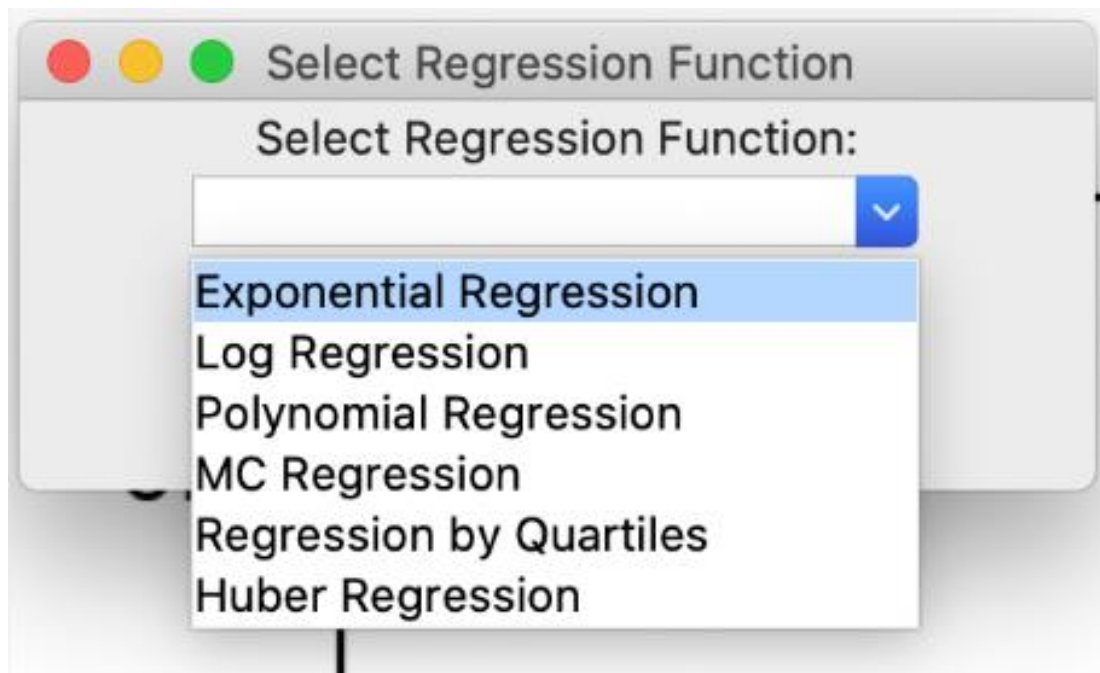
Log Regression: Tiene su utilidad en analizar la forma en la que una variable cambian con el tiempo o en relación con una variable independiente.

Polynomial Regression: Sirve si sospechas que la relación entre las variables no es lineal, por ejemplo, si esperas que el volumen cerebral cambie de manera no lineal con la edad o con alguna otra variable.

MC Regression: Puede ser útil como un método general para modelar la relación entre el volumen cerebral y una o más variables predictoras, encontrando una mejor línea de ajuste a los valores propuestos.

Regression by Quartiles: Es útil para capturar patrones no lineales en los datos a lo largo de diferentes rangos de valores, dividiendo los datos en grupos según su posición.

Huber Regression: Si tus datos contienen valores atípicos que podrían sesgar los resultados de la regresión, la regresión de Huber podría ser útil para obtener estimaciones más robustas de la relación entre los volúmenes cerebrales y otras variables.



Con las funciones de *export* y *delete outliers*, podemos identificar datos anómalos que se salen del rango que hemos considerado adecuado, pudiendo ser identificados con su exportación del id o borrados para crear una gráfica con únicamente los valores que consideramos aptos para el experimento en cuestión que se esté desarrollando en ese preciso momento.

El desarrollo de clasificación de datos médicos se llevó a cabo siguiendo una metodología estructurada que incluyó los siguientes pasos:

Diseño e Implementación de la Interfaz:

Se diseñó e implementó una interfaz de usuario para interactuar con la base de datos y visualizar los resultados del análisis. Esta interfaz permitió a los usuarios cargar datos, seleccionar las características a analizar y realizar diversas operaciones como graficar, ajustar modelos y exportar resultados.

Conexión de la Base de Datos al Código:

Se desarrolló una funcionalidad para conectar la base de datos al código, lo que permitió acceder y manipular los datos de forma programática. Esto facilitó la integración de la base de datos en el flujo de trabajo de análisis y clasificación.

Implementación de Funciones de Análisis:

Se crearon funciones específicas para realizar diferentes acciones de análisis, como la extracción de características relevantes, la clasificación de los datos y la evaluación de los modelos generados.

Selección de Características Relevantes:

Se llevó a cabo una cuidadosa selección de las características más relevantes de los escaneos cerebrales para su inclusión en el modelo de clasificación.

Experimentación y Evaluación:

Se realizaron experimentos utilizando diferentes algoritmos de clasificación y configuraciones de parámetros para evaluar su rendimiento en la tarea de clasificación de datos cerebrales.

Optimización y Ajuste de Modelos:

Se realizaron iteraciones en el proceso de desarrollo para optimizar y ajustar los modelos de clasificación con el fin de mejorar su rendimiento y generalización a datos nuevos. Esto implicó la exploración de diferentes técnicas de ajuste de modelos y la selección de la más adecuada para cada caso.

3.2- FUNCIONES DEL CÓDIGO A DESTACAR

def update_plot(self, data, graph_type="Histogram"): Tiene la función de actualizar la gráfica con los datos proporcionados, mostrando histogramas como vemos en la definición o gráficos de barras, línea o dispersión mediante el uso de “*elif graph_type == x*”. Es clave visualizar los datos de distintas formas para sacar el máximo de información posible.

def filter_data(self, age_range, sex_value, feature_of_interest, feature_of_interestB, quality_range): Su propósito es aplicar los distintos tipos de filtros disponibles a los datos en función de la edad, sexo o calidad que hayamos seleccionado. Permite centrarnos de forma efectiva en la información que realmente nos interesa.

def valid_age_format(string): Garantiza que el usuario ha introducido un rango de edad viable y en el formato adecuado, sirviendo de garantía para que no se produzcan fallos en este aspecto.

def valid_quality_format(string): Tiene un funcionamiento similar al *valid_age_format*, pero en este caso valida el formato de rango de calidad introducido por el usuario, en este caso, el formato es *float-float* de 0-1.

def resize_barplot_ticks(self): Ajusta automáticamente el tamaño de las variables en el gráfico para que resulten legibles.

def apply_regression(): Aplica un tipo de regresión en función de la selección que hayamos realizado. Junto a *fit_model()*, se encargan de ajustar la regresión a los datos.

def contains_word_from_list(string, word_list): Comprueba que el texto proporcionado contenga una palabra de una lista dada.

def search_optionsA/B(self, event, feature_list): En función de la base de datos proporcionada, muestra en el menú desplegable de la variable de interés A o B las opciones disponibles.

def export_data(self): Permite exportar los datos de interés para ser tratados posteriormente, en formato “.txt” por defecto.

def export_graph(self): Con un funcionamiento similar a la anterior función, permite exportar en formato “.png” por defecto la imagen del gráfico.

La regresión se puede aplicar gracias a la definición siguientes modelos, descritos teóricamente en el punto 3:

```
def reg_MC(self, x, y):  
    # Ajuste del modelo de regresión robusta por mínimos cuadrados  
    self.model = RANSACRegressor()  
    self.model.fit(x, y)  
  
    # Predicciones del modelo  
    y_fit = self.model.predict(x)  
  
    # Calcular el error del modelo  
    residuals = y - y_fit  
    scale = 1.4826 # Factor de escala para convertir MAD a desviación estándar (para distribución normal)  
    mad_residuals = mad(residuals)  
    std_error = scale * mad_residuals  
  
    # Límites del intervalo de confianza (por ejemplo, +/- 1.5 MAD)  
    scale_factor = 1.5  
    self.lower_limit = y_fit - scale_factor * std_error  
    self.upper_limit = y_fit + scale_factor * std_error  
  
    return x, y_fit
```

Ilustración 2. Modelo de Regresión Robusta por mínimos Cuadrados

```
def reg_quartiles(self, x, y):  
    # Ajuste del modelo de regresión robusta por mínimos cuadrados  
    self.model = RANSACRegressor()  
    self.model.fit(x, y)  
  
    # Predicciones del modelo  
    y_fit = self.model.predict(x)  
  
    # Calcular los residuos  
    residuals = y - y_fit  
  
    # Calcular los residuos ponderados (residuos absolutos)  
    weighted_residuals = np.abs(residuals)  
  
    # Utilizar más datos para calcular los cuartiles  
    weighted_residuals_all = np.concatenate((weighted_residuals, weighted_residuals))  
  
    # Calcular los cuantiles modificados de los residuos ponderados  
    q1 = np.percentile(weighted_residuals_all, 25)  
    q3 = np.percentile(weighted_residuals_all, 75)  
  
    # Calcular los límites del intervalo de confianza basados en los cuartiles modificados  
    iqr = q3 - q1  
    scale_factor = 1.5  
    self.lower_limit = y_fit - scale_factor * iqr  
    self.upper_limit = y_fit + scale_factor * iqr  
  
    # Regresar los resultados  
    return x, y_fit
```

Ilustración 3 Modelo de Regresión por RANSAC

```

def reg_huber(self, x, y):
    model = HuberRegressor()
    model.fit(x, y)

    # Predicciones del modelo
    y_fit = model.predict(x)

    # Si y es un DataFrame o Serie de Pandas, conviértelo a un array NumPy
    if isinstance(y, (pd.DataFrame, pd.Series)):
        y = y.to_numpy()

    # Calcular los residuos de Huber
    residuals = y - y_fit

    # Calcular la desviación estándar de los residuos
    std_error = np.std(residuals)

    # Límites del intervalo de confianza (por ejemplo, +/- 1.5 desviaciones estándar)
    scale_factor = 1.5
    self.lower_limit = y_fit - scale_factor * std_error
    self.upper_limit = y_fit + scale_factor * std_error

    return x, y_fit,

```

Ilustración 4. Modelo de Regresión por Huber

```

def reg_exponential(self, x, y):
    # Definir la función exponencial
    def exponential_func(x, a, b, c):
        return a * np.exp(b * x) + c

    # Inicializar los parámetros de la función exponencial
    p0 = (1.0, 1.0, 1.0) # Valores iniciales de los parámetros (a, b, c)

    # Ajuste de la curva exponencial a los datos
    try:
        popt, pcov = curve_fit(exponential_func, x, y, p0=p0)
    except RuntimeError:
        # Si no se puede ajustar la curva exponencial, devolver None para indicar un fallo
        print("No se puede usar la funcion exponencial")
        return None, None

    # Predicciones del modelo
    y_pred = exponential_func(x, *popt)

    # Calcular el error del modelo
    mse = mean_squared_error(y, y_pred)
    std_error = np.sqrt(mse) # Desviación estándar del error residual

    # Límites del intervalo de confianza (por ejemplo, +/- 1 desviación estándar)
    # Multiplicar la desviación estándar por un factor de escala
    scale_factor = 2
    self.lower_limit = exponential_func(x, *popt) - scale_factor * std_error
    self.upper_limit = exponential_func(x, *popt) + scale_factor * std_error

    return x, y_pred

```

Ilustración 5. Modelo de Regresión Exponencial


```

def reg_polynomial(self, x, y):
    # Transformación de características polinómicas
    polynomial_features = PolynomialFeatures(degree=2)
    self.X_poly = polynomial_features.fit_transform(x)

    # Ajuste del modelo de regresión lineal
    self.model = LinearRegression()
    self.model.fit(self.X_poly, y)

    # Predicciones del modelo
    X_fit = np.linspace(min(x), max(x), 100).reshape(-1, 1)
    X_fit_poly = polynomial_features.transform(X_fit)
    y_fit = self.model.predict(X_fit_poly)

    # Calcular el error del modelo
    y_pred = self.model.predict(self.X_poly)
    mse = mean_squared_error(y, y_pred)
    std_error = np.sqrt(mse) # Desviación estándar del error residual

    # Límites del intervalo de confianza (por ejemplo, +/- 1 desviación estándar)
    # Multiplicar la desviación estándar por un factor de escala
    scale_factor = 2
    self.lower_limit = y_fit - scale_factor * std_error
    self.upper_limit = y_fit + scale_factor * std_error

    print(self.lower_limit)

    return X_fit, y_fit

```

Ilustración 6. Modelo de Regresión Polinomial

E3.3-EXPERIMENTACIÓN: EN FORMA DE TABLAS Y EXPLICACIONES

Los distintos tipos de gráficos desarrollados para la puesta en práctica han sido los siguientes:

Histograma: Útil para visualizar la distribución de una variable continua en diferentes rangos o intervalos. Permite identificar la frecuencia con la que se presentan ciertos valores y proporciona una representación visual de la forma en que los datos están distribuidos.

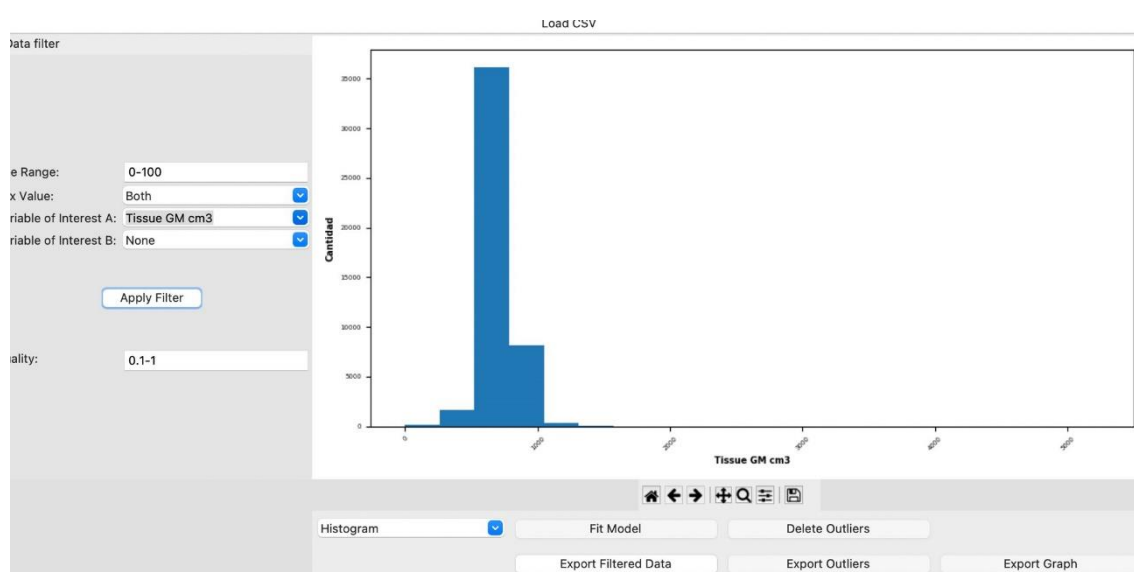


Gráfico de Barras: Permite comparar diferentes categorías o grupos de datos de forma visual. La altura de cada barra representa la magnitud de la variable en cada categoría, lo que facilita la comparación entre ellas.

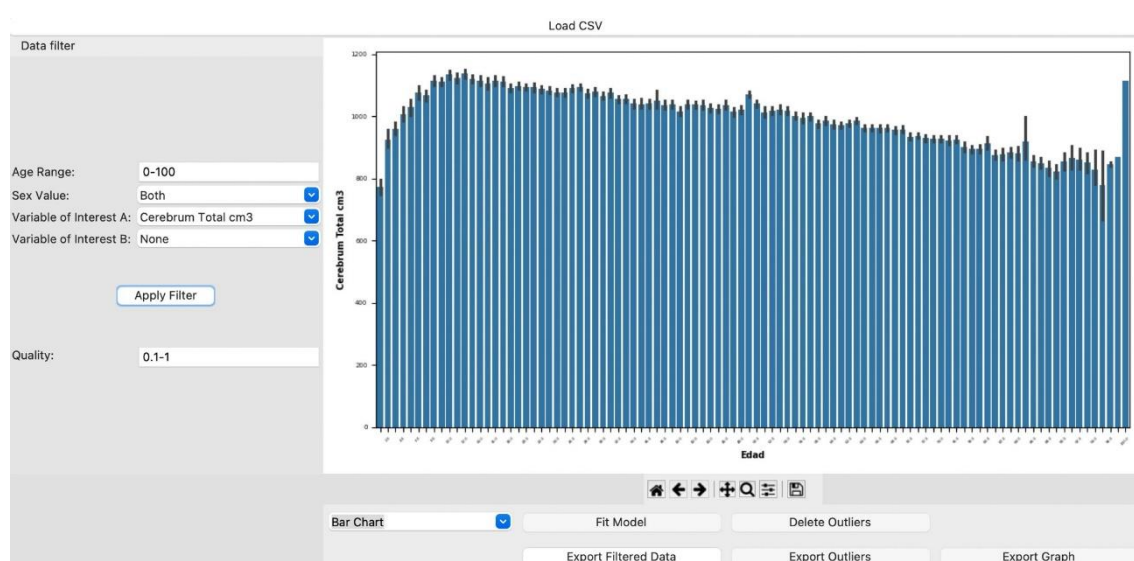
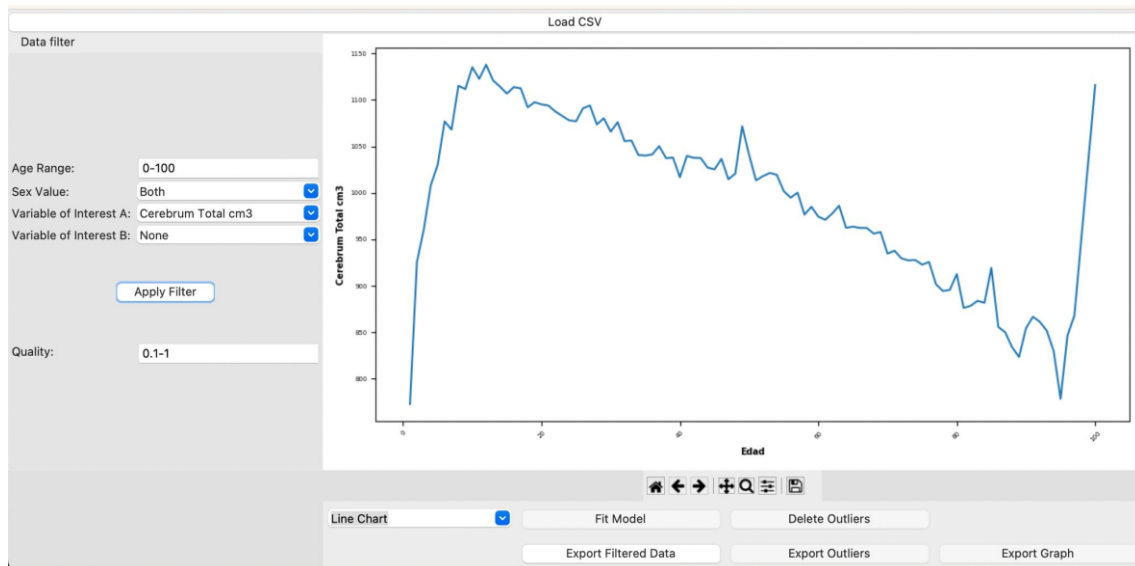
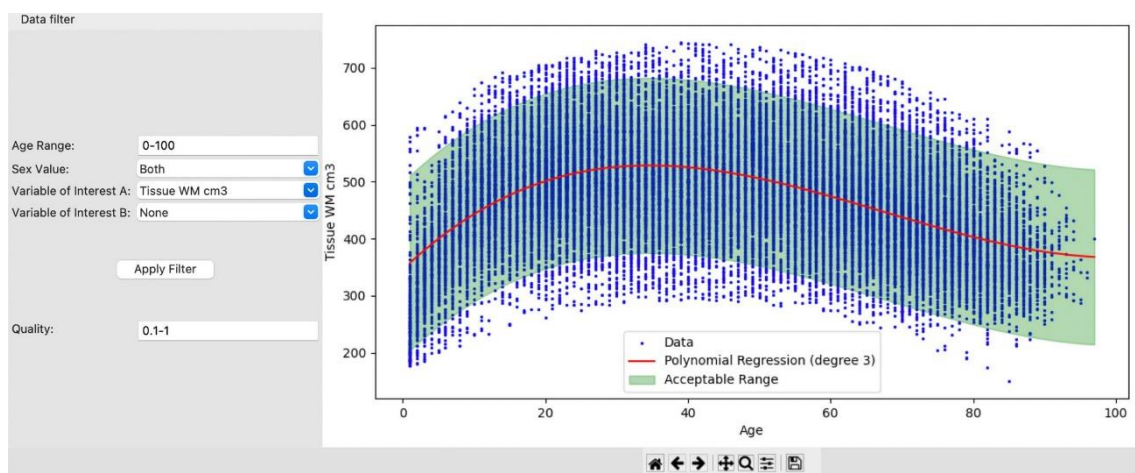


Gráfico de línea: Un gráfico de línea es útil para mostrar la evolución o tendencia de una variable a lo largo del tiempo o de una variable independiente. Este tipo de gráfico permite identificar patrones de cambio y tendencias a lo largo de un eje temporal o de otra variable relevante.



Dispersión: Útil para visualizar la relación entre dos variables continuas. En el análisis de datos médicos, podría utilizarse para explorar la relación entre diferentes medidas de tejido cerebral, como el volumen de tejido gris y el volumen de tejido blanco. Este tipo de gráfico permite identificar patrones de correlación, dispersión de los datos y posibles relaciones entre las variables analizadas.



3.4-EVALUACIÓN

En esta etapa, se llevaron a cabo una serie de experimentos para explorar el rendimiento de los modelos de clasificación en la base de datos de escaneos cerebrales. Los experimentos se diseñaron con el objetivo de evaluar la capacidad de los modelos para distinguir entre diferentes categorías de pacientes y para identificar patrones relevantes en los datos.

Los experimentos se organizaron de la siguiente manera:

Selección de Variables de Interés:

Se identificaron las variables más relevantes en la base de datos para la clasificación de pacientes. Esto incluyó características demográficas como edad y sexo, así como mediciones específicas extraídas de los escaneos cerebrales.

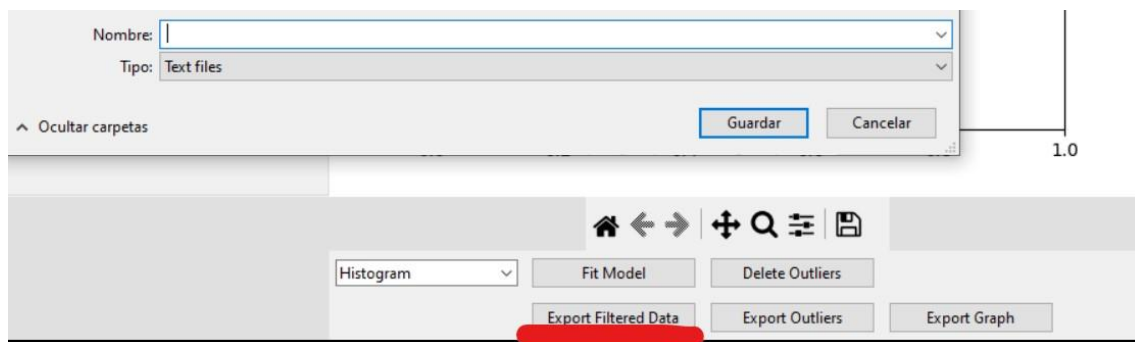
Análisis de Resultados:

Se analizaron los resultados de los experimentos para identificar patrones y tendencias en los datos, así como para evaluar la capacidad predictiva de los modelos en la clasificación de pacientes. Se prestaron especial atención a las características más discriminativas y a la capacidad de generalización de los modelos.

4-RESULTADOS

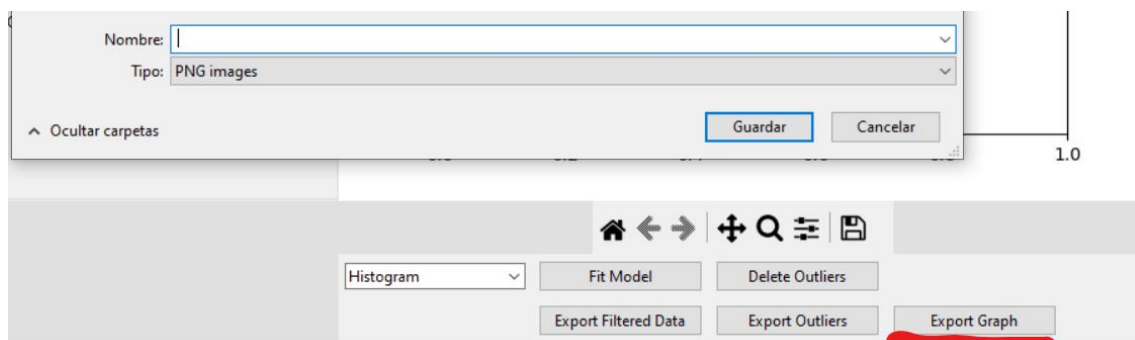
Se esperaba que los experimentos realizados proporcionaran información valiosa sobre la capacidad de los modelos de clasificación para distinguir entre diferentes categorías de pacientes basándose en los datos de escaneos cerebrales. Los resultados obtenidos permitirían identificar patrones y tendencias significativas en los datos, así como evaluar la eficacia de los modelos desarrollados.

Adicionalmente, la aplicación nos ha permitido, mediante las herramientas de Fit Model y Exportación, crear gráficos mucho más afines y fiables, así como conseguir aislar casos anómalos, identificados correctamente, para ser tratados posteriormente o ser eliminados en caso de carecer de utilidad.



Los datos filtrados también pueden ser extraídos en forma de fichero de texto, proporcionándonos la ID de cada uno de ellos de forma que tengan una identificación.

El gráfico también puede ser exportado en formato PNG, para ser utilizado en presentaciones o artículos científicos para dar soporte y contrastar una determinada información.



En este apartado, se discuten las implicaciones de los resultados obtenidos, se destacan las limitaciones del estudio y se proponen áreas para futuras investigaciones.

Implicaciones Clínicas y Científicas

Los resultados de este estudio tienen implicaciones significativas en el campo de la informática médica. El análisis de características cerebrales puede proporcionar información valiosa para el diagnóstico precoz, la estratificación de pacientes y el desarrollo de tratamientos personalizados.

Limitaciones y Consideraciones

Es importante tener en cuenta varias limitaciones en este estudio, como el tamaño limitado de la muestra y la heterogeneidad de los datos recopilados. Además, la falta de información detallada sobre ciertos factores clínicos y demográficos puede haber influido en los resultados obtenidos.

Direcciones Futuras

Para mejorar la validez y la generalización de los resultados, se sugieren varias líneas de investigación futuras. Esto incluye la realización de estudios longitudinales con cohortes más grandes, la incorporación de datos genéticos y ambientales, y el desarrollo de modelos predictivos más sofisticados mediante técnicas de aprendizaje profundo.

6-CONCLUSIONES

En resumen, el sistema descrito en esta memoria técnica proporciona una plataforma sólida y versátil para el análisis y visualización de datos médicos. Con su interfaz intuitiva, potentes capacidades de filtrado y una amplia gama de opciones de visualización, el sistema se posiciona como una herramienta valiosa para investigadores, profesionales médicos y cualquier persona involucrada en el análisis de datos médicos complejos. Con futuras mejoras y desarrollos, este sistema tiene el potencial de contribuir significativamente al avance del campo médico y la investigación biomédica.

MEJORAS POSIBLES

Visualización 3D: Se podrían incorporar técnicas de visualización en 3D para explorar datos volumétricos de imágenes cerebrales de manera más efectiva.

Interfaz Web: Se podría desarrollar una versión basada en web del sistema para facilitar el acceso desde diferentes dispositivos y plataformas.

Análisis de Series Temporales: Se podrían agregar herramientas para el análisis de series temporales de datos médicos, especialmente útiles en el seguimiento de la progresión de enfermedades.

7-REFERENCIAS

-