UNIVERSIDAD DE BURGOS

TFG DEL GRADO DE INGENIERÍA INFORMÁTICA - GII 22.40 APLICACIÓN WEB PARA REALIZAR ANÁLISIS DE COMPONENTES PRINCIPALES, DE CLUSTERING Y DE DETECCIÓN DE OUTLIERS.

ADCO

Presentado por Enrique Diez Fernández en
Universidad de Burgos el 9 de junio de 2023

Luis Rodrigo Izquierdo Millán

José Manuel Galán Ordax

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. José Manuel Galán Ordax y D. Luis Rodrigo Izquierdo Millan, profesores del departamento de Ingeniería de Organización, área de Organización de Empresas.

Expone:

Que el alumno D. Enrique Diez Fernández, con DNI 71364254S, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado ADCO.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 9 de junio de 2023

Vº. Bº. del Tutor:

D. José Manuel Galán Ordax

Vº. Bº. del Tutor:

D. Luis Rodrigo Izquierdo Millan

RESUMEN

El presente proyecto consiste en el desarrollo de una aplicación web desarrollada en Angular, con el propósito de proporcionar a la comunidad científica una herramienta gratuita para el análisis de datos.

La aplicación ha sido diseñada para ser fácil de usar y accesible para usuarios sin conocimientos avanzados de programación. A través de una interfaz intuitiva y amigable, los usuarios podrán cargar conjuntos de datos, realizar análisis de componentes principales para identificar patrones y reducir la dimensionalidad de los datos. Además, podrán calcular distancias entre diferentes puntos o elementos, realizar clustering para agrupar datos similares y detectar outliers en el conjunto de datos.

Con el objetivo de promover el acceso y el uso gratuito, la aplicación ha sido licenciada bajo una licencia gratuita y abierta. De esta manera, se busca fomentar la colaboración y contribución de la comunidad científica, permitiendo a los investigadores, académicos y profesionales utilizar la herramienta de manera libre y sin restricciones.

La aplicación ha sido desplegada en la plataforma de Netlify, que ofrece servicios gratuitos de alojamiento web. Esto permite que la aplicación esté disponible en línea y sea accesible desde cualquier dispositivo con conexión a Internet. Netlify garantiza la disponibilidad y la escalabilidad necesarias para manejar cargas de trabajo significativas, asegurando así que la aplicación pueda ser utilizada por una amplia gama de usuarios.

DESCRIPTORES

Aplicación web, Angular, análisis de datos, conjuntos de datos, análisis de componentes principales, cálculo de distancias, clustering, detección de outliers, licencia gratuita, Netlify, alojamiento web, disponibilidad, escalabilidad.

ABSTRACT

This project consists of developing a web application using Angular, with the purpose of providing the scientific community with a free tool for data analysis.

The application has been designed to be user-friendly and accessible to users without advanced programming knowledge. Through an intuitive and friendly interface, users will be able to upload datasets, perform principal component analysis to identify patterns and reduce data dimensionality. Additionally, they will be able to calculate distances between different points or elements, perform clustering to group similar data, and detect outliers in the dataset.

To promote access and free use, the application has been licensed under a free and open license. This aims to encourage collaboration and contribution from the scientific community, allowing researchers, academics, and professionals to use the tool freely and without restrictions.

The application has been deployed on the Netlify platform, which offers free web hosting services. This ensures that the application is available online and accessible from any device with an internet connection. Netlify guarantees the necessary availability and scalability to handle significant workloads, ensuring that the application can be used by a wide range of users.

KEYWORDS

Web application, Angular, data analysis, data sets, principal component analysis, distance calculation, clustering, outlier detection, free license, Netlify, web hosting, availability, scalability.

Índice de contenidos

Índice de contenidos 2
Índice de imágenes 4
Índice de ecuaciones 5
Índice de tablas 5
1. Introducción 6
1.1. Contexto 6
1.2. Estructura de la memoria 6
2. Objetivos 8
2.1. Objetivos generales 8
2.2. Objetivos personales 8
3. Conceptos teóricos 10
3.1. Algoritmo de Componentes Principales 10
3.2. Algoritmos de Clustering 19
3.2.1. Clustering 19
3.2.2. K-means 22
3.2.3. DBSCAN 23
3.2.4. OPTICS 24
3.3. Distancias 25
3.3.1. Mahalanobis 25
3.3.2. Euclídea 26
3.3.3. Euclídea normalizada 26
3.4. Outliers 27
3.4.1. Mahalanobis 27
3.4.2. DBSCAN 28
3.4.3. OPTICS 28
3.4.4. K-nearest neighbors 28
3.4.5. LOF 29
4. Técnicas y herramientas 30
4.1. Metodología de gestión de proyectos 30
4.2. Editor de texto para la memoria 31
4.3. Referencias bibliográficas, citas y plagio. 31
4.4. Repositorios de código 32
4.5. Herramientas de prototipado 32
4.6. Software para hacer documentación automática 33

4.7. Herramienta para el lanzamiento de la aplicación 33
4.8. Herramientas para la creación de código 35
4.9. Librerías del proyecto 35
4.9.1. XLSX 35
4.9.2. Librerías para representar Gráficos 35
4.9.3. Librerías de componentes: 36
4.9.4. PCA (pca-js y ml-pca) 37
4.9.5. Html2canvas y jsPDF 38
4.9.6. Math y math.js 38
4.9.7. Rxjs 38
5. Aspectos relevantes del desarrollo del proyecto 40
6. Trabajos relacionados 44
7. Conclusiones y líneas de trabajo futuras 46
7.1. Conclusiones 46
7.2. Líneas de trabajo futuras 46
8. Bibliografía 48

Índice de imágenes

Ilustración 1- Imagen del funcionamiento del ACP. Fuente: Elaboración propia 10
Ilustración 2 - ACP de una distribución gaussiana multivariante centrada en (1,3) con una desviación estándar de 3 en aproximadamente la dirección (0,866, 0,5) y de 1 en la dirección ortogonal 10
Ilustración 3 - Figuras en 3D y 2D. 12
Ilustración 4 - Ejemplo de la diferencia de magnitud entre las direcciones principales en dos dimensiones. 15
Ilustración 5 - Ejemplo de la transformación de datos con dos variables a una única componente tras el análisis PCA. 15
Ilustración 6 - Ejemplo Clustering de datos tras PCA. 16
Ilustración 7 - Imagen Propia del proyecto web 18
Ilustración 8 - Imagen Propia del proyecto web 18
Ilustración 9 - Enlace simple en datos Gaussianos. En 35 grupos, al principio el grupo más grande se fragmenta en grupos más pequeños, mientras que todavía está conectado al segundo mayor por el efecto de enlace simple. 19
Ilustración 10 - Enlace simple en agrupamiento basado en densidad. Se extrajeron 20 grupos, la mayoría contienen un único elemento, nos podemos percatar entonces que enlace simple no tiene una noción de ruido. 20
Ilustración 11 – Cómo K-means separa los datos 20
Ilustración 12 - En datos distribuidos con Gaussianas, EM trabaja bien, desde entonces se utilizan Gaussianas para la modelación de grupos. 21
Ilustración 13 - Grupos basados en densidad no pueden ser modelados utilizando distribuciones Gaussianas 21
Ilustración 14 - Funcionamiento de OPTICS. 24
Ilustración 15 - Principales aplicaciones de outliers 27
Ilustración 16 - Tamaños de carga en Netlify 34

Índice de ecuaciones

Ecuación 1 - Vector de pesos 11
Ecuación 2 - Componentes principales 11
Ecuación 3 - Cálculo de la proyección 11
Ecuación 4 - Peso de la primera componente principal 11
Ecuación 5 - Ecuación para el cálculo del peso de una componente 12
Ecuación 6 - Matriz de covarianza 13
Ecuación 7 - Proyección sobre la componente 17
Ecuación 8 - Distancia de K-means 22
Ecuación 9 - Posición promedio K-means 22
Ecuación 10 - Distancia de Mahalanobis 25
Ecuación 11 - Distancia euclídea 26
Ecuación 12 - Distancia euclídea normalizada 26

Índice de tablas

Tabla 1 - Datos de ejemplo 16
Tabla 2 - Tabla X^T * X 16
Tabla 3 - Matriz de covarianza 17
Tabla 4 - Polinomio característico de la matriz de covarianza 17
Tabla 5 - Autovectores 17
Tabla 6 - Proyección de los datos 18
Tabla 7 - Comparativa entre Mendeley y Zotero 31

1. Introducción

Este primer capítulo tiene como finalidad dar una visión general del objetivo del proyecto, así como explicar tanto su estructura como la metodología seguida para realizarlo.

1.1. Contexto

En la comunidad científica, se requieren herramientas accesibles y poderosas para el análisis de datos. La creación de una aplicación web en Angular, centrada en el análisis de componentes principales, cálculo de distancias, clustering y detección de outliers, tiene como objetivo proporcionar una solución gratuita que simplifique el análisis de datos complejos. Esto permitirá a los investigadores de diferentes disciplinas acceder a herramientas avanzadas, aumentar la eficiencia en la investigación, fomentar la colaboración interdisciplinaria y promover la innovación científica.

1.2. Estructura de la memoria

La memoria se estructurará de la siguiente manera:

1. **Introducción:** Proporciona una visión general del proyecto, presentando su propósito y contexto.

2. **Objetivos:** Establece los objetivos específicos que se pretenden alcanzar con el proyecto, delineando las metas a cumplir.

3. **Conceptos teóricos:** Explora los fundamentos teóricos y conceptuales relacionados con el tema del proyecto, brindando una base de conocimientos necesaria para su comprensión.

4. **Técnicas y herramientas:** Describe las técnicas y herramientas utilizadas en el desarrollo del proyecto, detallando su aplicación y relevancia.

5. **Aspectos relevantes del desarrollo del proyecto:** Presenta los aspectos más destacados del desarrollo del proyecto, como los desafíos encontrados, las decisiones clave tomadas y los logros alcanzados.

6. **Trabajos relacionados:** Revisa investigaciones previas y proyectos relacionados con el tema, resaltando su relevancia y estableciendo cómo el proyecto se diferencia o contribuye a este campo.

7. **Conclusiones y líneas de trabajo futuras:** Resume las conclusiones obtenidas a partir del proyecto y propone posibles direcciones para futuras investigaciones o mejoras.

8. **Bibliografía:** Lista las fuentes bibliográficas consultadas y referenciadas a lo largo de la memoria, asegurando la credibilidad y fundamentación del trabajo.

2. Objetivos

2.1. Objetivos generales

El objetivo principal de este proyecto es desarrollar una aplicación web en Angular que aproveche las capacidades de análisis de datos avanzado para realizar una serie de tareas fundamentales. La aplicación se centrará en la implementación de algoritmos de análisis de componentes principales, cálculo de distancias, clustering y detección de outliers, con el fin de proporcionar a los usuarios una herramienta eficiente y fácil de usar para explorar y comprender sus conjuntos de datos.

1. **Análisis de Componentes Principales:** Uno de los objetivos clave de la aplicación es permitir a los usuarios realizar un análisis de componentes principales (PCA) sobre sus datos. El PCA es una técnica poderosa que reduce la dimensionalidad de los conjuntos de datos, identifica las variables más influyentes y facilita la visualización de la estructura subyacente. La aplicación brindará a los usuarios la capacidad de cargar sus datos, realizar el PCA y explorar los resultados de manera interactiva.
2. **Cálculo de Distancias:** Otra funcionalidad importante que se implementará en la aplicación es el cálculo de distancias entre puntos o instancias en un conjunto de datos. Esto permitirá a los usuarios medir la similitud o la diferencia entre diferentes observaciones, lo que puede ser útil en diversas aplicaciones, como la clasificación de objetos o la agrupación de elementos similares.
3. **Clustering:** La aplicación también ofrecerá algoritmos de clustering para agrupar automáticamente las instancias en conjuntos homogéneos. Estos algoritmos permitirán a los usuarios identificar patrones ocultos o segmentos dentro de sus datos, lo que puede ayudar en la toma de decisiones y la comprensión de las características intrínsecas de los conjuntos de datos.
4. **Detección de Outliers:** Además, la aplicación proporcionará herramientas para detectar outliers o valores atípicos en los conjuntos de datos. Los outliers pueden indicar observaciones anómalas o errores en los datos y pueden influir en los resultados del análisis. Los usuarios podrán identificar y gestionar los outliers de manera efectiva, lo que mejorará la calidad de los análisis y las conclusiones obtenidas.

En resumen, el objetivo de este proyecto es desarrollar una aplicación web en Angular que integre de manera fluida las capacidades de análisis de componentes principales, cálculo de distancias, clustering y detección de outliers. Al proporcionar una interfaz intuitiva y potente, la aplicación permitirá a los usuarios explorar, comprender y aprovechar al máximo sus datos, brindando información valiosa para la toma de decisiones y la generación de conocimiento en diversos ámbitos.

2.2. Objetivos personales

En este apartado se establecen los siguientes objetivos personales:

1. Aplicar la metodología SCRUM en un proyecto real con el propósito de familiarizarme con ella y mejorar mi comprensión de su utilidad y funcionamiento.
2. Adquirir habilidades en el desarrollo de aplicaciones web utilizando Angular y Visual Studio Code.

3. Explorar y utilizar nuevas herramientas que no había utilizado previamente, ampliando así mi conjunto de habilidades.
4. Aplicar los conocimientos adquiridos en el análisis de datos, poniéndolos en práctica de manera efectiva.

3. Conceptos teóricos

3.1. Algoritmo de Componentes Principales

En estadística, el análisis de componentes principales (ACP: en inglés, PCA, *Principal Component Analysis*) es una técnica inventada en 1901 por Karl Pearson, empleada para crear, a partir de unas variables originales, un conjunto reducido de variables no correlacionadas, con la intención de perder la menor cantidad de información posible. Las nuevas variables son combinaciones lineales de las variables originales, y se denominan componentes.

Una vez creados, los componentes se ordenan por la cantidad de varianza original que describen, de forma descendiente. De esta forma, se pueden eliminar los componentes finales, los que recogen menor varianza, reduciendo el número de variables y a la vez perdiendo la menor cantidad de información posible (Ilustración 1).

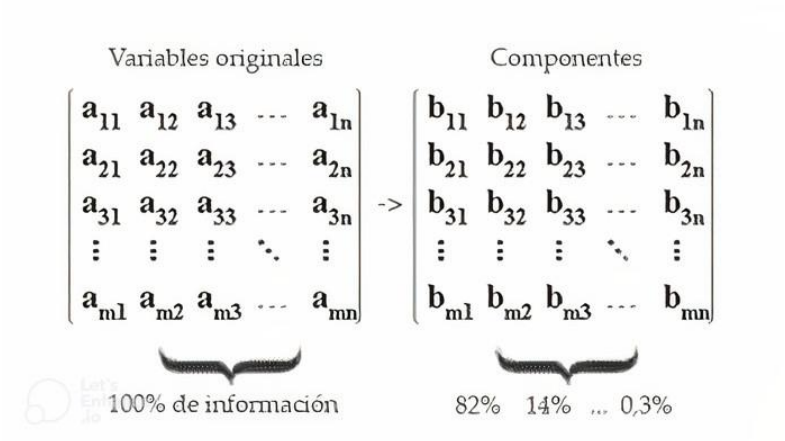

Ilustración 1- Imagen del funcionamiento del ACP. Fuente: Elaboración propia

La finalidad principal del uso de ACP en este proyecto, es la posibilidad de reducir la dimensionalidad de las variables iniciales. Esto se aplicará para poder representar los datos y posteriormente aplicar clustering para realizar un agrupamiento de los datos (Ilustración 2).

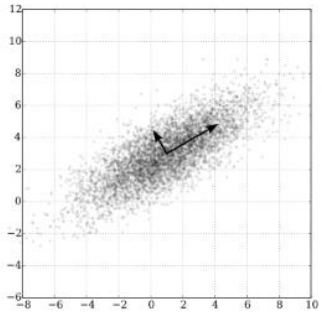

Ilustración 2 - ACP de una distribución gaussiana multivariante centrada en (1,3) con una desviación estándar de 3 en aproximadamente la dirección (0,866, 0,5) y de 1 en la dirección ortogonal

ACP es un algoritmo utilizado principalmente en el análisis exploratorio de datos y con la finalidad de construir modelos predictivos.

Para el análisis de datos, el primer componente principal para un conjunto n de variables es la combinación lineal de las variables originales que explica la mayor parte de la varianza. El segundo componente principal será aquel que explique la mayor parte de la varianza, una vez eliminado el efecto del primer componente principal. Esto sigue así, hasta las n iteraciones, donde encontraremos la combinación lineal que explica la última (y menor) parte de la varianza.

Algoritmo:

Consideramos una matriz de datos X, de tamaño n x p. Los datos se recogen por filas; es decir, tenemos n datos y p variables.

Una combinación lineal de las variables originales va a venir definida por un vector de p pesos (Ecuación 1), que multiplican a las p variables originales. El componente principal k viene determinada por los siguientes pesos:

$$W_{(k)} = (w_1, \dots, W_{(p)})_{(k)}$$

Ecuación 1 - Vector de pesos

Mediante el uso de estos vectores podemos mapear todas las columnas de X para obtener las coordenadas de los datos originales en la base de los componentes principales. Si tomamos l<=p componentes principales, la proyección del dato i sobre el subespacio generado por estos l componentes principales es:

$$t_{(i)} = (t_1, \dots, t_l)_{(i)}$$

Ecuación 2 - Componentes principales

La coordenada k del dato i en la base de los componentes principales viene dada por el producto escalar del vector $x_{(i)}$ de coordenadas del punto i en la base original, por el vector de pesos w(k) que define al componente principal k.

$$t_{k(i)} = x_{(i)} * w_{(k)} \quad \text{for} \quad i = 1, \dots, n \quad k = 1, \dots, l$$

Ecuación 3 - Cálculo de la proyección

Para obtener el conjunto de pesos W que definen a los componentes principales se tiene que ir analizando todas las varianzas para cada componente hasta llegar a las p varianzas.

El peso para la primera componente principal se va a obtener con:

$$w_{(1)} = \arg\max \left\{ \frac{w^T X^T X w}{w^T w} \right\}$$

Ecuación 4 - Peso de la primera componente principal

La cantidad para maximizar se puede reconocer como un cociente de Rayleigh. Un resultado estándar para una matriz semidefinida positiva como X^TX es que el valor máximo posible del cociente es el valor propio más grande de la matriz, lo que ocurre cuando w es el vector propio correspondiente.

Para calcular el resto de las componentes, debemos realizar los mismos cálculos, pero eliminando los primeros k-1 componentes principales de X, de esta forma, se nos queda la fórmula:

$$w_{(k)} = \arg max_{||w||=1} \left\{ \left| |\hat{X}_k w| \right|^2 \right\} = \arg\max \left\{ \frac{w^T \hat{X}_k^T \hat{X}_k w}{w^T w} \right\}$$

Ecuación 5 - Ecuación para el cálculo del peso de una componente

Ejemplo visual del ACP:

Una de las mejores formas de comprender la idea del ACP es trasponiéndolo a imágenes, lo cual es algo fácil de visualizar y de entender (Ilustración 3).

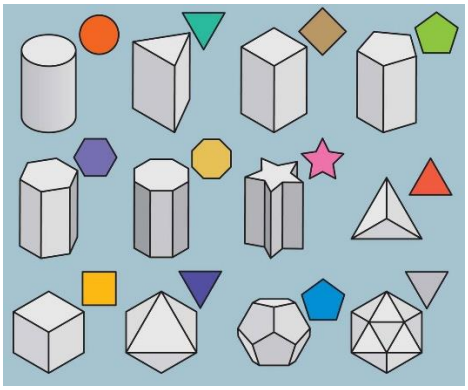

Ilustración 3 - Figuras en 3D y 2D.

Como podemos ver en estas imágenes, si tenemos una representación en tres dimensiones, podemos conocer a la perfección la forma geométrica que está siendo representada. Sin embargo, en el momento en el que la imagen pasa a tener una disposición en dos dimensiones, perdemos información importante sobre lo que estamos representando y de esta forma podríamos llegar a no saber diferenciar entre una pirámide y un prisma triangular. En el caso de que la dimensión fuera única, se habría perdido casi totalmente la información, ya que solo podríamos ver líneas rectas que se diferenciasen por el largo de la recta.

Esto es precisamente lo que intenta evitar el ACP. ACP trata de eliminar las variables que menor impacto tienen sobre la varianza, para que podamos seguir comprendiendo sobre qué objetos o elementos estamos trabajando. De esta forma, esa información perdida será la información menos relevante y nos permitirá clasificar datos de una manera más lógica y acorde a la realidad.

Matriz de covarianza:

Visualizar de forma gráfica cómo funciona el ACP es una forma sencilla de comprender el funcionamiento y el impacto de los pesos de las variables en función de la varianza, pero para poder trabajar con gran cantidad de variables que no se pueden representar de forma gráfica, lo lógico, es emplear la matriz de covarianza.

Una matriz de covarianza es una matriz cuadrada que nos indica cómo se relacionan dos o más variables entre sí (Ilustración 4). Es decir, nos muestra la medida en que dos variables cambian juntas. Los valores en la diagonal principal de la matriz representan las varianzas de cada variable, mientras que los valores fuera de la diagonal representan las covarianzas entre las variables. En resumen, la matriz de covarianza nos da una idea de cómo las variables en un conjunto de datos varían juntas.

$$\Sigma = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \cdots & \mathrm{Cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_n, X_1) & \mathrm{Cov}(X_n, X_2) & \cdots & \mathrm{Var}(X_n) \end{bmatrix}$$

Ecuación 6 - Matriz de covarianza

La varianza es una medida estadística que indica cuánto se dispersan los valores de un conjunto de datos respecto a su media. En otras palabras, mide la variabilidad o la diferencia entre los valores individuales y la media de los datos. Cuanto mayor sea la varianza, más dispersos estarán los valores y viceversa.

$$Var(X) = \frac{\sum_1^n (x_i - \bar{X})^2}{n}$$

Ecuación 7 - Ecuación de la Varianza

En la ecuación 1, x_i es cada valor en el conjunto de datos, $\bar{x}$ es la media de los datos, Σ representa la suma de todos los valores, y n es el número de valores en el conjunto de datos.

La covarianza es una medida estadística que indica cómo se relacionan dos variables en un conjunto de datos. Se define como el valor esperado del producto de las desviaciones de cada variable con respecto a su media.

$$Cov(X, Y) = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{n}$$

Ecuación 8 - Ecuación de la Covarianza.

En la ecuación 2, x_i e y_i son los valores individuales de cada variable, $\bar{x}$ e $\bar{y}$ son las medias de cada variable, Σ representa la suma de todos los valores, y n es el número de valores en el conjunto de datos.

En resumen, la matriz de covarianza se utiliza para calcular los componentes principales porque nos permite medir la covariación entre las variables originales y, por lo tanto, nos permite encontrar las combinaciones lineales que mejor explican la variabilidad total de los datos. Además, al calcular los componentes principales utilizando la matriz de covarianza, se asegura que los componentes sean ortogonales entre sí, lo que facilita su interpretación.

Vectores y valores propios:

Los valores propios son números escalares que representan la cantidad de variabilidad que es explicada por cada componente principal en un conjunto de datos. Un valor propio mayor indica que el componente principal correspondiente explica una mayor cantidad de variabilidad en los datos.

Las características de un valor propio son las siguientes:

- Cada valor propio tiene infinitos vectores propios asociados, dado que existen infinitos números reales que pueden formar parte de cada vector propio.
- Son escalares, pueden ser números complejos (no reales) y pueden ser idénticos (más de un valor propio iguales).
- Existen tantos valores propios como número de filas o columnas tiene la matriz original, ya que solo tienen sentido en matrices cuadradas.

Los vectores propios son vectores que indican la dirección de cada componente principal en el espacio de las variables originales. Cada vector propio está asociado con un valor propio y su longitud o magnitud indica la contribución de cada variable original en la formación del componente principal correspondiente.

Ejemplo:

Queremos obtener los vectores y valores propios de la siguiente matriz:

$$Z_{2x2} = \begin{pmatrix} 3 & -1 \\ -2 & 4 \end{pmatrix}$$

Debemos sustituir la matriz Z en la ecuación característica.

$$\left(\begin{pmatrix} 3 & -1 \\ -2 & 4 \end{pmatrix} - h \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) * V = 0$$

Simplificando, llegaremos a esta ecuación:

$$\begin{pmatrix} 3-h & -1 \\ -2 & 4-h \end{pmatrix} = 0$$

Buscamos el determinante de la matriz:

$$(3-h) * (4-h) - (-2 * -1) = 0$$

$$12 - 3h - 4h + h^2 - 2 = 0$$

$$h^2 - 7h + 10 = 0$$

La solución a la ecuación es h=2 y h=5; estos son los valores propios. Ahora, con esta información, podemos calcular los vectores propios asociados.

Para encontrar los vectores propios, hay que resolver las siguientes ecuaciones:

$$\begin{pmatrix} 3 & -1 \\ 2 & 4 \end{pmatrix} * \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 2 * \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

$$\begin{pmatrix} 3 & -1 \\ 2 & 4 \end{pmatrix} * \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 5 * \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

Por ejemplo, $(v_{1,}, v_2) = (1, 1)$ para h=2 y $(v_{1,}, v_2) = (-1, 2)$ para h=5:

$$\begin{pmatrix} 3 & -1 \\ -2 & 4 \end{pmatrix} * \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 2 * \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & -1 \\ -2 & 4 \end{pmatrix} * \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3*1-1*1 \\ -2*1+4*1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} = 2 * \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 3 & -1 \\ -2 & 4 \end{pmatrix} * \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 5 * \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & -1 \\ -2 & 4 \end{pmatrix} * \begin{pmatrix} -1 \\ 2 \end{pmatrix} = \begin{pmatrix} 3*-1-1*2 \\ -2*-1+4*2 \end{pmatrix} = \begin{pmatrix} -5 \\ 10 \end{pmatrix} = 5 * \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

De esta forma hemos encontrado los vectores propios que hacen que los datos de entrada que introducimos en v_1, v_2 sean los mismos datos que obtenemos en la salida.

Proyección y agrupamiento:

El hecho de buscar las direcciones que representen la máxima varianza posible se da debido a que se busca poder representar los datos de la forma más distante posible, lo cual nos ayudará a visualizar y diferenciar los diferentes puntos a representar.

Con una representación gráfica (ilustración 5) es sencillo interpretar y entender el funcionamiento. El siguiente ejemplo representa cómo diferentes datos que se representan en forma de círculo, cuando encontramos sus componentes principales y proyectar esos puntos iniciales sobre estos, vemos que hemos encontrado la forma de agrupar los valores de la forma más lejana y donde se pueden estudiar mejor los agrupamientos de los datos.

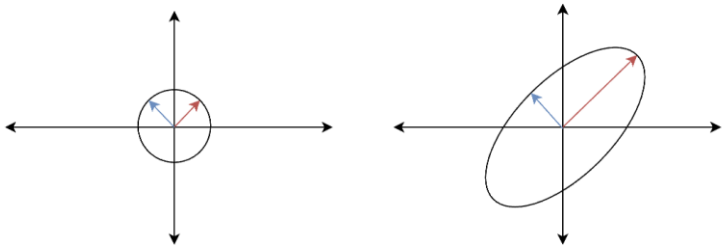

Ilustración 4 - Ejemplo de la diferencia de magnitud entre las direcciones principales en dos dimensiones.

Resulta evidente que, en este caso, el vector rojo va a ser el primer componente principal, ya que es aquel que representa mayor variabilidad, mientras que el vector azul es el segundo componente principal.

Ahora, ¿cómo se representan los puntos sobre la recta? (Ilustración 6) Pues resulta que los puntos se proyectan de forma perpendicular sobre la recta roja, De esta forma, si queremos llegar a proyectar los puntos por ejemplo sobre una única dimensión y pasar de R^2 a R^1, encontraremos que nos queda el siguiente esquema:

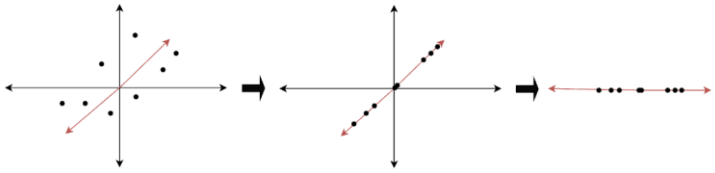

Ilustración 5 - Ejemplo de la transformación de datos con dos variables a una única componente tras el análisis PCA.

El número de variables de los datos que queramos reducir va a depender en cierta manera de la cantidad de información y calidad que estemos dispuestos a perder, en algunos casos, reducir la dimensionalidad supondrá una gran pérdida de información, mientras que, en otros, casi no veremos esta diferencia. Esto siempre dependerá de la varianza que expliquen los componentes principales que estemos aplicando, donde a mayor varianza, mejor podremos clasificar posteriormente los resultados (Ilustración 7).

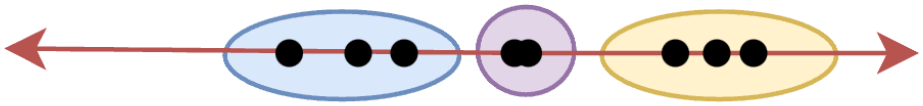

Ilustración 6 - Ejemplo Clustering de datos tras PCA.

Ejemplo de análisis de componentes principales:

Como ejemplo vamos a aplicar un caso muy simple que nos va a ayudar a comprender el impacto de este algoritmo.

Supongamos el caso de cuatro empresas para las cuales tendremos tres parámetros que conocemos, sus ingresos anuales en miles de euros, sus gastos anuales en miles de euros y el grado de satisfacción de los clientes en valores que toman del 0 al 100.

Debido a que los datos son muy similares, hacer una diferenciación clara con pocos componentes principales va a ser algo complicado. Por lo que suponemos inicialmente que la pérdida de información de los datos será grande, como en el ejemplo de las figuras anteriormente mencionado.

Disponemos de los siguientes datos:

	Ingresos	Gastos	Satisfacción
Empresa 1	40	50	60
Empresa 2	50	70	60
Empresa 3	80	70	90
Empresa 4	50	60	80

Tabla 1 - Datos de ejemplo

Para ello lo primero que debemos de hacer es calcular la matriz de covarianza. Necesitamos multiplicar X^T*X:

X^T * X		
9600	9900	11700
9900	11250	12400
11700	12400	16200

*Tabla 2 - Tabla X^T * X*

Ahora dividimos cada uno de estos entre el número de observaciones n = 4.

Cov(x) = (X^T * X) / n		
2400	2475	2925
2475	2812,5	3100
2925	3100	4050

Tabla 3 - Matriz de covarianza

El siguiente paso es calcular los auto vectores, para ello buscamos los valores resolviendo el polinomio característico de la matriz de covarianza e igualándolo a 0.

2400 - λ	2475	2925
2475	2812,5 - λ	3100
2925	3100	4050 - λ

Tabla 4 - Polinomio característico de la matriz de covarianza

De aquí obtenemos que los autovalores son:

λ1 = 520,099

λ2 = 78,105

λ3 = 18,463

Lo siguiente, es calcular los auto vectores, la fórmula es $(cov(x) - \lambda * \mathrm{I}) * v = 0$.

Aplicando los datos anteriormente calculados, obtenemos los auto vectores:

0.745	-0.231	0.626
0.285	-0.738	-0.612
0.603	0.634	-0.484

Tabla 5 - Autovectores

Los valores iniciales sobre los componentes principales se obtienen multiplicando los datos originales por los vectores propios.

Para proyectar los datos originales sobre el primer componente principal (PC1), que corresponde al vector propio v1, puedes usar la siguiente fórmula:

$$PC1 = X * v_1$$

Ecuación 7 - Proyección sobre la componente

De manera similar, podemos obtener las proyecciones sobre los otros componentes principales (PC2, PC3) multiplicando los datos originales por los vectores propios correspondientes (v2, v3):

$$PC2 = X * v_2$$

$$PC3 = X * v_3$$

Así, obtenemos la tabla con las proyecciones sobre los primeros componentes principales:

PC1	PC2	PC3
-22.276	4.765	4.308
-9.128	-12.304	-1.673
31.317	-0.218	2.591
0.087	7.757	-5.227

Tabla 6 - Proyección de los datos

En este caso, vamos a proyectar sobre la primera componente principal, de la cual obtenemos los siguientes resultados, una de las empresas es la que encabeza una ventaja competitiva frente al resto de una forma clara, la empresa más cercana al punto (0,0), podríamos determinar que es una empresa que se encuentra sobre la media y las dos últimas empresas estarían generando un rendimiento por debajo de la media.

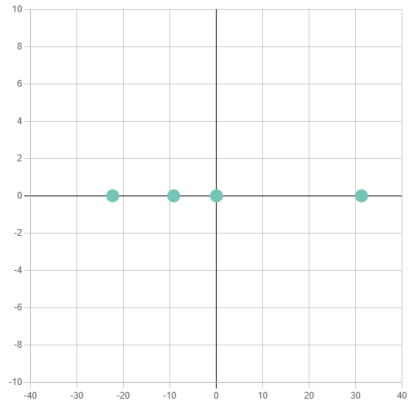

Ilustración 7 - Imagen Propia del proyecto web

En el caso de aplicar las dos primeras componentes principales, veríamos unos resultados similares, salvo que las dos empresas finales, ahora sí que se ve una gran diferencia sobre la segunda proyección entre ellas.

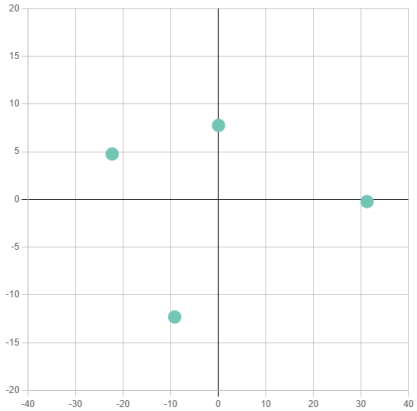

Ilustración 8 - Imagen Propia del proyecto web

3.2. Algoritmos de Clustering

3.2.1. Clustering

El Clustering, también conocido como análisis de grupos o agrupamiento, es la tarea encargada de agrupar objetos en función de su similitud. De esta forma, podemos decir que los miembros de cada grupo tienen características similares.

El análisis de grupos estuvo originado en antropología por Driver y Kroeber en 1932 e introducido a psicología por Zubin en 1938.

Esta técnica es comúnmente aplicada en el análisis de datos estadísticos. Existen infinidad de algoritmos de agrupamiento, donde por lo general, varían en la forma en la que el al autor considera qué es un grupo y cómo se realizan.

Por esto mismo, en este proyecto estudiaremos únicamente 3 formas de Clustering, pese a los miles que existen. Las 3 variaciones son: DBSCAN, OPTICS, K-means.

Podemos agrupar todas las técnicas de Clustering principalmente en 3:

Agrupamiento basado en conectividad (agrupamiento jerárquico):

El agrupamiento basado en conectividad, también conocido como agrupamiento jerárquico, está basado en la idea principal de que los objetos más cercanos están más relacionados que los que están alejados.

De esta forma, los grupos se distribuyen por distancias. Es un concepto sencillo de entender y de aplicar. El principal problema de este tipo de algoritmos es que se vuelven muy lentos cuando se empiezan a manejar cantidades grandes de datos.

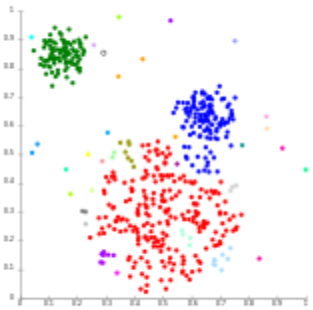

Ilustración 9 - Enlace simple en datos Gaussianos. En 35 grupos, al principio el grupo más grande se fragmenta en grupos más pequeños, mientras que todavía está conectado al segundo mayor por el efecto de enlace simple.

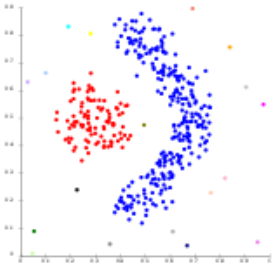

Ilustración 10 - Enlace simple en agrupamiento basado en densidad. Se extrajeron 20 grupos, la mayoría contienen un único elemento, nos podemos percatar entonces que enlace simple no tiene una noción de ruido.

Agrupamiento basado en centroide

El agrupamiento basado en centroide es un algoritmo de aprendizaje no supervisado que se utiliza para clasificar y agrupar datos en categorías o grupos similares. El proceso comienza con la selección de un número predeterminado de centroides, que son puntos representativos del conjunto de datos. Luego, se asignan los puntos de datos a los centroides más cercanos basándose en alguna medida de distancia, como la distancia euclidiana. Los puntos de datos asignados a un centroide particular forman un grupo o clúster.

Una vez que se ha realizado la asignación inicial de los datos a los centroides, se actualizan los centroides recalculando su posición en función de los puntos de datos asignados a ellos. Este proceso de asignación y actualización de centroides se repite iterativamente hasta que los centroides convergen a posiciones estables y los clústeres se vuelven estacionarios.

El agrupamiento basado en centroide es ampliamente utilizado en diversas aplicaciones, como segmentación de clientes, clasificación de documentos, análisis de imagen y reconocimiento de patrones. Sin embargo, también tiene algunas limitaciones, como la sensibilidad a la selección inicial de los centroides y la necesidad de especificar el número de clústeres de antemano.

Ejemplos con k-means

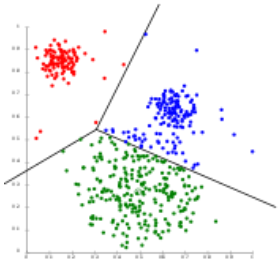

Ilustración 11 – Cómo K-means separa los datos

Agrupamiento basado en distribuciones

El agrupamiento basado en distribuciones es un enfoque de aprendizaje no supervisado que se utiliza para identificar patrones o estructuras en datos mediante la modelización de las distribuciones de los datos. Este método implica la suposición de que los datos en un conjunto de datos se distribuyen en diferentes grupos o clústeres, y busca estimar las distribuciones subyacentes a cada uno de estos clústeres.

El proceso comienza con la selección de un modelo de distribución adecuado para representar los datos, como la distribución gaussiana o la distribución de Poisson. Luego, se ajusta el modelo a los datos observados para estimar los parámetros de la distribución. Estos parámetros representan las características estadísticas de los clústeres, como la media y la varianza.

Una vez que se ha ajustado el modelo de distribución, se utiliza para asignar nuevos puntos de datos a los clústeres correspondientes. Esto se hace mediante la estimación de la probabilidad de que un punto de dato pertenezca a cada una de las distribuciones modeladas. Los puntos de datos con una probabilidad alta de pertenecer a una distribución en particular se asignan a ese clúster (Ilustración 14).

El agrupamiento basado en distribuciones se utiliza en varias aplicaciones, como clasificación de imágenes, detección de anomalías, y reconocimiento de patrones en datos biológicos. Sin embargo, también tiene algunas limitaciones, como la necesidad de seleccionar un modelo de distribución apropiado y la sensibilidad a la calidad de los datos observados (Ilustración 15).

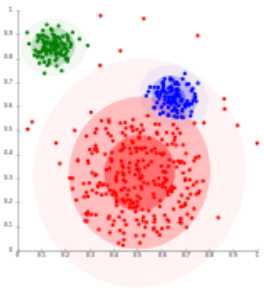

Ilustración 12 - En datos distribuidos con Gaussianas, EM trabaja bien, desde entonces se utilizan Gaussianas para la modelación de grupos.

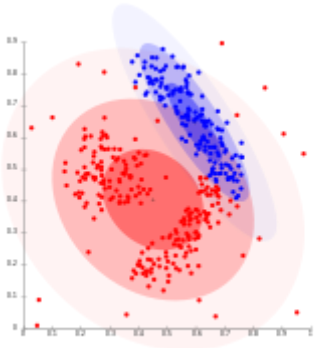

Ilustración 13 - Grupos basados en densidad no pueden ser modelados utilizando distribuciones Gaussianas

3.2.2. K-means

K-means es un algoritmo propuesto por Stuart Lloyd en 1957. Es un algoritmo de clasificación no supervisada, es decir, el algoritmo se ajusta a las observaciones, que agrupa los objetos en k grupos, en función de sus características.

El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o clúster. Se suele usar la distancia cuadrática, que es el caso de uso dentro del proyecto.

El algoritmo consta de tres pasos característicos.

1. Definición del valor k, el parámetro k nos va a indicar el número de grupos o clústeres que se deben de buscar dentro de los objetos.
2. Asignación de centroides, se establecen k puntos aleatorios como centroides y los diferentes objetos se asignan en función de la distancia a ellos.

   Fórmula:

$$S_i^{(t)} = \{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \}$$

Ecuación 8 - Distancia de K-means

3. Actualización de centroides, se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo.

   Fórmula:

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

Ecuación 9 - Posición promedio K-means

Los pasos 2 y 3 se van a repetir continuamente, hasta que los centroides no cambien. Pese a esto, el algoritmo nunca se compromete a encontrar los centroides óptimos ya que en la asignación se genera un pequeño margen de error que puede comprometer a esta optimización.

Se puede describir K-means como un problema de optimización, donde se busca minimizar las distancias entre los puntos y los centroides.

Ventajas y desventajas de usar K-means frente a otros algoritmos de clustering:

Ventajas de K-means:

- Simple y fácil de implementar.
- Eficiente en grandes conjuntos de datos.
- Resultados interpretables.
- Flexibilidad en la elección de distancia y métrica.
- Escalabilidad.

Desventajas de K-means:

- Requiere especificar el número de clústeres de antemano.
- Sensible a los valores iniciales.
- No es adecuado para clústeres de formas irregulares o tamaños diferentes.
- Sensible a valores atípicos.
- Sensible a la escala de los datos.

3.2.3. DBSCAN

BSCAN es el algoritmo de clustering basado en densidad con ruido más conocido. Este algoritmo fue propuesto por Martin Ester, Hans-Peter Kriegel, Jörg Sander y Xiaowei Xu en 1996.

Para la utilización de este algoritmo se necesita tener dos parámetros de antemano, la distancia máxima y el número de puntos mínimos que se deben de encontrar dentro de esa distancia para poder formar un clustering.

Por lo tanto, podemos diferenciar los puntos entre alcanzables o ruido (outlier).

1. Un punto p es un punto núcleo si al menos minPts puntos están a una distancia ε de él y, esos puntos son directamente alcanzables desde p. No es posible tener puntos directamente alcanzables desde un punto que no sea un núcleo.

2. Un punto q es alcanzable desde p si existe una secuencia de puntos $p_1 \dots p_n$ donde $p_1 = p \; y \; p_n = q$ tal que cada punto p_{i+1} es directamente alcanzable desde p_i; es decir, todos los puntos de la secuencia deben ser puntos núcleos, con la posible excepción de q.

3. Un punto que no sea alcanzable desde cualquier otro punto es considerado ruido.

Si p es un punto núcleo, esta forma un clúster junto a otros puntos (núcleo o no) que sean alcanzables desde él. Cada clúster contiene al menos un punto núcleo. Los puntos no núcleos alcanzables pueden pertenecer a un clúster, pero actúan como una barrera puesto que no es posible alcanzar más puntos desde estos.

Un clúster, satisface por lo tanto dos propiedades:

1. Todos los puntos del clúster están densamente conectados entre sí.

2. Si un punto A es densamente alcanzable desde cualquier otro punto B del clúster, entonces A también forma parte del clúster.

Usar DBSCAN frente a otros algoritmos tiene sus ventajas y desventajas:

Ventajas de DBSCAN:

- No requiere la especificación del número de clústeres de antemano.
- Capacidad para detectar clústeres de cualquier forma.
- Robusto frente al ruido y datos atípicos.
- Eficiente en términos computacionales, especialmente para conjuntos de datos grandes.
- No se ve afectado por la inicialización.

Desventajas de DBSCAN:

- Sensible a la configuración de parámetros.
- No es adecuado para conjuntos de datos de alta dimensionalidad.
- Sensible a la densidad de los clústeres.
- Dificultad para manejar diferentes tamaños de clústeres.
- No adecuado para conjuntos de datos con distribución no uniforme.

3.2.4. OPTICS

OPTICS es un algoritmo muy similar a DBSCAN que nace como solución a uno de los mayores problemas de DBSCAN, los grupos de densidad variable. Estos son agrupamientos de puntos cuya frecuencia de densidad es mayor o mejor al resto de conjuntos de puntos.

Al igual que DBSCAN, se requieren dos parámetros iniciales para el correcto funcionamiento, el primero que describe el radio máximo al que un punto puede alcanzar a otros y el segundo, el número mínimo de puntos requeridos para formar un clúster.

OPTICS busca todas las distancias vecinas dentro de la Distancia de búsqueda especificada, comparando cada una de estas distancias con la distancia de núcleo. Si cualquier distancia es menor que la distancia de núcleo se asigna a la entidad esa distancia de núcleo como su distancia de alcanzabilidad. Si todas las distancias son mayores que la distancia de núcleo, la más pequeña de estas distancias se asigna como distancia de alcanzabilidad. Si no hay más puntos dentro de la distancia de búsqueda, el proceso se reinicia en un nuevo punto que no se ha visitado anteriormente. En cada iteración, las distancias de alcanzabilidad se recalculan y se ordenan. La menor de las distancias se utiliza para la distancia de alcanzabilidad final de cada punto (Ilustración 16).

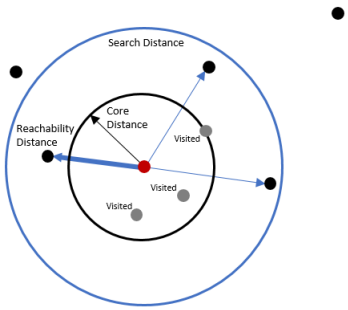

Ilustración 14 - Funcionamiento de OPTICS.

Ventajas y desventajas de usar OPTICS frente a otros algoritmos de clustering:

Ventajas de OPTICS:

- No requiere la especificación del número de clústeres de antemano.
- Capacidad para detectar clústeres de cualquier forma y tamaño.
- Puede identificar automáticamente la estructura de densidad variable.
- Permite la identificación de puntos de ruido y datos atípicos.
- Flexibilidad en la configuración de parámetros.

Desventajas de OPTICS:

- Sensible a la configuración de parámetros.
- Mayor complejidad computacional.
- Puede generar una gran cantidad de clústeres y/o ruido.
- No adecuado para conjuntos de datos de alta dimensionalidad.
- Sensible a la escala de los datos.

3.3. Distancias

En el contexto del clustering de datos, las distancias son medidas utilizadas para calcular la similitud o disimilitud entre pares de datos.

Las distancias juegan un papel crucial en la determinación de la similitud o disimilitud entre los datos, lo cual a su vez afecta la forma en que se agrupan en clústeres.

3.3.1. Mahalanobis

La distancia de Mahalanobis es una medida de distancia utilizada para calcular la similitud o disimilitud entre dos puntos en un espacio multidimensional, teniendo en cuenta la correlación y las varianzas de las variables. A diferencia de otras medidas de distancia, como la distancia Euclidiana o la distancia de Manhattan, la distancia de Mahalanobis tiene en cuenta la estructura de covarianza de los datos, lo que la hace adecuada para datos con correlaciones entre variables.

El uso más común de la distancia de Mahalanobis es encontrar valores atípicos multivariados, lo que indica combinaciones inusuales de dos o más variables.

Definición formal:

La distancia de Mahalanobis entre dos objetos se define como:

$$d = [(X_B - X_A)^T * C^{-1} * (X_B - X_A)]^{0.5}$$

Ecuación 10 - Distancia de Mahalanobis

Donde:
X_A y X_B es un par de objetos, y $\boldsymbol{C}$ es la matriz de covarianza de la muestra.

3.3.2. Euclídea

La distancia euclídea, es una medida de la distancia entre dos puntos en un espacio euclidiano de cualquier número de dimensiones.

La definición normal de la distancia euclidiana entre dos puntos, P y Q, en un espacio euclidiano n-dimensional, se calcula utilizando el teorema de Pitágoras. La fórmula es la siguiente:

$$dist(P,Q) = sqrt((x_2 - x_1)^2 + (y_2 - y_1)^2 + \cdots + (z_n - z_{n-1})^2$$

Ecuación 11 - Distancia euclídea

donde:

$P = (x_1, y_1, \ldots, n_1)$ es el primer punto en el espacio n-dimensional

$Q = (x_2, y_2, \ldots, n_2)$ es el segundo punto en el espacio n-dimensional

Esta fórmula se puede generalizar a cualquier número de dimensiones y se utiliza comúnmente en matemáticas, ciencias de la computación, física y otras áreas donde se necesita medir la distancia entre puntos en un espacio n-dimensional.

3.3.3. Euclídea normalizada

La distancia euclidiana normalizada se refiere a la distancia euclidiana entre dos puntos que se ha normalizado o escalado para que los valores de las diferentes dimensiones estén en una escala común. La normalización se realiza para evitar que las dimensiones con valores grandes dominen las dimensiones con valores pequeños en el cálculo de la distancia.

La normalización de la distancia euclidiana se realiza dividiendo cada diferencia de coordenadas entre los dos puntos por el rango de valores de esa dimensión. El rango se define como la diferencia entre el valor máximo y el valor mínimo de la dimensión correspondiente.

La fórmula de la distancia euclidiana normalizada entre dos puntos, P y Q se calcula de la siguiente manera:

$$dist(P,Q) = sqrt((x_2 - x_1)^2 + (y_2 - y_1)^2 + \cdots + (z_n - z_{n-1})^2$$

Ecuación 12 - Distancia euclídea normalizada

Como podemos observar, es idéntica al cálculo de la distancia euclídea, solo que aplicando un conjunto de datos normalizados.

La distancia euclidiana normalizada se utiliza comúnmente en la minería de datos y el aprendizaje automático para comparar patrones o características de datos que tienen diferentes rangos o unidades de medida.

3.4. Outliers

La detección de anomalías (outliers) es una técnica de minería de datos que permite el reconocimiento de nuevos patrones con comportamientos inusuales sobre un conjunto de datos.

Estos datos se pueden traducir como errores de cálculo, acciones no válidas o anómalas sobre los datos.

A lo largo de la historia de los algoritmos de detección de anomalías, se ha decretado que la mejor forma de generar estos datos es mediante técnicas de aprendizaje profundo (deep learning), es decir, aquellas que son capaces de aprender de sí mismas, a partir de su propia experiencia.

Estos tipos de mecanismos tienen un conjunto de campos donde sus aplicaciones son las más utilizadas. Estamos hablando de aplicaciones médicas, para detección de enfermedades por diferentes patologías correlacionadas entre sí. En el procesamiento de imágenes médicas, un conjunto de pixeles anómalos, pueden ayudar a detectar ciertos tipos de tumores o cáncer, etc (Ilustración 17).

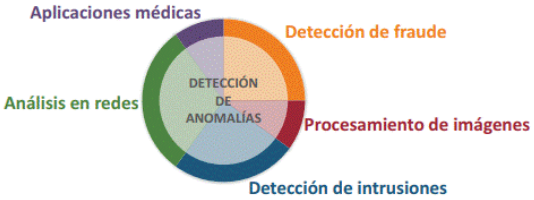

Ilustración 15 - Principales aplicaciones de outliers

En este caso nos situamos ante la necesidad de querer obtener las anomalías que se generan sobre un conjunto de datos de x dimensionalidades, para ello, lo mejor es trabajar directamente con las distancias entre los diferentes objetos. Debido a ello, se han incorporado los siguientes 5 algoritmos para detectar anomalías.

3.4.1. Mahalanobis

Como ya tratamos previamente en el apartado 3.3.1. Mahalanobis es un tipo de distancia entre puntos que aplica dos puntos y un espacio multivariado, por lo que las distancias dependen no solo de los dos puntos, sino de la matriz de covarianza de todos los puntos de puntos del espacio.

De esta forma, podemos obtener una matriz de distancias de Mahalanobis entre los diferentes puntos y, mediante un umbral definido por el usuario como distancia máxima, encontrar aquellos puntos que superen este parámetro y detectarlos como anomalías.

3.4.2. DBSCAN

DBSCAN es otro algoritmo del que ya hemos hablado más en detalle en el apartado 3.2.3., este algoritmo está basado en la densidad de aplicaciones con ruido. Pese a que este algoritmo no fue diseñado para la detección de outliers, sino más bien para el agrupamiento de datos, se pueden analizar los resultados de forma opuesta y obtener estos outliers.

Como ya sabemos, para la aplicación de este algoritmo basado en distancias, necesitamos dos parámetros de entrada, un número mínimo de puntos para determinar un clúster y una distancia máxima a partir de la cual los puntos que estén dentro puedan formar parte de ese clúster.

De esta forma, todos aquellos puntos que no hayan sido asignados a ningún clúster se van a definir como outliers.

3.4.3. OPTICS

OPTICS es otro algoritmo ya mencionado en el apartado 3.2.4. Como ya se habló, este algoritmo es una derivación de DBSCAN que busca mejorar la eficiencia con un ordenamiento de las distancias.

Para su implementación se definen dos parámetros de entrada, el umbral distancia máxima entre puntos y el número mínimo de puntos para definir un clúster.

De esta forma, al igual que DBSCAN, todos aquellos puntos que no se encuentren agrupados dentro de ningún clúster, se denominarán outliers.

3.4.4. K-nearest neighbors

KNN es un algoritmo empleado para la clasificación y regresión de datos. Este algoritmo no fue diseñado para la detección de outliers. El enfoque de KNN para la detección de outliers se basa en la idea de que los outliers estarán alejados de sus vecinos más cercanos en el espacio de características. Es importante normalizar los datos a utilizar para mejorar la precisión de este algoritmo.

Para el cálculo de outliers se aplican los siguientes pasos:

1. Cálculo de distancias: Para cada punto en el conjunto de datos, se calcula su distancia a todos los demás puntos utilizando una métrica de distancia apropiada, como la distancia euclidiana. Esto implica calcular la distancia entre cada par de puntos en el conjunto de datos.

2. Definición del umbral: Se define un parámetro k para determinar cuántos puntos se van a considerar outliers.

3. Etiquetado de outliers: Se ordenan las distancias calculadas de menor a mayor y cada punto se etiqueta como un outlier si su posición coincide con los k ultimas distancias ordenadas, obteniendo de esta forma los puntos más alejados del conjunto de datos.

4. Análisis de resultados: Se analizan los puntos etiquetados como outliers.

3.4.5. LOF

El algoritmo LOF (Local Outlier Factor) es un método de detección de outliers basado en densidad que proporciona una medida de anormalidad para cada punto en un conjunto de datos. A diferencia de otros enfoques que solo consideran la distancia a los puntos vecinos más cercanos, LOF tiene en cuenta la densidad local de los puntos y la comparación con la densidad de sus vecinos.

Para el cálculo de outliers se aplican los siguientes pasos:

1. Cálculo de la distancia: Para cada punto en el conjunto de datos, se calcula su distancia a todos los demás puntos utilizando una métrica de distancia apropiada, como la distancia euclidiana. Esto implica calcular la distancia entre cada par de puntos en el conjunto de datos.

2. Cálculo de la densidad local: Para cada punto, se determina la densidad local considerando el número de puntos dentro de un radio predefinido alrededor de ese punto. Esto se puede hacer utilizando el método KNN (K-Nearest Neighbors), donde se cuentan los puntos vecinos dentro del radio.

3. Cálculo del factor de anormalidad LOF: El LOF de un punto se calcula comparando su densidad local con la densidad local de sus vecinos. El LOF se define como la relación entre la densidad local del punto y la densidad local promedio de sus vecinos. Un LOF superior a 1 indica que el punto tiene una densidad local más baja en comparación con sus vecinos, lo que sugiere que puede ser un outlier.

4. Etiquetado de outliers: Los puntos se etiquetan como outliers en función de sus valores de LOF. Puntos con un LOF significativamente mayor que 1 se consideran outliers, mientras que puntos con un LOF cercano a 1 se consideran datos normales.

4. Técnicas y herramientas

4.1. Metodología de gestión de proyectos

En el proyecto hemos decidido utilizar una metodología ágil, que nos proporcionara una vista más rápida de las partes que se prevén hacer y sus correspondientes plazos.

Podemos definir metodología ágil como una forma de pensar en los flujos de trabajo y la colaboración, definiendo una serie de valores que guíen a nuestras decisiones con respecto a lo que vamos haciendo y a la forma en la que se hace.

En definitiva, las metodologías ágiles de desarrollo de software buscan crear en poco tiempo de software que funcione.

Esta metodología se basa en la valoración de cuatro características fundamentales:

- Las personas y las interacciones, antes que los procesos y las herramientas.
- El software en funcionamiento, antes que la documentación exhaustiva.
- La colaboración con el cliente, antes que la negociación contractual.
- La respuesta ante el cambio, antes que el apego a un plan.

En este caso, vamos a escoger el marco ágil para el desarrollo del software denominado SCRUM. Este marco es el más conocido y popular en la actualidad.

En el modelo SCRUM, se dividen los espacios temporales en Sprints. Un Sprint es un periodo de tiempo, generalmente calculado en semanas, en donde se proponen fragmentos del software para desarrollar y un tiempo límite para realizarlo.

La pila de sprint o sprint backlog es una lista de todas las tareas requeridas necesarias para crear historias de usuario para ser ejecutadas en un sprint. Las historias de usuario se dividen en módulos de tamaño adecuado para supervisar el progreso a diario e identificar sin esfuerzo riesgos y problemas sin un proceso de gestión complejo.

Para definir estas tareas a realizar en un sprint, se realizan reuniones de planificación del sprint, las cuales tienen lugar al inicio del sprint. En estas reuniones, se determinan cuáles y cómo van a ser las funcionalidades que se van a incorporar al producto al terminar el sprint. Por otra parte, a cada una de las tareas se les asigna un número, generalmente uno de la sucesión de Fibonacci, comprendido entre el 1 y el 21, donde se indica la dificultad y el tiempo que va a llevar realizar esta tarea (menor 1 y mayor 21).

Por otra parte, existen las reuniones de revisión del sprint, en donde el equipo de trabajo se junta durante una o dos horas al final de cada sprint, para enseñar las mejoras y recoger sugerencias.

<u>Kanban:</u>

Kanban es una técnica para gestionar de forma clara y visual un flujo continuo de avance. Se compone de un tablero con varias columnas, cada una con un estado diferente (New issues, Product Backlog, Sprint Backlog, In Progress, Review, Done, Closed, etc.) donde se van a anotar las diferentes tareas a realizar durante un sprint y su estado actual.

Para realizar este modelo vamos a utilizar la extensión de GitHub, **ZenHub**, la cual nos permite visualizar el tablero, directamente desde el repositorio del proyecto, crear los sprints y cada una de las tareas que lo va a componer.

Tras usar durante semanas ZenHub, ha surgido el inconveniente de que únicamente era una versión de prueba y continuar usando supone un coste de $12.50 al mes. Por esto, he decidido migrar el control de sprints a la herramienta Trello.

Trello es un software de administración de proyectos con interfaz web y con cliente para las plataformas de iOS y Android.

Surgió en 2010 como un proyecto de Federico Stella. Es básicamente una herramienta de organización de actividades con forma de tarjetas visuales. Permite agregar listas, adjuntar archivos, etiquetar eventos, agregar comentarios y compartir tableros.

4.2. Editor de texto para la memoria

Como editor de texto he decidido emplear desde un primer momento Microsoft Word. Se me propuso la idea de hacer este documento sobre LaTeX, pero dado que considero que mi manejo de Word es avanzado y que trabajar con LaTeX implicaría tener que aprender de cero esta herramienta, finalmente he decidido que Microsoft Word es la opción más conveniente para documentar el proyecto.

4.3. Referencias bibliográficas, citas y plagio.

Para esta parte del trabajo surgió la duda entre Mendeley o Zotero como software para el almacenaje, organización y compartición de referencias bibliográficas del proyecto.

Ambos softwares son gratis, pero estas son algunas de las características que me hicieron decantarme por Mendeley:

Mendeley	Zotero
Permite compartir las referencias con otros usuarios mediante grupos.	No permite compartir las referencias con otros usuarios mediante grupos.
2GB de almacenamiento en la nube.	300MB de almacenamiento en la nube.
7000 estilos de citación CSL gratis.	7000 estilos de citación CSL gratis.
Se pueden añadir notas a los PDFs guardados.	No se pueden añadir notas a los PDFs guardados.
No es open source.	Es open source.
Compatible con todos los navegadores.	Versiones diferentes para Firefox, que para otros navegadores (Zotero Standalone).

Tabla 7 - Comparativa entre Mendeley y Zotero

Por otra parte, escogí el software de Mendeley, porque los tutores me enseñaron primeramente de su existencia y me demostraron un pequeño inicio en su manejo, donde vi su pequeña complejidad de aprendizaje, porque la propia universidad ofrece cursos de UbuAbierta para comprender y facilitar el manejo de este software.

Por otra parte, posee una interfaz muy amigable y fácil de usar, gracias a la cual es muy cómodo aprender su uso y manejo.

Mendeley cuenta con un plug-in para navegadores (Mendeley Web Importer), el cual hace que en un par de clicks tengas tu página web, libro o cualquier tipo de referencia guardada en tu almacenamiento personal de Mendeley.

4.4. Repositorios de código

Para esta parte del trabajo se me valoraron muchas opciones, pero principalmente por comodidad y siendo con el que he trabajado durante todos los años de carrera, Github ha sido el tipo de repositorio que he escogido sin ningún tipo de duda.

GitHub es una plataforma de desarrollo colaborativo que se utiliza principalmente como control de versiones Git. Actualmente GitHub se considera la plataforma más importante de repositorios de código abierto.

GitHub fue fundado el 8 de febrero de 2008 y desde 2018 es propiedad de Microsoft. Esta adquisición por parte de una empresa tan prominente tiene ventajas significativas. En primer lugar, contar con el respaldo de Microsoft implica una mejor integración con otras herramientas como Visual Studio Code, lo cual evita muchos problemas y facilita la conectividad. En segundo lugar, al formar parte de esta gran empresa, GitHub goza de un amplio respaldo de la comunidad, lo que significa que siempre podrás recibir ayuda sin dificultad en cualquier momento.

4.5. Herramientas de prototipado

En el tema de la elección de herramientas de prototipado tenía varias opciones Adobe XD, Pencil, Figma, etc.

Al final, tras investigar un poco sobre las limitaciones y capacidades de cada una, decidí escoger la herramienta de Pencil.

AdobeXD, también conocido como Adobe Experience Design, es un software que como su propio nombre indica, fue creado para dar al usuario una experiencia única a la hora de diseñar. El principal problema de este software es que, pese a que previamente era gratis y accesible, ahora tienes la necesidad de subscribirte al plan de Adobe Creative Cloud para poder utilizarlo como herramienta de trabajo.

Por otra parte, tenemos Figma, un programa muy completo, capaz de realizar diseños ultra realistas y con una gran capacidad de componentes y estilos. Es una aplicación que se utiliza en línea, lo que la hace muy portable y no tienes la necesidad de instalar programas ni librerías. Este software se podría decir que fue una de las principales ideas para utilizar, ya que tanto su interfaz, como la experiencia de usuario está hecha para aprender de forma muy sencilla y dar rienda suelta a tu imaginación.

Pero tras hablar con varias personas sobre qué software es mejor, llegamos a la conclusión que hasta una servilleta de papel te sirve como elemento para poder diseñar interfaces de una forma rápida y sencilla.

Ahí es donde entra Pencil, una herramienta que ya he manejado durante la carrera y que es tan simple que cualquier persona podría usarla. La principal pega de este software es que viene con unas librerías o paquetes de componentes muy antiguos, donde podemos encontrar componentes con la apariencia de dispositivos de hace 10 o 15 años. La solución más recomendada es buscar un paquete de diseños, ya sean de lo opcionales que ellos te proporcionan, o de los miles que hay por internet. Estos estilos le dan un soplo de aire fresco y actual a tu proyecto.

Es por esto por lo que he decidido utilizar la herramienta de Pencil para este proyecto: por su simpleza y su fácil manejo.

4.6. Software para hacer documentación automática

Como desconocía de algún software de documentación automática, decidí investigar un poco las dos proposiciones de mis tutores: GitBook y Read the Docs.

GitBook es una aplicación GUI multiplataforma que nació para crear una solución moderna y simple a la creación de la documentación, la escritura digital y la publicación de contenidos usando Markdown y Git/GitHub. Nos permite mostrar el contenido de nuestro repositorio como un sitio web o un PDF. El principal objetivo de GitBook es poder crear documentación que sea fácilmente editable y abierta a contribuciones.

Para poder utilizar GitBook, debemos realizar su instalación mediante el comando

$ npm install gitbook -g

Y podemos crear un libro fácilmente usando,

$ gitbook serve ./repository

Read the Docs, es un software dedicado a simplificar la documentación de software, mientras se crea el propio software, teniendo un control de versiones de forma automática.

Es un software Open Source y dedicado al usuario, ha sido usado en más de 100.000 proyectos, tanto grandes como pequeños, es comúnmente aceptado y aplicado en empresas de creación del software.

Tras profundizar y analizar ambas opciones, he decidido utilizar GitBook, ya que me parece una opción más completa y fácil de aplicar y utilizar. Por otra parte, la documentación es más simple, fácil de entender y por detrás tiene una comunidad mayor que Read the Docs, lo cual sugiere que, en la aparición de problemas o fallos al usar el software, será más sencillo de resolver.

4.7. Herramienta para el lanzamiento de la aplicación

Para poder lanzar la aplicación web de una forma gratuita y sin muchas complicaciones, he decidido utilizar la página web y servicios de Netlify.

Netlify te permite crear, alojar y mantener tu sitio web o aplicación con implementación continua, y de firma gratuita.

Esta ha sido la opción escogida, ya que fue la que mis tutores me mostraron como ejemplo y me ha parecido fácil de utilizar y de una gran ayuda.

Para utilizar esta herramienta, solo necesitas de un repositorio de GitHub, donde una vez que vayas realizando los diferentes commits en tu aplicación de desarrollo, Netlify se encargará automáticamente de compilar tu aplicación y generarte una URL para que puedas acceder a tu web desde cualquier sitio.

Sí que es cierto que he encontrado varios problemas a la hora de desplegar mi aplicación en Netlify y estos son algunos de ellos y sus soluciones:

1. Límite de tamaños de los archivos y de la caché. Netlify por defecto viene con una configuración que carga en tu proyecto donde establece unos límites para los tamaños de algunos elementos. La solución es muy simple y es que en el 'angular.json' se crea una propiedad donde se establecen estos límites de tamaños. Simplemente basta con aumentar estos tamaños (sin necesidad de que sean excesivos) y el error se solucionará.

Ilustración 16 - Tamaños de carga en Netlify

2. El siguiente problema que he tenid, es que, a la hora de compilar el código para ser subido, Netlify utiliza cypress, una librería que genera tests para el debug visual de la aplicación. Ellos directamente te dan la solución en su 'README.md', donde únicamente tienes que eliminar del fichero 'netlify.toml' las referencias de los plugins y desinstalar los plugins de netlify y cypress.
   "$ npm uninstall -D netlify-plugin-cypress".
   "$ npm uninstall cypress".

Los problemas a los que se enfrenta uno en estas situaciones tienen soluciones relativamente sencillas. Sin embargo, cuando se vuelven a presentar, se convierten en una fuente constante de preocupación, ya que es difícil discernir si el error reside en nuestra propia aplicación o en alguna biblioteca que estamos utilizando.

Por el resto, funciona de una forma fácil, qué más allá de estos dos errores, no me ha dado ningún problema. También te genera una serie de estadísticas, dándote notas sobre 100 de diferentes ámbitos de tu proyecto (Performance, Accesibility, Best Practies, SEO y PWA). También te proporciona consejos sobre qué puede estar ralentizando la aplicación y cómo solucionarlo.

4.8. Herramientas para la creación de código

En este caso, la herramienta para la creación de código que he decidido utilizar es Visual Studio Code (VSC), un editor de código fuente desarrollado por Microsoft para Windows, Linux, MacOS y web. Visual Studio Code fue presentado el 29 de abril de 2015 en la conferencia Build de 2015.

VSC ofrece funcionalidades como la capacidad de depuración, control integrado de Git, destacado de la sintaxis, completado inteligente de código, fragmentos y optimización del código.

El fundamento de Visual Studio Code se apoya en Electron, un marco de trabajo utilizado para crear aplicaciones de escritorio mediante la combinación de Chromium y Node.js. Esta combinación se ejecuta sobre el motor de diseño Blink.

Como el lenguaje a utilizar para este proyecto era Angular (HTML, SCSS y TS), el mejor IDE en mi opinión es VSC, debido a las siguientes razones:

1. Es un potente soporte para TypeScript, incluyendo resaltado de sintaxis, sugerencias de código, completado automático, refactoring y depuración, lo que facilita el desarrollo en Angular.
2. Integración con Angular CLI (Angular Command Line Interface), esto agiliza el proceso de construcción, generación de componentes, servicios, módulos, etc.
3. Amplia gama de extensiones.
4. Integración con Git y control de versiones.

4.9. Librerías del proyecto

4.9.1. XLSX

La librería xlsx en JavaScript es una herramienta poderosa que permite la manipulación y creación de archivos Excel (.xlsx) directamente en tu navegador.

Esta librería funciona mediante el uso de una serie de funciones y métodos que permiten la lectura, escritura y manipulación de datos en hojas de cálculo de Excel.

Al utilizar la librería xlsx en JavaScript, puedes cargar archivos .xlsx existentes, extraer y editar datos específicos, agregar nuevas hojas de cálculo, crear fórmulas, aplicar formato y guardar el archivo resultante en tu computadora.

Además, xlsx en JavaScript también te permite trabajar con archivos CSV y otros formatos de hojas de cálculo.

Principalmente he utilizado esta librería para la inserción de datos, evitando así que el usuario tenga que introducir todos los datos de forma manual cada vez que quisiera utilizar la aplicación y para, al finalizar todo el cálculo de los datos, poder guardarlos en una nueva hoja de cálculo.

4.9.2. Librerías para representar Gráficos

Para este caso, decidí probar dos librerías bastante utilizadas y recomendadas cuando necesitas dibujar un gráfico, ya sean por su infinidad de opciones y todos los diferentes tipos de gráficos que pueden llegar a dibujar ngx-charts y ng2-charts.

En mi caso, necesitaba dibujar un gráfico de puntos (scartt chart). Ambas librerías, me proporcionaban funciones similares, ya que ambas son una capa superpuesta de la librería base de gráficos chart.js.

ChartJS se define a sí misma como una librería simple, flexible y dedicada a hacer las webs más modernas. Actualmente está en su versión 4.0, donde podemos encontrar que se han centrado en el estilo y funcionalidad, en vez de realizar una implementación de nuevos tipos de gráficos.

La primera librería de la que hemos hablado, ngx-charts, es una de las librerías más conocidas a nivel global para la incorporación de gráficas a nuestro proyecto.

Ngx-charts utiliza Angular para renderizar y animar los elementos SVG con la velocidad que aplica este framework y utiliza d3 para el uso de funciones matemáticas, escalas, ejes y generadores de forma.

Uno de sus principales objetivos es hacer que la apariencia de los gráficos sea lo más estética posible y para ello incorpora infinidad de estilos preestablecidos y te da la oportunidad de, mediante el uso del archivo CSS de estilos, poder definir los tuyos propios.

Para el uso de esta librería, se necesita instalar mediante npm con el siguiente comando:

“$ npm i @swimlane/ngx-chart”.

La segunda librería de la que hemos hablado es ng2-charts, una librería open-source, que utiliza HTML5 y actualmente se encuentra en la versión 4.1.1. Esta librería también te permite todo tipo de personalización de los gráficos a tu gusto.

Ng2-charts se instala mediante el uso de npm con el comando:

“$ npm install ng2-charts –save”.

Como anteriormente he mencionado, utiliza chart.js por debajo por lo que también debemos de instalarlo.

“$ npm install chart.js –save”.

4.9.3. Librerías de componentes:

Angular Material, es una librería gratuita de diseño de componentes. Su principal funcionamiento es describir tanto funcionalidad, utilizando JS y HTML, como estilos propios en CSS. En este caso es la librería que he utilizado para este proyecto.

Esta librería de componentes busca tener una alta calidad, versatilidad y compatibilidad con otras librerías. Todos sus componentes cumplen las funciones de internacionalización y accesibilidad. Y está apoyada por una gran comunidad de usuarios que ayudan a que evolucione a grandes pasos.

Ahora mismo, se encuentra en su versión 15.2.3, ya que buscan la mayor compatibilidad con las últimas versiones de Angular.

Por otra parte, he encontrado otras librerías que me han parecido interesantes y las cuales me he planteado utilizar.

Ng-Bootsrtap. Esta librería utiliza Angular, Bootsrtap 5 y Popper. Es muy similar a lo que nos proporciona el propio Boostrap, salvo que añade algunas funcionalidades extra. Decidí no utilizar esta librería de componentes, ya que los estilos no me acababan de gustar, ya que son muy básicos y con colores monótonos.

MDB-Bootstrap Angular. Esta opción, es probablemente una de las librerías más completas que puedes incorporar a Angular, cuenta con más de 700 componentes diferentes y es utilizada por una gran cantidad de empresas para el desarrollo de sus aplicaciones web. El principal problema que he encontrado es que, pese a ser tan completa, una gran parte de las funcionalidades que se diferencian de otras librerías de componentes es de pago, lo cual hace que para este proyecto sea algo difícil de considerar.

Por otra parte, su documentación es muy detallada y completa, lo que ayuda al desarrollador a utilizarla de una manera fácil. Además, cuenta con un soporte técnico capaz de resolver cualquier problema o duda que pueda surgir.

Tailwind Elements es una librería de componentes de interfaz de usuario que se construye sobre Tailwind CSS. Esta librería incluye gran cantidad de componentes preconstruidos y personalizables. La mayor utilidad de Tailwind Elements es que puedes aplicar plantillas preconstruidas que puedes usar para crear una página web de forma completa de una manera simple. La principal pega de esta librería es que, como MDB-Bootstrap, una parte del material a utilizar es de pago, lo cual limita la experiencia del programador, ya que hay elementos que no te da esta librería.

Font Awesome es una librería de iconos que cuenta con 22.643 iconos diferentes. Su versión actual es la 6. Los iconos en FontAwesome se entregan como fuentes escalables, lo que significa que pueden ser escalados fácilmente sin perder calidad o nitidez. La biblioteca es compatible con muchos marcos y plataformas, como Bootstrap, React, Angular, Vue.js, iOS y Android.

4.9.4. PCA (pca-js y ml-pca)

El cálculo de componentes principales es uno de los puntos fuertes de este proyecto.

Para ello, he encontrado dos librerías que consiguen todo lo necesario para trabajar con el algoritmo.

La primer es pca-js, una librería para JavaScript que me ha ayudado principalmente a entender el algoritmo y su funcionamiento. Incorporé en un primer momento esta librería en el proyecto y la utilicé para hacer una primera aproximación al resultado final actual. Después de realizar un uso básico de la librería, me empezó a generar conflictos con otras librerías.

De ahí surge el uso de ml-pca para este proyecto, una librería más completa para el cálculo de componentes principales que también genera otros datos estadísticos como la varianza, los autovectores, autovalores, etc. De esta forma, he podido aplicar y aumentar los resultados obtenidos en el cálculo de componentes principales.

4.9.5. Html2canvas y jsPDF

Estas dos librerías se usan conjuntamente para transformar elementos HTML a PDF. Html2canvas se encarga de renderizar y capturar el código HTML en forma de HTML5, manteniendo las propiedades de estilos del SCSS. Por otra parte, jsPDF es la librería encargada de crear y manipular archivos PDF en el front. La lógica de estas dos librerías juntas es que permite insertar la imagen renderizada del HTML en un PDF que usaremos para guardar los diferentes resultados que obtendremos durante la ejecución del programa.

Gracias a esto, podemos generar documentos muy visuales que, en el caso de la aplicación, viene muy bien para la representación de los gráficos de puntos con sus correspondientes grupos en el apartado de clustering.

Html2canvas es una librería de gran utilidad, pero a la hora de manejar componentes muy grandes y transformarlos en imágenes, puede llegar a tardar varios segundos, lo cual compromete la experiencia de usuario ya que hace parecer lento al programa.

Para este problema, he intentado buscar mejores opciones para optimizar este proceso o implementar otras librerías que hagan esto mismo, pero no he encontrado nada mejor.

4.9.6. Math y math.js

Las librerías "math" y "math.js" son dos bibliotecas matemáticas ampliamente utilizadas en el desarrollo de software para realizar cálculos matemáticos complejos.

La librería "math" es una librería incorporada en muchos lenguajes de programación, como Python, JavaScript y Java. Proporciona una amplia gama de funciones matemáticas y constantes predefinidas que permiten realizar operaciones comunes, como sumas, restas, multiplicaciones, divisiones, exponenciaciones, raíces cuadradas, trigonometría y más. Estas funciones se pueden utilizar para resolver problemas matemáticos básicos y realizar cálculos en aplicaciones o scripts.

Por otro lado, "math.js" es una librería matemática de JavaScript de código abierto que amplía las capacidades matemáticas más allá de lo que se ofrece en la librería "math" estándar.

Principalmente, estas dos librerías se han aplicado en el proyecto, en conjunto con la librería Matrix, para hacer cálculos sobre matrices, ya sean en los algoritmos de clustering o de cálculo de distancias.

4.9.7. Rxjs

RxJS (Reactive Extensions for JavaScript) es una biblioteca de programación reactiva ampliamente utilizada en TypeScript y JavaScript. Proporciona un conjunto de herramientas y patrones para trabajar con secuencias de eventos asíncronos, como eventos de usuario, peticiones HTTP y datos en tiempo real, entre otros. RxJS está basado en el patrón Observador-Observable, donde el Observador (subscriber) se suscribe a un Observable y recibe notificaciones cada vez que ocurre un evento.

Las principales funcionalidades de RxJS en TypeScript incluyen:

Observables: Los Observables son secuencias de eventos que pueden emitir valores de manera asíncrona. Pueden representar eventos únicos o flujos continuos de datos. Los Observables

ofrecen operadores para manipular, combinar y transformar los datos que emiten, como filtros, transformaciones, combinaciones, agrupaciones, entre otros.

Operadores: RxJS proporciona una amplia gama de operadores que permiten manipular y transformar los datos emitidos por los Observables. Estos operadores incluyen map, filter, reduce, merge, concat, debounce, distinct, entre muchos otros. Los operadores permiten realizar acciones como filtrar datos, transformarlos, agruparlos o combinarlos de diferentes maneras, facilitando la manipulación de flujos de eventos.

Suscripciones: Las suscripciones son utilizadas para recibir los datos emitidos por un Observable. Una vez que te suscribes a un Observable, puedes proporcionar una función de callback para manejar los valores emitidos, errores o notificaciones de completado. Además, las suscripciones se pueden cancelar para dejar de recibir datos del Observable, evitando fugas de memoria.

Manejo de errores: RxJS ofrece herramientas para manejar y controlar errores dentro de los Observables. Puedes utilizar operadores como catch, retry, throwError, etc., para capturar, manejar o lanzar errores dentro del flujo de eventos.

Schedulers: Los Schedulers son utilizados para controlar la concurrencia y el orden de ejecución en RxJS. Permiten especificar cómo se programan y ejecutan los eventos emitidos por los Observables, lo que puede ser útil para gestionar tareas asíncronas, evitar bloqueos o priorizar ciertas operaciones.

En resumen, RxJS en TypeScript proporciona una forma poderosa y expresiva de trabajar con secuencias de eventos asíncronos.

5. Aspectos relevantes del desarrollo del proyecto

En esta sección, se abordarán las diferentes decisiones tomadas durante el desarrollo de la aplicación, así como los puntos críticos y los desafíos encontrados a lo largo del proyecto.

Es importante destacar las diversas decisiones que se tomaron durante el desarrollo de la aplicación. Además, se analizarán los puntos críticos que surgieron y los problemas que se encontraron en la realización del proyecto.

1. Arquitectura del proyecto

Desde el inicio, se optó por desarrollar la aplicación utilizando Angular en un enfoque de frontend puro, sin necesidad de un backend ni una base de datos. Por lo tanto, la estructura y arquitectura del proyecto se basan en el patrón Modelo-Vista-Vista de Modelo (MVVM) de Angular, que se describe a continuación:

En lugar del patrón Modelo-Vista-Controlador (MVC), Angular utiliza el patrón MVVM, que presenta algunas diferencias importantes. En MVVM, el modelo sigue representando los datos y la lógica relacionada con ellos, pero la vista se encarga únicamente de la presentación y la interfaz de usuario. En lugar de tener un controlador, se introduce el concepto de la vista de modelo, que actúa como intermediario entre la vista y el modelo.

El modelo en MVVM se encarga de contener los datos y la lógica de negocio de la aplicación, al igual que en MVC. Sin embargo, el modelo también puede incluir propiedades y comandos específicos para la vista, lo que ayuda a desacoplar la vista del modelo subyacente.

La vista en MVVM se centra únicamente en la presentación de los datos y la interfaz de usuario. Utiliza enlaces de datos bidireccionales proporcionados por Angular para mostrar los datos del modelo en la vista y reflejar los cambios en el modelo cuando se realizan en la vista.

La vista de modelo actúa como un intermediario entre la vista y el modelo. Es responsable de proporcionar los datos y las acciones necesarias para que la vista funcione correctamente. La vista de modelo se comunica con el modelo para obtener y actualizar los datos, y expone propiedades y comandos que se enlazan con la vista.

Angular facilita la implementación del patrón MVVM a través de sus componentes y servicios. Los componentes en Angular representan tanto la vista como la vista de modelo, mientras que los servicios se utilizan para encapsular la lógica de negocio y la interacción con el modelo.

2. Configuración del entorno de desarrollo

Este apartado pretende explicar la instalación y configuración de Visual Studio Code para usar Angular dentro del entorno. Debido a la extensión, esto se recoge en el Anexo IV – Manual del programador.

3. Implementación del análisis de datos

En la implementación de los diversos algoritmos utilizados en la aplicación, se seleccionaron cuidadosamente una serie de bibliotecas npm disponibles con licencia gratuita. Además, se incorporaron algoritmos que, aunque no estaban incluidos en bibliotecas, se encontraban disponibles en la red a través de plataformas como Kaggle. En algunos casos, se implementaron algoritmos utilizando pseudocódigos como referencia.

De esta manera, se aprovechó el ecosistema de librerías y recursos en línea para utilizar algoritmos probados y eficientes, así como para adaptar y desarrollar algoritmos específicos que satisfacen las necesidades de la aplicación.

Al seleccionar las bibliotecas y los algoritmos externos, se consideraron aspectos como su rendimiento, su calidad de código y su idoneidad para el problema que se estaba abordando. Además, se llevó a cabo un proceso de validación y pruebas exhaustivas para garantizar su correcto funcionamiento dentro del contexto de la aplicación.

Esta combinación de bibliotecas npm, algoritmos de código abierto disponibles en línea y la implementación de algoritmos personalizados permitió lograr una solución integral y efectiva para los desafíos específicos abordados en el proyecto.

4. Visualización de los resultados

En cuanto a la visualización de los resultados, se ofrecen dos formas principales al usuario. En primer lugar, se presentan tablas y gráficas dentro de la aplicación, generadas a partir de la aplicación de los distintos algoritmos. Estas representaciones visuales permiten al usuario examinar y comprender de manera intuitiva los resultados obtenidos.

Además, se brinda al usuario la opción de descargar los resultados y visualizarlos en formato Excel o PDF, según el análisis que esté realizando. Esta funcionalidad amplía las posibilidades de análisis y permite un mayor grado de personalización en la visualización de los datos.

La combinación de tablas, gráficas y opciones de descarga en diferentes formatos ofrece al usuario una experiencia flexible y completa en la exploración y presentación de los resultados. Esto facilita su comprensión y permite un análisis más detallado de los datos, de acuerdo con las necesidades y preferencias individuales del usuario.

5. Pruebas y depuración

Se realizaron exhaustivas pruebas de rendimiento en la aplicación para garantizar su correcto funcionamiento. Durante estas pruebas, se identificó que, al procesar un gran volumen de datos iniciales, el rendimiento de la aplicación podía decaer, lo que resultaba en tiempos de espera prolongados para que los usuarios obtuvieran sus resultados.

Con el objetivo de abordar este problema, se propone una mejora consistente en trasladar algunos de los cálculos a un backend diseñado específicamente para esta tarea. Esta

optimización permitiría un procesamiento más eficiente y ágil de los datos, reduciendo significativamente los tiempos de espera.

Es importante mencionar que esta solución se contempla como parte de las futuras implementaciones del programa. Se reconoce la necesidad de mejorar el rendimiento en situaciones de gran carga de datos, y la incorporación de un backend especializado es una de las estrategias consideradas para resolver este desafío.

Al abordar este aspecto en futuras implementaciones, se busca brindar a los usuarios una experiencia fluida y sin interrupciones, independientemente del volumen de datos con el que trabajen. Esta mejora contribuirá a garantizar un rendimiento óptimo y una respuesta eficiente en la aplicación.

6. Despliegue y escalabilidad

En cuanto al despliegue de la aplicación, se optó por utilizar la plataforma gratuita Netlify. Esta plataforma permite el despliegue de aplicaciones web de manera sencilla y se integra con el repositorio de GitHub. Al vincular ambos, se generan las compilaciones necesarias en función de los commits realizados en el repositorio.

Esta elección proporciona una solución eficiente y conveniente para el despliegue continuo de la aplicación, facilitando la actualización y puesta en marcha de nuevas funcionalidades.

En cuanto a la escalabilidad, se aborda este tema de manera más específica en la sección de futuras implementaciones del documento. Se reconoce la importancia de tener en cuenta la escalabilidad en el crecimiento y desarrollo del proyecto.

Se están evaluando diferentes estrategias y tecnologías para garantizar que la aplicación pueda manejar eficientemente un aumento en la carga y el volumen de usuarios. Esto incluye consideraciones como el uso de tecnologías de contenedores, la implementación de servicios en la nube o la optimización de la arquitectura.

El enfoque en la escalabilidad permitirá que la aplicación se adapte y crezca de manera efectiva a medida que aumente su demanda y se añadan nuevas funcionalidades.

6. Trabajos relacionados

En esta sección, exploraremos diferentes herramientas y programas relacionados que se asemejan a la aplicación desarrollada. Estas herramientas ofrecen funcionalidades similares y pueden ser consideradas alternativas válidas según las necesidades y preferencias individuales. A continuación, presentaremos una breve descripción de cada una de estas herramientas y cómo se comparan con nuestro programa:

RapidMiner

RapidMiner es una plataforma de análisis de datos que se distingue por su enfoque en el modelado y análisis de datos sin necesidad de programación. Proporciona una interfaz gráfica intuitiva que permite a los usuarios arrastrar y soltar componentes para construir flujos de trabajo de análisis. Con RapidMiner, puedes realizar análisis de componentes principales, clustering y detección de outliers sin necesidad de escribir código. Además, ofrece una amplia gama de algoritmos de aprendizaje automático y herramientas de visualización para explorar y comprender los datos.

MATLAB

MATLAB es un entorno de programación y análisis numérico ampliamente utilizado en ciencia e ingeniería. Ofrece una amplia gama de funciones y herramientas para realizar análisis y manipulación de datos. MATLAB proporciona capacidades avanzadas para el análisis de componentes principales, clustering y cálculo de distancias. Además, incluye funciones estadísticas y herramientas de visualización que facilitan el procesamiento y la interpretación de los resultados del análisis. MATLAB es muy flexible y permite a los usuarios personalizar y automatizar sus análisis a través de programación.

Orange

Orange es un software de minería de datos y visualización que se destaca por su enfoque en la accesibilidad y facilidad de uso. Proporciona una interfaz gráfica intuitiva y amigable que permite a los usuarios construir flujos de trabajo de análisis de datos arrastrando y soltando componentes visuales. Orange ofrece una amplia gama de herramientas para realizar análisis de componentes principales, clustering y detección de outliers. Además, cuenta con herramientas de visualización interactiva que facilitan la comprensión de los datos y los resultados del análisis. Orange también permite la integración de código Python personalizado para realizar tareas más avanzadas y personalizadas.

7. Conclusiones y líneas de trabajo futuras

7.1. Conclusiones

Tras finalizar el desarrollo de la aplicación web en Angular, puedo concluir con gran satisfacción que se han cumplido todos los objetivos previstos inicialmente y se han agregado funcionalidades extras que han enriquecido aún más el producto final. Esta consecución exitosa refleja un esfuerzo y dedicación constantes a lo largo del proyecto.

Superar los objetivos personales inicialmente propuestos ha sido un logro significativo. Durante el proceso de desarrollo, he adquirido habilidades técnicas sólidas y he aprendido a aplicar metodologías ágiles en la gestión del proyecto. Esto ha permitido una planificación y ejecución eficiente, asegurando un flujo de trabajo fluido y adaptativo.

Un aspecto destacado es que la aplicación se ha diseñado y desarrollado de manera que pueda ser utilizada por cualquier usuario, incluso sin tener experiencia en el ámbito del análisis de datos. La interfaz intuitiva y la simplicidad en la navegación hacen que la aplicación sea accesible y fácil de usar para una amplia audiencia. Esto amplía su alcance y beneficio, permitiendo a los usuarios aprovechar los datos de manera efectiva sin requerir conocimientos técnicos especializados.

Personalmente, esta experiencia ha sido enriquecedora y gratificante. El proceso de construir una aplicación web desde cero me ha permitido enfrentarme a desafíos, aprender de ellos y aplicar soluciones efectivas. La adquisición de nuevas habilidades técnicas y la comprensión de las metodologías ágiles me han fortalecido como profesional en el desarrollo de software.

En general, la experiencia de desarrollo de esta aplicación en Angular ha sido altamente positiva. He logrado cumplir con los objetivos, superarlos y adquirir conocimientos valiosos en el camino. Estoy orgulloso del resultado final y emocionado por continuar creciendo en mi trayectoria profesional en lo que es mi pasión y será mi futuro dentro de este mundo, el desarrollo de software.

7.2. Líneas de trabajo futuras

En este apartado se van a exponer las diferentes ideas que se dispondrán en el futuro sobre la aplicación. Son las siguientes:

1. **Alojamiento en un servidor propio**

Estableceremos un hito importante al alojar nuestra aplicación en un servidor propio. Esto nos brindará un mayor control sobre la disponibilidad y rendimiento de la aplicación, asegurando una experiencia fluida para nuestros usuarios. Además, podremos implementar medidas adicionales de seguridad y escalabilidad para satisfacer las demandas crecientes de nuestros clientes.

2. Adaptación multiplataforma

Ampliaremos significativamente el alcance de nuestra aplicación al hacerla totalmente responsive y compatible con múltiples plataformas. Nuestros usuarios podrán acceder y utilizar la aplicación de manera fluida tanto en dispositivos móviles como en tabletas, brindándoles la comodidad de acceder y analizar datos desde cualquier lugar y en cualquier momento.

3. Internacionalización a otros idiomas

Expandiremos nuestro público objetivo al realizar la internacionalización de la aplicación. Al proporcionar una experiencia localizada para los usuarios, ampliaremos nuestras oportunidades de dar a conocer la aplicación en otros sitios. Aseguraremos que la interfaz y los contenidos se adapten de manera precisa y fluida a la lengua y cultura, ofreciendo una experiencia óptima para nuestros usuarios.

4. Aplicación de escritorio sin conexión

Brindaremos a nuestros usuarios la flexibilidad de utilizar nuestra aplicación incluso en entornos sin conexión a internet. Al desarrollar una versión de escritorio que no dependa de la conexión en línea para cargarse, garantizaremos la accesibilidad y disponibilidad de la aplicación en diversas situaciones, como entornos aislados o de baja conectividad. Esto permitirá a los usuarios realizar análisis de datos sin restricciones y sin depender de la infraestructura de red.

8. Bibliografía

Academia Serrano. (2022). *Análisis de componentes principales (PCA) - YouTube*. https://www.youtube.com/watch?v=7My_PBhxeP4

Adobe XD. (2023). *Adobe XD Learn & Support*. https://helpx.adobe.com/support/xd.html

Amstrong Jim. (2018). *TSDataStats/DataStats.ts at master · theAlgorithmist/TSDataStats · GitHub*. https://github.com/theAlgorithmist/TSDataStats/blob/master/src/DataStats.ts

Angular. (2023). *Angular - What is Angular?* https://angular.io/guide/what-is-angular

Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering Points To Identify the Clustering Structure. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 49–60. https://doi.org/10.1145/304182.304187

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), 243–256. https://doi.org/10.1016/j.patcog.2012.07.021

Avila Leyanis, Mendoza Niusvel, & Alonso Andres. (2019). *Detección de anomalías basada en aprendizaje profundo: Revisión*. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992019000300107

Badge Codacy. (2023). *@swimlane/ngx-charts - npm*. https://www.npmjs.com/package/@swimlane/ngx-charts

Bello Omar. (2021). *Análisis de Componentes Principales: Maximización de varianza - YouTube*. https://www.youtube.com/watch?v=-blx46lDhFY

Benites Luis. (2022a). *▷ Distancia de Mahalanobis: definición simple, ejemplos en 2023 → STATOLOGOS®*. https://statologos.com/distancia-mahalanobis/

Benites Luis. (2022b). *▷ Distancia de Mahalanobis: definición simple, ejemplos en 2023 → STATOLOGOS®*. https://statologos.com/distancia-mahalanobis/

Binani Harsh. (2020). *R and Javascript : Execution, Libraries, Integration | HackerNoon*. https://hackernoon.com/r-and-javascript-execution-libraries-integration-40a30726f295

Boostrap. (2023). *Get started with Bootstrap · Bootstrap v5.3*. https://getbootstrap.com/docs/5.3/getting-started/introduction/

Bootstrap. (2023). *Angular powered Bootstrap*. https://ng-bootstrap.github.io/#/home

Bostock Mike. (2021). *Scatterplot / D3 | Observable*. https://observablehq.com/@d3/scatterplot

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (1999). Principles of Data Mining and Knowledge Discovery. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 1704). Springer Verlag. https://doi.org/10.1007/978-3-540-48247-5_28

Breuniq, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-based Local Outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, *29*(2), 93–104. https://doi.org/10.1145/335191.335388

Capterra. (2020). *Netlify - Opiniones, precios y características - Capterra España 2023*. https://www.capterra.es/software/154989/netlify

Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, *40*(1), 200–210. https://doi.org/10.1016/j.eswa.2012.07.021

ChatJS. (2023). *Chart.js | Open source HTML5 Charts for your website*. https://www.chartjs.org/

Clare Timothy, & Potix Corporation. (2012). *ZK MVVM*. Potix. http://books.zkoss.org/wiki/Small_Talks/2012/February/New_Features_of_ZK_6#ZK_MVVM

Clusterfck. (2010). *clusterfck - JavaScript hierarchical clustering*. https://harthur.github.io/clusterfck/

Coomans, D., & Massart, D. L. (1982). Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, *136*(C), 15–27. https://doi.org/10.1016/S0003-2670(01)95359-0

Cotton Ben. (2016). Making documentation easy with Read the Docs. *Opensource.Com*. https://opensource.com/business/16/8/introduction-read-docs

Coyler Chris. (2013). *A Complete Guide to Flexbox | CSS-Tricks - CSS-Tricks*. https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Cypress. (2023). *JavaScript Web Testing and Component Testing Framework | cypress.io*. https://www.cypress.io/

D3Gallery. (2023). *D3 Gallery / D3 | Observable*. https://observablehq.com/@d3/gallery

Damian A. (2015). *GitBook, escribe documentación para tus proyectos desde Ubuntu | Ubunlog*. https://ubunlog.com/gitbook-editor-documentacion-ubuntu/

De Maesschalck, R., Jouan-Rimbaud, D., & Massart, D. L. (2000). The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, *50*(1), 1–18. https://doi.org/10.1016/s0169-7439(99)00047-7

Deza, M. M., & Deza, E. (2013). Encyclopedia of Distances. *Encyclopedia of Distances*. https://doi.org/10.1007/978-3-642-30958-8

Dixon, & Moe. (2023). *Códigos de Colores HTML*. https://htmlcolorcodes.com/es/

Edix. (2022). *¿Qué es Adobe XD y para qué sirve? ¡Te lo contamos todo!* https://www.edix.com/es/instituto/adobe-xd/

Ekstein Nikki. (2019). Using Trello to Plan Your Next Vacation (Really). *Www.Bloomberg.Com*. https://www.bloomberg.com/tosv2.html?vid=&uuid=a1895840-ec37-11e9-bd54-8dc30bda8b5f&url=L25ld3MvYXJ0aWNsZXMvMjAxOS0wMy0xNC9ob3ctdG8tdXNlLXRyZWxsby10by1wbGFuLXRyYXZlbC1ib29rLWZsaWdodHMtbWFwLXZhY2F0aW9ucw==

Esri. (2019). *Cómo funciona el clustering basado en densidad—ArcGIS Pro | Documentación*. https://pro.arcgis.com/es/pro-app/latest/tool-reference/spatial-statistics/how-density-based-clustering-works.htm

Estivill-Castro, V., & Estivill-Castro, V. (2002). Why so many clustering algorithms — A Position Paper. *ACM SIGKDD Explorations Newsletter*, *4*(1), 65–75. https://doi.org/10.1145/568574.568575

Figma. (2023). *Free Design Tool for Websites, Graphic Design and More | Figma*. https://www.figma.com/design/

FontAwesome. (2023). *Find Icons with the Perfect Look & Feel | Font Awesome*. https://fontawesome.com/icons

Fowler, M. (2004). *The Presentation Model Design Pattern*. Martin Fowler.com. http://martinfowler.com/eaaDev/PresentationModel.html

Gabriel João. (2017). *Clustering with Javascript — Part 3: Clustering Algorithms in Practice | by João Gabriel Lima | Medium*. https://medium.com/@joaogabriellima/clustering-with-javascript-part-3-clustering-algorithms-in-practice-75631b241917

García-Pérez, C. A. (2016). Me gusta citar. *Geotechnical, Geological and Earthquake Engineering*, *16*, 129–145. https://doi.org/10.1007/978

Geekscoach. (2020). *K-Means | Clustering. K-means o en español seria K medias es… | by Geekscoach | Medium*. https://geekscoach.medium.com/k-means-clustering-cebbcb4e38ec

Geun Kim, M. (2000). Multivariate outliers and decompositions of Mahalanobis distance. *Communications in Statistics – Theory and Methods*, *29*(7), 1511–1526. https://doi.org/10.1080/03610920008832559

GitBook. (2023a). *GitBook - Where technical teams document.* https://www.gitbook.com/

GitBook. (2023b). *Introduction to GitBook - GitBook Documentation*. https://docs.gitbook.com/

GitBook. (2023c). *¿Qué es Gitbook? · GitBook*. https://ull-esit-dsi-1617.github.io/estudiar-las-rutas-en-expressjs-alberto-diego/Alberto/gitbook/queesgitbook.html

González Guillermo. (2020). *▷ Análisis de componentes principales (PCA): mejor explicado | Machine Learning Studio® 2023*. https://mlstudio.jaol.net/principal-components-analysis-pca-better-explained/

Gossman, J. (2005). *Tales from the Smart Client: Introduction to Model/View/ViewModel pattern for building WPF apps*. https://docs.microsoft.com/en-us/archive/blogs/johngossman/introduction-to-modelviewviewmodel-pattern-for-building-wpf-apps

Gossman, J. (2006). *Tales from the Smart Client: Advantages and disadvantages of M-V-VM*. https://docs.microsoft.com/en-gb/archive/blogs/johngossman/advantages-and-disadvantages-of-m-v-vm

Harmouch Mahmoud. (2021). *17 Clustering Algorithms Used In Data Science and Mining | by Mahmoud Harmouch | Towards Data Science*. https://towardsdatascience.com/17-clustering-algorithms-used-in-data-science-mining-49dbfa5bf69a

Holscher, E. (2022). Read the Docs 2021 Stats. *Read the Docs Blog*. https://blog.readthedocs.com/read-the-docs-2021-stats.html

Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis*. 561.

Hsu, D., Kakade, S. M., & Zhang, T. (2012). A spectral algorithm for learning Hidden Markov Models. *Journal of Computer and System Sciences*, *78*(5), 1460–1480. https://doi.org/10.1016/j.jcss.2011.12.025

Jia, Z., & Song, L. (2020). Weighted k-Prototypes Clustering Algorithm Based on the Hybrid Dissimilarity Coefficient. *Mathematical Problems in Engineering*, *2020*. https://doi.org/10.1155/2020/5143797

Jos de Jong. (2023). *josdejong/mathjs: An extensive math library for JavaScript and Node.js*. https://github.com/josdejong/mathjs

Kaggle. (2023). *Kaggle: Your Home for Data Science*. https://www.kaggle.com/

KeepCoding Team. (2023). *¿Qué es el patrón de arquitectura MVVM?* https://keepcoding.io/blog/que-es-el-patron-de-arquitectura-mvvm/

Li, J., Song, S., Zhang, Y., & Zhou, Z. (2016). Robust K-median and K-means clustering algorithms for incomplete data. *Mathematical Problems in Engineering*, *2016*. https://doi.org/10.1155/2016/4321928

Liu Shuyi, & Wu Wei. (2017). *GENERALIZED MAHALANOBIS DEPTH IN POINT PROCESS AND ITS APPLICATION IN NEURAL CODING on JSTOR*. https://www.jstor.org/stable/26362214

López José Francisco, & Coll Francisco. (2020a). *Covarianza - Qué es, definición y concepto | 2023 | Economipedia*. https://economipedia.com/definiciones/covarianza.html?nab=1&utm_referrer=https%3A%2F%2Fsearch.brave.com%2F

López José Francisco, & Coll Francisco. (2020b). *Varianza - Qué es, definición y significado | 2023 | Economipedia*. https://economipedia.com/definiciones/varianza.html?nab=1&utm_referrer=https%3A%2F%2Fsearch.brave.com%2F

Madhukumar, S., & Santhiyakumari, N. (2015). Evaluation of k-Means and fuzzy C-means segmentation on MR images of brain. *Egyptian Journal of Radiology and Nuclear Medicine*, *46*(2), 475–479. https://doi.org/10.1016/J.EJRNM.2015.02.008

Maisam Muhammad. (2023). *Calcular la distancia de Mahalanobis en Python | Delft Stack*. https://www.delftstack.com/es/howto/python/python-mahalanobis-distance/

Massey, S. (2011). *Presentation Patterns in ZK*. http://www.slideshare.net/simbo1905/design-patterns-in-zk-java-mvvm-as-modelviewbinder

MathJS. (2020). *math.js | an extensive math library for JavaScript and Node.js*. https://mathjs.org/

Matlab. (2023). *MATLAB - El lenguaje del cálculo técnico*. https://es.mathworks.com/products/matlab.html

MDBoostrap. (2023a). *Bootstrap 5 & Angular 12 - Free Material Design UI KIT*. https://mdbootstrap.com/docs/angular/

MDBoostrap. (2023b). *Bootstrap 5 & Angular 12 - Free Material Design UI KIT*. https://mdbootstrap.com/docs/angular/

Microsoft. (2005). Introduction to Model/View/ViewModel pattern for building WPF apps. *Microsoft Developer Network*. https://blogs.msdn.microsoft.com/johngossman/2005/10/08/introduction-to-modelviewviewmodel-pattern-for-building-wpf-apps/

Microsoft. (2012). The MVVM Pattern. *Msdn.Microsoft.Com*. https://msdn.microsoft.com/en-us/library/hh848246.aspx

Microsoft. (2022). How to implement MVVM (Model–View–ViewModel) in TDD (test-driven development). *Microsoft Developer Network*. https://code.msdn.microsoft.com/How-to-implement-MVVM-71a65441

Modzilla. (2023). *Making content editable | MDN*. https://developer.mozilla.org/es/docs/conflicting/Web/HTML/Global_attributes/contenteditable

Moreno, I. (2023). *Introducción al Clustering DBSCAN - StatDeveloper*. https://www.statdeveloper.com/clustering-dbscan/

Murtagh, F., & Kurtz, M. J. (2012). *A History of Cluster Analysis Using the Classification Society's Bibliography Over Four Decades*. http://arxiv.org/abs/1209.0125

Ng, R. T., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, *14*(5), 1003–1016. https://doi.org/10.1109/TKDE.2002.1033770

NGX-Charts. (2022a). *Introduction - ngx-charts*. https://swimlane.gitbook.io/ngx-charts

NGX-Charts. (2022b). *NgxCharts*. https://swimlane.github.io/ngx-charts/#/ngx-charts/bar-vertical

NIST. (2010). *6.5.4.1. Mean Vector and Covariance Matrix*. https://www.itl.nist.gov/div898/handbook/pmc/section5/pmc541.htm

Orange Data Mining. (2023). *Orange Data Mining - Data Mining*. https://orangedatamining.com/

Osmami Addy. (2012). Understanding MVVM: A Guide for JavaScript Developers. *AddysOnmani.Com*. http://addyosmani.com/blog/understanding-mvvm-a-guide-for-javascript-developers/

Palacio Marta. (2022). *Scrum Master Libro*. https://www.scrummanager.com/files/scrum_master.pdf

Parzibyte. (2020). *HTML a PDF con JavaScript - Parzibyte's blog*. https://parzibyte.me/blog/2020/09/05/html-pdf-javascript/

Paula, & Cansu. (2023). *Biplot for PCA Explained (Example & Tutorial) - How to Interpret*. https://statisticsglobe.com/biplot-pca-explained

Pavlutin Dimitri. (2022). *Covariance and Contravariance in TypeScript*. https://dmitripavlutin.com/typescript-covariance-contravariance/

Piechocki Miłosz. (2020). *Strict function types in TypeScript: covariance, contravariance and bivariance - codewithstyle.info*. https://codewithstyle.info/Strict-function-types-in-TypeScript-covariance-contravariance-and-bivariance/

Ponce, R. V., Luis, J., & Alcaraz, G. (2013). Evaluation of Technology using TOPSIS in Presence of Multi-collinearity in Attributes: Why use the Mahalanobis distance? *Rev. Fac. Ing. Univ. Antioquia N*, 31–42.

Popper. (2023). *Popper - Tooltip & Popover Positioning Engine*. https://popper.js.org/

prabhjotkushparmar. (2023). *Covariance Matrix - Definition, Formula, Examples, Properties and FAQs*. https://www.geeksforgeeks.org/covariance-matrix/

Prabhnkaran Selva. (2019). *Mahalanobis Distance - Understanding the math with examples (python) - Machine Learning Plus*. https://www.machinelearningplus.com/statistics/mahalanobis-distance/

R Documentation. (2022). *depth.Mahalanobis: Calculate Mahalanobis Depth in ddalpha: Depth-Based Classification and Calculation of Data Depth*. https://rdrr.io/cran/ddalpha/man/depth.Mahalanobis.html

RapidMiner. (2023). *RapidMiner | Amplify the Impact of Your People, Expertise & Data*. https://rapidminer.com/

Read The Docs. (2023). *Inicio | Read the Docs*. https://readthedocs.org/

Read the Docs. (2023). *Read the Docs: documentation simplified — Read the Docs user documentation 9.7.0 documentation*. https://docs.readthedocs.io/en/stable/

RedHat. (2022). *¿Qué es la metodología ágil?* https://www.redhat.com/es/devops/what-is-agile-methodology

Reynolds, D. (2009). Gaussian Mixture Models. *Encyclopedia of Biometrics*, 659–663. https://doi.org/10.1007/978-0-387-73003-5_196

Rodó Paula, & Sevilla Andrés. (2019). *Vectores y valores propios - Qué es, definición y concepto | 2023 | Economipedia*. https://economipedia.com/definiciones/vectores-y-valores-propios.html?nab=1&utm_referrer=https%3A%2F%2Fsearch.brave.com%2F

Rodriguez Txema. (2014). *GitBook, crea documentación técnica y libros usando Markdown y Git/Github de forma flexible*. https://www.genbeta.com/desarrollo/gitbook-crea-documentacion-tecnica-y-libros-usando-markdown-y-git-github-de-forma-flexible

Salton Kevin. (2017). *How DBSCAN works and why should we use it? | by Kelvin Salton do Prado | Towards Data Science*. https://towardsdatascience.com/how-dbscan-works-and-why-should-i-use-it-443b4a191c80

Sanz Francisco. (2018). *Algoritmo K-Means - Clustering y cómo funciona*. https://www.themachinelearners.com/k-means/

Sass. (2023). *Sass: Documentation*. https://sass-lang.com/documentation/

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems*, *42*(3). https://doi.org/10.1145/3068335

Schubert, E., Zimek, A., & Kriegel, H. P. (2012). Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery*, *28*(1), 190–237. https://doi.org/10.1007/s10618-012-0300-z

SciPy Documentation. (2022). *scipy.spatial.distance.mahalanobis — SciPy v1.10.1 Manual*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.mahalanobis.html

Sejda. (2023). *Convertir HTML a PDF en línea*. https://www.sejda.com/es/html-to-pdf

Shifflett, K. (2023). *Learning WPF M-V-VM.* http://karlshifflett.wordpress.com/2008/11/08/learning-wpf-m-v-vm/

Shvets Alexander. (2020). *Patrones de diseño / Design patterns*. https://refactoring.guru/es/design-patterns

Sibson, R., & Sibson, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, *16*(1), 30–34. https://doi.org/10.1093/comjnl/16.1.30

Smith, J. (2009). WPF Apps with the Model–View–ViewModel Design Pattern. *MSDN Magazine*. http://msdn.microsoft.com/en-us/magazine/dd419663.aspx

Statistics Kingdom. (2022). *Cluster analysis*. https://www.statskingdom.com/cluster-analysis.html

Statlogos. (2021). *▷ Cómo calcular la distancia euclidiana en Excel en 2023 → STATOLOGOS®*. https://statologos.com/distancia-euclidiana-excel/

Stonis Michael, Jain Tarun, & Pine David. (2022). *Model-View-ViewModel | Microsoft Learn*. https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm

Tailwind CSS. (2023). *Installation - Tailwind CSS*. https://tailwindcss.com/docs/installation

Trello. (2016a). Breve historia de Trello. *Trello.Com*. https://trello.com/about

Trello. (2016b). What is Trello? *Trello.Com*. http://help.trello.com/article/708-what-is-trello

twelch. (2021). *@turf/clusters-dbscan - npm*. https://www.npmjs.com/package/@turf/clusters-dbscan

UniOviedo. (2016). *kmeans*. https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html

Universidad de Burgos. (2023). *Aul@Abierta: formación 2022-2023 | Universidad de Burgos*. https://www.ubu.es/aprendizaje-e-investigacion/aulabierta-formacion-guias/aulabierta-formacion-2022-2023

Valdes Michael. (2018). *javascript - Como convertir un HTML a PDF con jsPDF, sin perder los estilos CSS - Stack Overflow en español*. https://es.stackoverflow.com/questions/156647/como-convertir-un-html-a-pdf-con-jspdf-sin-perder-los-estilos-css

Veltman. (2020). *mahalanobis - npm*. https://www.npmjs.com/package/mahalanobis

Villadiego, C. /, & Burgos. (2023). *UNIVERSIDAD DE BURGOS ESCUELA POLITÉCNICA SUPERIOR Pag. 1 de 8 REGLAMENTO SOBRE TRABAJO FIN DE GRADO Y TRABAJO FIN DE MASTER*.

WanaTop. (2021). *3 mejores herramientas de prototipado | Experts Academy*. https://expertsacademy.es/blog/mejores-herramientas-prototipado/

Weisstein, E. W., & Weisstein, E. W. (2020). Covariance Matrix. *MathWorld*. http://mathworld.wolfram.com/CovarianceMatrix.html

Wicklin Rick. (2012). *What is Mahalanobis distance? - The DO Loop*. https://blogs.sas.com/content/iml/2012/02/15/what-is-mahalanobis-distance.html

Wikipedia. (2023a). *Aprendizaje no supervisado - Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/wiki/Aprendizaje_no_supervisado

Wikipedia. (2023b). *Distancia de Mahalanobis - Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/wiki/Distancia_de_Mahalanobis

Wikipedia. (2023c). *Distancia de Mahalanobis - Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/wiki/Distancia_de_Mahalanobis

Wikipedia. (2023d). *GitHub - Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/wiki/GitHub

Wikipedia. (2023e). *k-means clustering - Wikipedia*. https://en.wikipedia.org/wiki/K-means_clustering

Wikipedia. (2023f). *Matriz de covarianza - Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/wiki/Matriz_de_covarianza

Wikipedia. (2023g). *Model–view–viewmodel - Wikipedia*. https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel

Wikipedia. (2023h). *Rayleigh quotient - Wikipedia*. https://en.wikipedia.org/wiki/Rayleigh_quotient

Wikipedia. (2023i). *Sample mean and covariance - Wikipedia*. https://en.wikipedia.org/wiki/Sample_mean_and_covariance

Wildermuth, S. (2010). *Windows Presentation Foundation Data Binding: Part 1*. Microsoft. http://msdn.microsoft.com/en-us/library/aa480224.aspx

Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, *2*(2), 165–193. https://doi.org/10.1007/S40745-015-0040-1

York University. (2020). *Zotero vs Mendeley Comparison - York University Libraries*. https://www.library.yorku.ca/web/research-learn/citing-your-work-academic-integrity/citations/zotero-vs-mendeley-comparison/

Yufeng. (2022). *Understanding OPTICS and Implementation with Python | by Yufeng | Towards Data Science*. https://towardsdatascience.com/understanding-optics-and-implementation-with-python-143572abdfb6

Zenhub Team. (2015). *ZenHub 2.0 : Project management, evolved.* https://blog.zenhub.com/zenhub-2-0-the-evolution-of-zenhub/