MEMORIA CAPSTONE
DATA SCIENCE
PREDICT FRAUD - RETAIL

Aleix Blanch
Arturo Ramos
Nil Romans

ÍNDICE

Introducción

En la actualidad, los datos son un elemento fundamental en la toma de decisiones en el día a día de las corporaciones y entidades. Se han convertido en un intangible muy valioso, que puede determinar la dirección y enfoque del negocio. La suma de la programación, el análisis y la parte del negocio son tres aspectos clave para optimizar nuestros esfuerzos.

En nuestro caso, DS Market, una pequeña cadena de centros comerciales ubicada en Estados Unidos, conocida anteriormente como TradiStores, se ha ido reinventando, aunque introduciéndose tarde en la era digital, en comparación con la competencia. Unos de los cambios relevantes ha sido optar por el cambio de nombre de la corporación, además de tomar la gran decisión de incorporarse a la transformación digital a partir de un plan "renove" de una duración de cinco años.

Uno de los primeros pasos que ha seguido DSMarket ha sido incorporar una directora digital para liderar esta transformación de la mecanización a la digitalización de todos los ámbitos de la cadena. A partir de ciertos estudios, se ha determinado que el error obtenido a partir de los datos mal utilizados está afectando de manera directa a DSMarket, incluyendo todas las áreas que la forman.

Con todo esto, la prioridad de tratar los datos para conseguir un mejor rendimiento del capital y producto de la corporación de DSMarket es un hecho, para poder ser más potencial en el sector en el que opera, considerando la transformación hacia una empresa basada en los datos, como algo vital.

Para ello, centraremos el análisis de cada ángulo de la actividad de la cadena, empezando por tres de las ciudades donde se ubica la empresa, Boston, Nueva York y Filadelfia.

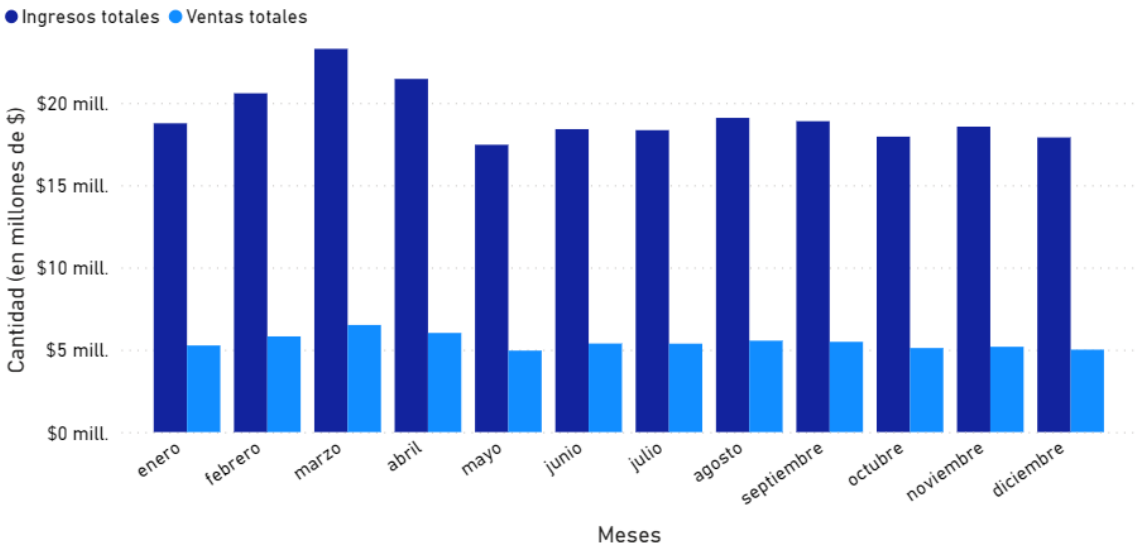
Análisis

El reto de optimizar las operaciones de la corporación de DSMarket empieza por hacer un análisis de cada sector que forma la empresa. Es por eso que se prioriza controlar las zonas de Boston, Nueva York y Filadelfia para, más adelante, expandir los análisis por el resto del territorio.

Actualmente, la cadena dispone de diez tiendas entre estos tres territorios que producen unos ingresos de más de 230 millones de dólares, con unas ventas que se acercan a los 66 millones de artículos. Estos datos hacen referencia al periodo de tiempo comprendido entre 2011 y 2016.

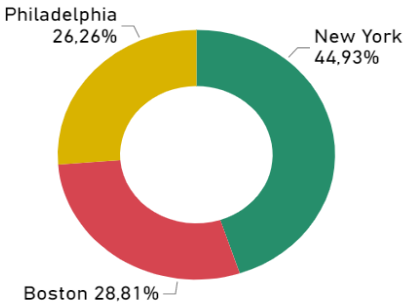
Si comparamos el nivel total de ingresos con el nivel total de ventas distribuido durante todos los meses del año, vemos que durante los primeros meses, concretamente entre febrero y abril, se produce el mayor número de ventas, pero sobre todo son los meses en que la corporación genera más ingresos, llegando a superar los 20 millones de dólares en ganancias.

Figura 1. COMPARATIVA ENTRE SUMA DE INGRESOS Y NÚMERO DE VENTAS (POR MES)



Además, vemos que la distribución de los ingresos totales no es parecida entre estas tres regiones. Nueva York es la zona geográfica que destaca por su volumen, tanto a nivel de ingresos como de número de ventas, generando cerca del 50 % del total de la corporación, concretamente un 44,93 % sobre el total. Las otras dos regiones tienen registros similares, en torno al 26 - 30 % de los ingresos.

Figura 2. DISTRIBUCIÓN DE LOS INGRESOS POR REGIONES

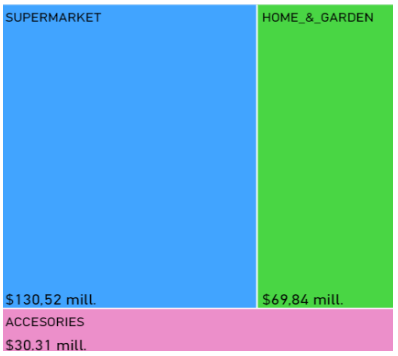


Por último, sabemos que nuestros artículos se dividen en tres categorías, en artículos de **supermercado**, de **hogar y jardín** y un último grupo de **accesorios**.

Después de analizar cada categoría con detalle, vemos que el grupo de artículos que pertenecen a la categoría de **supermercado** generan más de 130 millones de dólares, con lo que es la categoría por excelencia de la empresa, puesto que genera mucho más que las otras. En segundo plano, tenemos los artículos de **hogar y jardín**, que llegan casi a los 70 millones de dólares, y finalmente, el grupo de **accesorios**, que logra llegar a la cantidad de 30 millones de dólares, como se puede ver en la figura nº. 3.

Si nos centramos en las ventas de la cadena, vemos que destacan los meses de marzo y abril, meses en que las ventas superan los seis millones de artículos vendidos en el periodo de seis años comentado anteriormente.

Figura 3. INGRESOS POR CATEGORÍA



Si hacemos hincapié en cada región, Nueva York es la región que produce más ventas, con más de 28,5 millones de ítems vendidos, y de la misma manera las tiendas que más venden pertenecen a esta región, Tribeca y Greenwich Village con unos valores que alcanzan los 11 y los 7 millones de artículos vendidos respectivamente. Las otras dos regiones venden a un nivel parecido y las tiendas que más productos venden son Roxbury en Boston, con poco más de 7 millones de ítems, y la tienda Yorktown de Filadelfia con un valor de 6,5 millones de artículos vendidos.

La categoría de supermercado es la que produce más ventas en detrimento de las otras dos, quizás por disponer de productos o ítems de un valor más elevado para nuestros clientes o porque son esenciales para la vida cotidiana de estos. Además, no todos nuestros clientes tienen casa con jardín, por lo que no compran artículos de esta categoría o quieren comprar accesorios en una tienda más especializada de referencia donde pueden obtenerlos.

Por último, si somos más concretos y nos fijamos en nuestros ítems, destacamos el artículo **90** de supermercado que supera el millón de ventas, seguido de muy cerca por el artículo **586** también de supermercado (919.000 ventas). Estos dos artículos son los más destacados y, por lo tanto, los más rentables para la empresa. Estos datos son equiparables a las regiones de Nueva York y Boston, donde este patrón es similar y se produce en ambas zonas, pero en Filadelfia no es así. En este caso, en la región más sureña, destaca el artículo **226** de supermercado.

Y si nos centramos en los artículos con menos ventas, llegamos a la conclusión de que los ítems menos vendidos son, por consecuencia, los de las categorías con menos ventas, resultando así los artículos **101** y **175** de hogar y jardín y los **84** y **119** de accesorios los menos vendidos, con valores por debajo de las 750 ventas.

Los ingresos se distribuyen más o menos de la misma manera que las ventas en relación con el porcentaje entre regiones, es decir, de los 230 millones de dólares de ingresos, casi la mitad, concretamente un 43,65 % pertenecen a Nueva York, mientras que los ingresos en Boston y Filadelfia son del 28,77 % y 27,58 % respectivamente. Destacan los ítems 586, 120 y 90 de supermercado, en este orden, que superan el millón y medio de dólares en ingresos.

Finalmente, centrando la atención en el producto, vemos que tanto en Boston como en Nueva York destacan tres artículos sobre el resto, los 90, 586 y 252 de supermercado, variando el orden entre los dos primeros en cada región. Pero en Filadelfia, la situación es totalmente diferente, así como hemos comentado en párrafos anteriores. Tan solo repiten los dos primeros artículos de las otras regiones (90 y 586) entre los cinco más vendidos y, además, destaca sobresaliendo por encima del resto el artículo 226, que no encontramos en los ítems “líderes” de las otras regiones.

Por otro lado, destacamos que los artículos que más se venden son los que acumulan los mayores ingresos, aunque existe alguna excepción como en el caso del accesorio 354 en Boston, del que se venden muchas menos unidades, pero que acumula una gran cantidad de ingresos.

Initial Preprocessing

Después de realizar el análisis de la empresa, podemos empezar a tratar los datos. La primera tarea a efectuar es importar las librerías necesarias y cargar los datos, empezando por el primer dataset, el de **eventos**.

Vemos la información que contiene el dataset y tomamos algunas decisiones: pasamos las fechas a formato *datetime* y guardamos el primer día que tengamos. También cambiamos el orden de las columnas para mayor comodidad y mejor visualización del dataset y hacemos un pequeño estudio de este (*head,* diferentes eventos únicos que tengamos, valores nulos, etc.). En este punto, rellenamos los eventos nulos con la palabra ***'None'***, indicando que en ese día no existe ningún tipo de evento. Más tarde, hacemos un *OneHotEncoding* de los eventos para tenerlos separados por columnas y creamos una nueva columna para tener los días de la festividad de Ramadán completos, rellenando los valores nulos y eliminando la columna sobrante. Finalmente, juntamos el dataset por semanas como se pide en la tarea, reseteamos el índice y lo guardamos.

Acto seguido, nos centramos en el dataset de **precios**. Lo cargamos y hacemos una visualización de la información que contiene (*info, head,* valores nulos…). Procedemos a eliminar todos los valores nulos y ponemos la columna ***'yearweek'*** a fecha y renombramos la columna a ***'date'*** para que coincida con el resto de datasets. El siguiente punto es ordenar el dataset, resetear el índice y cambiar el orden de las columnas para guardarlo.

Por último, cargamos el dataset de **ventas** y hacemos un pequeño estudio de la información que contiene. Creamos un dataset con solo las columnas que nos interesan y las renombramos. También creamos la columna ***'date'*** formada por la columna *día* (modificada por el número) y agrupamos el dataset por semanas. Finalmente, juntamos las columnas anteriores que habíamos dejado de lado y hacemos un pequeño estudio del dataset final, para ordenar las columnas y guardarlo.

Una vez preprocesados los tres datasets, procedemos a juntarlos. Con la unión de los tres datasets en un nuevo, **full_df**, creamos la columna *revenue* y lo guardamos.

Modelo

Una vez con el preprocesado de los datos hecho y los tres datasets unificados en uno solo, empezamos a trabajar para poder aplicar el modelo.

Primeramente, importamos todas las librerías necesarias para configurar el notebook para que pandas devuelva datasets en lugar de *arrays*. Cargamos el dataset y definimos el nombre de las columnas en mayúsculas, para posteriormente darle un orden específico y creamos una función de *preprocessing* inicial que ponga en mayúsculas el nombre de las columnas, cambie el formato *datetime* de las fechas, realice un *encoder* de las columnas necesarias, elimine columnas que no aportan información, renombre las columnas y, finalmente, las ordene. En este punto, aplicamos la función a nuestro *dataframe* y hacemos un pequeño estudio de este (*info, head,* recuento de valores…). Guardamos la primera y la última fecha.

A partir de estos pasos, procedemos a buscar los valores nulos y vemos que tenemos muchos nulos en dos columnas en concreto, la del precio medio por semana y en la de ganancias semanales. Para rellenarlos optamos por adjudicar un cero en la columna de ganancias semanales, ya que si tenemos un valor nulo en esta columna significa que no ha habido ventas, y en cuanto al precio, utilizamos la función *backfill* y *ffill*, que rellena los valores nulos con el valor que encuentra, en primera instancia, por debajo o hacia atrás (*backfill*) y, en caso de no tener ningún valor por debajo (la última fila), rellena el valor que encuentra por arriba o hacia delante (*ffill*). Es decir, rellenamos los precios medios por semana que no tenemos con los precios de la semana anterior o posterior, para que tengamos un valor lógico, antes que rellenar con un cero y no disponer de información útil.

Acto seguido, creamos diferentes gráficas para visualizar las ventas semanales durante toda la serie temporal y las respectivas autocorrelaciones y autocorrelaciones parciales. También visualizamos las ventas por región, categoría del ítem y por tienda.

Realizamos un *np.clip* de las ventas semanales entre los valores 0 y 15 para eliminar valores *outliers*. En caso de que un valor supere el límite, ya sea por arriba o por abajo, se le otorga el valor máximo o mínimo que le indicamos, en este caso, con un valor mínimo de 0 y un valor máximo de 15.

Más adelante creamos una función para crear nuevos datasets con nuevas columnas o variables. Por ejemplo, podemos agrupar las ventas por las columnas que elijamos y aplicarle una función como la media, el máximo, el mínimo, etc. En este caso, creamos columnas que estén formadas por la media de ventas semanales por tienda o la suma de ventas semanales por tienda, por región y por categoría de los ítems. Una vez generadas todas las nuevas variables, procedemos a la anexión de estas en un mismo dataset.

Finalmente, eliminamos las columnas autoexplicativas, es decir, que contengan información de otras columnas. Por último, ponemos el ID como índice del *dataframe*.

Con el *preprocessing* y el *eda* hechos, pasamos a aplicar el modelo.

Para separar el *train*, *validation* (o *test*) y *test* (o *predict*), seleccionamos todo el dataset excepto las últimas ocho semanas para el *train*, las cuatro siguientes para *validation*, y las últimas cuatro, para predecir, cómo nos piden desde el departamento.

Separamos los respectivos valores **X** y **y** de cada dataset y guardamos el X *test*.

Aplicamos el modelo en el *train*, en este caso, el XGBRegressor con *early stopping rounds* con valor de 10, para no pasar tiempo extra ejecutando el modelo y cuando deje de mejorar, parar la ejecución automáticamente. Procedemos a guardar el modelo y realizamos una gráfica con la importancia de las diferentes variables existentes en el modelo. Vemos que el ítem **department** es la variable más importante para el modelo, seguido por otras dos que hemos creado nosotros, la suma de ***weekly sales*** por categoría y la suma de ***weekly sales*** por tienda.

Después de estos pasos, aplicamos el modelo en la validación y vemos que el error es bastante bajo, así que aplicamos el modelo en las predicciones y estudiamos los resultados.

Vemos las ventas por ítem, la suma total de ventas, el *mean squared error* de las predicciones, que también es bajo, etc.

Finalmente, creamos gráficas en las que podemos visualizar y comparar las ventas reales con las ventas predecidas y nos damos cuenta de que el modelo se ajusta bastante a la realidad.

Clustering

Con el objetivo de agrupar los productos en categorías relevantes, se ha llevado a cabo un análisis de clustering de los productos. El clustering es una técnica fundamental en el análisis de datos, ya que permite identificar patrones y similitudes entre los productos, lo que resulta útil para diversas aplicaciones, como la gestión de inventario y la segmentación de clientes.

El primer paso que se realizó fue un análisis de los productos, por ejemplo los ítems por categoría o por departamento, las ganancias por categoría, etc.

Una vez vistas las tendencias de nuestro dataset empezamos a tratar este para realizar el clustering. Pasamos las fechas a datetime y creamos columnas diferentes para día, mes y año.

Por falta de recursos decidimos trabajar con un sample, en nuestro caso de 200 registros.

Encodeamos el dataset y rellenamos los nulos para poder realizar el PCA y simplificar los datos. Con este nuevo dataset, aplicamos kmeans y buscamos el codo en la gráfica. Decidimos coger 5 clusters.

Una vez creados los clusters, pasamos a analizarlos:

Si nos fijamos en las ganancias, vemos que el grupo 1, que podríamos llamarlo top products, tiene unas ganancias medias muy superiores a los otros grupos. La ganancia mínima es superior al percentil 75 del siguiente grupo con más ganancias. El maximo quatriplica las

ganancias máximas del grupo "perseguidor". Además, todo esto sabiendo que el número de productos es muy inferior al de los otros grupos. De esta manera podemos pensar que este grupo de productos o tiene un precio muy superior al de los demás o que se vende mucho más, luego lo veremos.

Por otro lado, tenemos el grupo 0 que podríamos llamarlo "worst products" ya que vemos que siendo un número muy grande de productos, tenemos ganancias muy muy bajas, ya sea, igual que anteriormente, por precios bajos o por pocas ventas, luego lo veremos. Así, podemos ver que la media de este grupo no llega ni a la mitad del siguiente y su máximo es 1/3 de este.

En el resto de grupos, aunque seguimos viendo diferencias, no son tan exageradas y representativas como en estos dos.

Hablamos ahora del precio de ventas. Si nos fijamos en el grupo 0, el que aportaba menos ganancias, vemos que la media es la más baja de todas, aunque no hay una diferencia muy grande pero que tienen el producto más caro. Así, vemos que este grupo que hay un número muy elevado de productos tiene el producto más barato y más caro. Si hablamos del grupo 1 que era el que más ganancias daba, vemos que el precio del producto es similar a los otros grupos. En general, el sell price de los productos es similar en todos los clusters y no nos aporta mucha información, de manera que la diferencia de ganancias probablemente sea debida al número de ventas. Podemos ver que tanto medias como máximos y mínimos son similares en todos los clusters, destacando el grupo 4 que, a lo mejor, tiene unos precios más elevados.

Vamos a hablar ahora de número de ventas, donde deberíamos ver las diferencias más significativas y el porqué de la diferencia en cuanto a ganancias totales, ya que en el precio hay bastantes similitudes. Así, empezamos fijándonos en el grupo 1, top products, que tiene unas ganancias muy superiores a los demás. Vemos que esto se debe al número de ventas. Con un número de productos mucho menor a los demás (lo que indica que tenemos un

grupo reducido de productos estrella a los que tenemos que dar importancia), tenemos un número de ventas más elevado que en cualquier otro grupo. La media es muy superior a los siguientes grupos, 2 y 4. Tenemos una media que triplica al perseguidor (4). En cuanto a máximos y mínimos, vemos también que hay muchas diferencias. El grupo 1 tiene tanto máximos como mínimos de casi el doble del grupo 4. Así, vemos que el número de ventas de estos productos es muy superior al de los demás, dándonos unas ganancias tan grandes como las comentadas anteriormente.

En cuanto al grupo 0, en el que tenemos un gran número de productos, vemos que estos tienen unas ventas muy bajas y por eso tenemos unas ganancias tan pequeñas.

Con estos resultados sacamos las conclusiones de que las ganancias dependen sobre todo del número de ventas, que tenemos un grupo muy pequeño de productos extremadamente populares y una gran cantidad de productos que prácticamente no se venden.

Por otro lado, los otros 3 grupos siguen una tendencia creciente tanto de revenue como de ventas de 3, 2, a 4.

Tenemos que tener en cuenta así que productos son realmente importantes, grupo 1 sobre todo y 4 y que productos no se venden tanto, grupo 0.

Si hablamos de regiones por clusters, podemos ver que todos los grupos siguen una tendencia parecida y la mayoria de productos estan o se venden mayoritariamente en NY, seguido en menor medida en Philadelphia y con valores parecidos boston, pero no encontramos grandes diferencias entre ellos (más allá del número total, que ya sabemos que es por el número de productos totales "count" del cluster).

Así, afirmamos que los grupos se crean por el revenue que viene dado por el número de ventas, ya que no vemos grandes diferencias en el precio ni en la región en el producto.

Así, con precios similares, tenemos productos que se venden mucho y otros que se venden poco.

Sales forecasting

Con el objetivo de implementar el modelo predictor de ventas, el despliegue se llevaría a cabo de la siguiente manera:

1. Infraestructura de servidores:

   La cadena de supermercados debería contar con una infraestructura de servidores que respalde el despliegue de la aplicación. Esto podría implicar el uso de servidores locales en sus instalaciones o la elección de servicios en la nube, como Amazon Web Services (AWS), Google Cloud Platform (GCP) o Microsoft Azure. La elección dependerá de los recursos y requisitos específicos de la cadena de supermercados.

2. Almacenamiento de datos:

   Para el almacenamiento de datos, se recomendaría utilizar una base de datos centralizada donde se registren las ventas históricas y otros datos relevantes para el modelo predictor. Esta base de datos podría ser una solución como MySQL, PostgreSQL o MongoDB, dependiendo de las necesidades y preferencias de la cadena de supermercados.

   Además, el modelo predictor de ventas sería necesario almacenarlo en un servidor o un servicio en la nube accesible para la aplicación web.

3. Despliegue de la aplicación web en Docker:

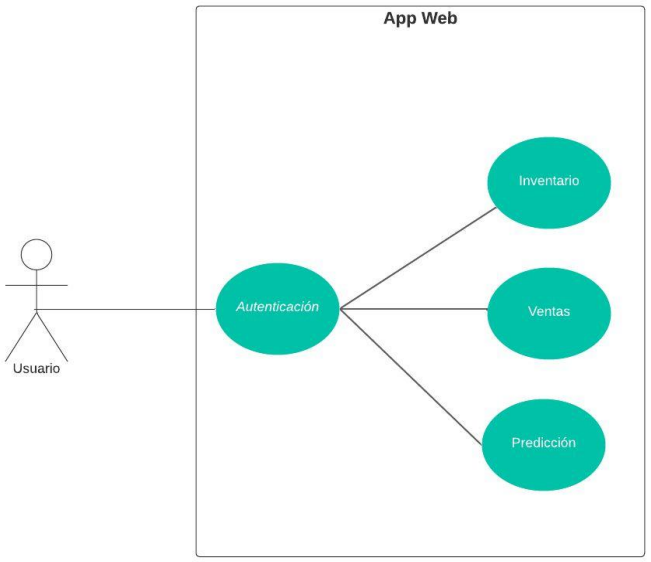
   Una vez que se haya configurado la infraestructura y se hayan preparado los datos y el modelo predictor, se procedería al despliegue de la aplicación web en Docker. Esto implica seguir los pasos mencionados anteriormente para crear un contenedor en Docker que incluya la aplicación web y todas sus dependencias, como el servidor web, el lenguaje de programación y las bibliotecas adicionales necesarias para su funcionamiento.

4. Integración de datos y modelo:

La aplicación web se conectaría a la base de datos centralizada para acceder a los datos históricos de ventas y utilizarlos como entrada para el modelo predictor. Esto permitiría generar pronósticos de ventas basados en datos actualizados. Además, la aplicación web establecería una conexión con el servidor o servicio que alberga el modelo predictor de ventas para enviar los datos relevantes y recibir las predicciones correspondientes.

5. Actualización y mantenimiento: Es importante tener en cuenta que tanto la base de datos como el modelo predictor de ventas pueden requerir actualizaciones periódicas. Esto puede incluir la incorporación de nuevos datos históricos a la base de datos, el reentrenamiento del modelo con los datos más recientes y la implementación de mejoras en el modelo. Estas actualizaciones y el mantenimiento general del sistema deberían ser considerados como parte del proceso de despliegue y gestión continua de la aplicación.

La aplicación web se compondría de una pantalla de autenticación y, una vez el usuario haya iniciado sesión, se accedería a una pantalla principal con las ventas realizadas en un cierto rango de tiempo. Desde esa pantalla, se podrá acceder a una segunda pantalla con el stock disponible, o a la pantalla de predicciones, en la que, tras subir los datos de input para el modelo, se devolvería un archivo con las predicciones de ventas devueltas por el modelo. En la imagen inferior se puede apreciar un esquema de la aplicación web.



Anexo: acceso a repositorio

En este anexo, se proporciona el enlace al repositorio de GitHub que contiene todo el código relacionado con el trabajo realizado. El código está disponible para su consulta y revisión en el siguiente enlace:

Enlace al Repositorio de Código en GitHub.

El repositorio incluye todos los archivos relevantes, estructuras de carpetas y versiones del código desarrollado durante el proyecto. Los lectores interesados pueden acceder al enlace para obtener una visión más detallada y profunda del trabajo realizado.