PRICESPY

La plataforma que combina analítica de datos e inteligencia artificial para proporcionar información valiosa sobre el mercado de viviendas en España y asi ayudar a la toma decisiones para determinar el valor de un inmueble.

- MEMORIA -

4Geeks Academy

Visita:
pricespy.onrender.com

ALUMNOS:

ABEL MONTIEL Y JAIME DODERO

TUTOR:

CARLOS VAZQUEZ

FECHA:

MAYO - SEPTIEMBRE 2023

PROMOCIÓN

1 DS.

DEDICATORIA:

Queremos expresar nuestro más profundo agradecimiento a 4Geek Academy por brindarnos una experiencia educativa excepcional que ha transformado nuestra comprensión de la ciencia de datos. Este viaje de aprendizaje ha sido enriquecedor y estimulante, y no habría sido posible sin la dedicación y el apoyo de esta institución.

Un agradecimiento especial se dirige a nuestro tutor, Carlos. Tu guía experta, paciencia y pasión por compartir conocimientos han sido invaluables para nuestro crecimiento en este campo. Tus consejos y orientación han iluminado nuestro camino en momentos de desafío, y tu compromiso con nuestro éxito ha sido inspirador.

También queremos reconocer a nuestros compañeros de bootcamp, cuya colaboración y amistad han enriquecido esta experiencia. Juntos, hemos explorado, aprendido y superado obstáculos, construyendo una comunidad de aprendizaje que perdurará mucho más allá de este bootcamp.

Este logro no es solo nuestro, sino de todos aquellos que han invertido su tiempo y esfuerzo en nuestro desarrollo. Estamos emocionados por lo que el futuro tiene reservado y llevaremos con nosotros las lecciones y conexiones que hemos obtenido en 4Geek Academy. **¡Gracias por hacer posible esta transformación!**

RESUMEN

Tras el desarrollo del bootcamp de Data Science de 4 Geek Academy, como proyecto de fin de bootcamp se ha procedido al desarrollo de un proyecto de aprendizaje automático utilizando información de Idealista, plataforma que conecta a personas y organizaciones con la finalidad de proceder a la compraventa y alquiler de inmuebles.

El objetivo principal de este proyecto ha sido aplicar los conocimientos adquiridos en el bootcamp de Data Science para analizar y modelar datos del mundo real. El equipo se ha propuesto utilizar técnicas de preprocesamiento de datos, visualización, análisis exploratorio y modelado de machine learning para extraer información valiosa y relevante de los datos de Idealista.

En la fase inicial, el equipo se ha sumergido en el proceso scraping, limpieza y preprocesamiento de datos. Esto ha involucrado la identificación y manejo de valores faltantes, la normalización de características y la transformación de datos en formatos adecuados para el análisis. Una vez que los datos han estado listos, se ha procedido a realizar un análisis exploratorio detallado para comprender mejor las tendencias, patrones y relaciones presentes en los datos.

La visualización de datos ha desempeñado un papel esencial en este proyecto. Utilizando gráficos y herramientas visuales, el equipo ha podido representar de manera efectiva la distribución de datos, las relaciones entre variables y otros aspectos importantes. Esto ha permitido una comprensión más profunda de la información subyacente y ha ayudado a guiar las decisiones en las etapas posteriores del proyecto.

La fase de modelado ha sido el corazón del proyecto, donde el equipo ha aplicado técnicas de machine learning para construir modelos predictivos. Se exploraron algoritmos como regresión, clasificación y agrupación para abordar diferentes objetivos, como predecir la demanda de ciertos tipos de oportunidades, clasificar oportunidades en categorías relevantes… hasta llegar a una conclusión y tomar la decisión de que modelo era el más clarificador y útil.

Después de entrenar y ajustar los modelos, el equipo ha evaluado su rendimiento utilizando métricas apropiadas para cada tipo de problema. Esto ha permitido una comparación objetiva de diferentes enfoques y ayudó a refinar aún más los modelos para obtener resultados más precisos.

INTRO

1.1. MOTIVACIÓN

Antes de la toma de decisiones sobre hacia donde enfocar este proyecto final, el equipo optó por observar cuales eran sus principales ventajas, como podía aportar la figura de un jurista y un diseñador en un proyecto de DataScience, combinando sus fortalezas y apoyándose en sus debilidades.

La primera observación fue la de aplicar nuestros conocimientos previos al Bootcamp al propio proyecto en combinación con todos aquellos adquiridos dentro de este.

Atendiendiendo que el jurista es experto en el sector urbanístico e inmobiliario, la propuesta de desarrollar un proyecto sobre el sector, es una oportunidad para la aplicación de esa dualidad conceptual: Derecho y Ciencia de Datos.

Para el diseñador, atendiendo a su versatilidad y su capacidad de adaptación, es una oportunidad perfecta para poder aplicar su dualidad de conocimientos tanto en lo obtenido en el Bootcamp de Ciencia de Datos como en la importancia de la presentación del proyecto, haciendo visual, intuitivo y sencillo la aplicación del modelo a usuarios sin conocimientos en la materia. (Cualidad esencial en un futuro para exponer los resultados a miembros del equipo y de la empresa que desconocen el sector.

La segunda observación era motivacional. Ambos miembros observan que la oportunidad es un reto para autosuperarse. En este caso, tras estudiar las diversas posibilidades de obtención de los datos, concluyen en que la web de mayor dificultad de la cual obtenerlos es www.idealista.com. (Mas adelante se observa las dificultades para el escrapeo)

La tercera: ampliar los conocimientos obtenidos. Si bien es de razón aplicar aquellos estudiados dentro del Bootcamp, la aventura de descubrir, errar y aplicar conocimientos nuevos es una labor esencial del Data Science. La forma en que se aplica en este proyecto es mediante el uso de librerías no conocidas, la búsqueda de código útil y la lectura de ideas de otros desarrolladores.

1.2. OBJETIVOS

El objetivo del presente proyecto, no es más que mostrar tanto los conocimientos adquiridos en el trayecto del Bootcamp como en mostrar las capacidades del equipo.

Objetivos humanos:

- Desarrollo de la cooperación y coordinación del equipo
- Fomentar la comunicación abierta y el intercambio de ideas entre los miembros del equipo.
- Trabajar en la construcción de relaciones positivas y de confianza con los colegas.
- Participar activamente en reuniones de equipo y contribuir con aportes significativos.

División de tareas, colaboración y puesta en común de ideas:
- Identificar las fortalezas y habilidades individuales de los miembros del equipo para asignar tareas de manera efectiva.
- Colaborar con otros en proyectos, compartir conocimientos y experiencias para lograr un enfoque multidisciplinario.
- Participar en sesiones de lluvia de ideas y aportar soluciones creativas a los desafíos del equipo.

Manejo de conflictos y resolución de problemas:
- Abordar los desacuerdos con empatía y una actitud constructiva para encontrar soluciones mutuamente beneficiosas.
- Utilizar habilidades de comunicación efectiva para resolver conflictos y prevenir malentendidos.
- Identificar las causas subyacentes de los problemas y trabajar en soluciones duraderas.

Gestión del tiempo:
- Establecer prioridades claras y elaborar planes para gestionar eficazmente el tiempo y las tareas.
- Utilizar herramientas como calendarios y listas de tareas para mantenerse organizado y cumplir plazos.
- Aprender a delegar tareas cuando sea necesario para optimizar la eficiencia personal y del equipo.
- Análisis y control de estrés ante situaciones de alto rendimiento:
- Desarrollar estrategias para manejar la presión en momentos de alta demanda, manteniendo la claridad mental.

Objetivos materiales:

Manejo de librerías esenciales para Data Science:
- Utilizar bibliotecas como Pandas para la manipulación y análisis de datos.
- Emplear bibliotecas de visualización como Matplotlib o Seaborn para representar datos de manera efectiva.
- Aplicar librerías de aprendizaje automático como Scikit-Learn para construir modelos predictivos.

Obtención y análisis de información relevante:
- Recolectar datos de diversas fuentes, como bases de datos, APIs o archivos CSV.
- Realizar limpieza y preprocesamiento de datos para garantizar su calidad y adecuación para el análisis.
- Realizar análisis exploratorio de datos (EDA) para identificar patrones, tendencias y posibles relaciones en los datos.

Desarrollo de modelos útiles, funcionales y prácticos:
- Diseñar modelos de aprendizaje automático adaptados al problema en cuestión, ya sean modelos de regresión, clasificación u otro tipo. Ajustar hiperparámetros y evaluar el rendimiento de los modelos utilizando métricas relevantes.
- Implementar técnicas de validación cruzada para evaluar la capacidad de generalización de los modelos.

Control del flujo de código, manejo de versiones.
- Utilizar control de versiones, como Github, para rastrear cambios en el código y colaborar de manera efectiva en proyectos.
- Organizar el código en funciones y clases para mejorar la modularidad y reutilización.
- Aplicar buenas prácticas de programación para garantizar un código legible y mantenible.

WEB SCRAPING

2.1. EL RETO DE IDEALISTA

La obtención de los datos de la página web de Idealista ha representado un reto significativo debido a la complejidad de los métodos de obstrucción implementados por el sitio. La decisión de abordar este desafío se basó en el reconocimiento de que los datos disponibles en Idealista son los más completos y fiables del mercado además de la complejidad de su obtención. La dificultad inherente en el scraping de datos ha sido un factor motivador clave, ya que la superación de estas barreras técnicas ofrecía una oportunidad para demostrar habilidades técnicas y creatividad en el proceso.

La selección de la página web de Idealista como fuente de datos no solo tenía en cuenta su relevancia para el proyecto, sino también la oportunidad de enfrentar y resolver problemas técnicos complejos. La extracción de datos de una plataforma con medidas de seguridad avanzadas no solo ha requerido de conocimientos de web scraping, sino también la capacidad de adaptarse y evolucionar a medida que las tácticas de obstrucción cambiaban con el tiempo. (Se desarrolla mas delante)

En última instancia, el éxito en la obtención de datos de Idealista no solo proporcionaría información valiosa para el análisis, sino que también destaca la habilidad y determinación para superar desafíos técnicos en la búsqueda de soluciones innovadoras.

Por ende, la suma de reto, calidad de datos y confianza de las capacidades del equipo, son la razón esencial de porque idealista.

2.2. METODO DE EXTRACCIÓN

Siendo plenamente conscientes de las diferentes metodologías disponibles para el scraping de datos, como el uso de APIs y la programación directa de código, se optó por utilizar Python como base de extracción. Esta elección ha sido el resultado de un proceso de evaluación en el que se han considerado las distintas opciones disponibles.

La razón principal detrás de la elección ha sido la falta de una API que satisficiera todos los requisitos: fiabilidad, precisión y disponibilidad gratuita. Aunque las APIs podrían haber sido una opción más sencilla para la obtención de datos estructurados, se encontraron dificultades para hallar una que cumpliera con todas las necesidades específicas. Las APIs a menudo han tenido limitaciones en términos de la cantidad de datos que se pueden obtener, o han requerido un pago por acceso a ciertos niveles de información.

Dado que la prioridad era obtener un conjunto completo y detallado de datos para el proyecto, se ha decidido buscar y desarrollar código de scraping utilizando Python. Esta elección ha permitido tener un control total

sobre el proceso de extracción y la flexibilidad necesaria para ajustar el código a medida que han surgido desafíos y obstáculos durante el proceso.

La búsqueda y el desarrollo de código se han convertido en componentes esenciales de esta fase del proyecto. Se ha invertido unas semanas en investigar las mejores prácticas de web scraping y en comprender las estructuras de la página web de Idealista. Esto nos ha permitido diseñar y programar un script de scraping eficiente y efectivo que ha podido navegar por el sitio, interactuar con sus elementos y extraer la información requerida de manera sistemática.

En última instancia, la elección de utilizar Python y desarrollar parte del código de scraping refleja la determinación de superar los desafíos inherentes a la obtención de datos de una fuente con obstáculos técnicos. Esta experiencia no solo ha enriquecido nuestro conocimiento y habilidades técnicas, sino que también ha permitido obtener un conjunto completo de datos que servirá como base sólida para el análisis y los resultados del proyecto.

Al código, se le ha sumado el uso de la API HMA. Esta tiene como funcionalidad la rotación de VPN/IP de forma automática por el tiempo en que se determine. En este caso se ha puesto a rotar cada 5 minutos por regla general, aunque a veces por dar celeridad entre 1 y 3 minutos. Esta API es gratuita los 7 primeros días, por lo que se ha procedido a scraping de firma completa e intensa en ese periodo de tiempo, con el terminal scrapeando dia y noche.

2.3. PRINCIPALES OBSTÁCULOS

Ciertamente, Idealista y otros sitios web suelen implementar diversas medidas para evitar o dificultar la extracción automatizada de datos de sus plataformas. Estas medidas, conocidas como técnicas de obstrucción o anti-scraping, están diseñadas para proteger la integridad de los datos y prevenir el uso no autorizado de la información disponible en el sitio. Algunas de las técnicas comunes que Idealista utiliza incluyen:

PROBLEMA	SOLUCIÓN
Límite de velocidad: Evitar que bots realicen gran cantidad de peticiones.	División del flujo de scraping en código, dividiendo el trabajo. Se saca primero las Ids de la vivienda y de forma posterior se analizan una a una, de esta forma, la extracción toma mayor tiempo y por tanto, parece más real.
Captcha: medida común para detectar y bloquear bots automatizados. Teniendo en cuenta de la dificultad de la creación de código para que resuelva un Captcha, las medidas de solución del conflicto deben de llevarse por otro enfoque más sencillo, ya que son conscientes en idealista de lo poderosos que son los Captchas ante el scraping involuntario.	Utilización de VPN/IP con rotación temporal (de entre 1 y 5 minutos) Cuando Idealista detecta un comportamiento extraño, salta el Captcha. Como el código cuando detecta desconexión con idealista se detiene 10 segundos y continua (y así sucesivamente hasta reconectar), en cuanto el VPN/IP se cambia, idealista intuye que es una nueva conexión y restaura la página por donde el código solicita, solucionándose de esta manera el problema.
Sesiones de usuario: usuario para rastrear la actividad en el sitio y rastrear alto volumen de solicitudes provenientes de una misma dirección IP.	VPN/IP rotativo. Se ha utilizado el programa HMA (Programa de rotación de VPN/IP por tiempo).
Limitación en la estructura de páginas: Idealista no permite observar más de 60 páginas por búsqueda. No existe la pagina 61 de idealista, por lo que el scraping a gran escala no puede desarrollarse. Esto implica que, cada página de idealista contiene 30 viviendas, y su página máxima es la 60, por lo que por scraping solo puede sacarse 1800. A partir de entonces comienza a duplicar viviendas o a dar datos falsos (Error habitual en una API).	Scrapear en tandas de 1800 máximo, limitando el scraping por metraje, precios… Por tanto, para scrapear las casi 40.000 propiedades se ha requerido tandas de scraping de más de 45 veces. (Ya que no siempre se podía limitar a 1800 viviendas, porque existen lugares donde no hay tantas, o al contrario, duplicar las tandas porque algún sitio tiene más de 1800).
Análisis del comportamiento: Idealista monitorea patrones de comportamiento de las solicitudes. Si detecta un comportamiento que parece automatizado, como acceder a múltiples páginas en secuencia en muy poco tiempo, el sitio puede tomar medidas.	VPN/IP rotativo. Se ha utilizado el programa HMA (Programa de rotación de VPN/IP por tiempo) y uso de Time.Sleep para simular comportamiento humano.
Bloqueo de direcciones IP: Si un servidor realiza demasiadas solicitudes en un corto período de tiempo, el sitio web puede bloquear temporalmente o permanentemente la dirección IP desde la que provienen las solicitudes.	VPN/IP rotativo. Se ha utilizado el programa HMA (Programa de rotación de VPN/IP por tiempo) y uso de Time.Sleep para simular comportamiento humano.
Análisis del comportamiento: Idealista monitorea los patrones de comportamiento de las solicitudes. Si detecta un comportamiento que parece automatizado, como acceder a múltiples páginas en secuencia en muy poco tiempo, salta el Captcha.	VPN/IP rotativo. Se ha utilizado el programa HMA (Programa de rotación de VPN/IP por tiempo) y uso de Time.Sleep para simular comportamiento humano.

2.4. CODIGO DE SCRAPING

ATENDIENDO A LA importancia del método de scrapeo, se comparte las características principales del cogido usado.

LIBRERIAS

- Importamos la biblioteca ‘requests’ para hacer solicitudes web.
- Importamos la clase ‘BeautifulSoup’ de la biblioteca ‘bs4’ para el análisis HTML y scraping.
- Importamos la biblioteca ‘random’ para generar valores aleatorios.
- Importamos la biblioteca ‘time’ para manejar el tiempo.
- Importamos la biblioteca ‘pandas’ para el manejo de datos en tablas.
- Importamos la biblioteca ‘numpy’ para operaciones numéricas.
- Importamos el módulo ‘webdriver’ de la biblioteca ‘selenium’ para automatizar el navegador y scraping.
- Importamos la clase ‘By’ para localizar elementos web.
- Importamos la clase ‘Keys’ para simular pulsaciones de teclas.
- Importamos ‘undetected_chromedriver’ para usar el controlador de Chrome no detectado y que Crhome crea que somos personas.
- Destacar el uso de esta librería, la cual es desconocida para gran parte del sector y de la cual no existe información en Chat GPT por ser posterior a 2021.
- Esta librería únicamente funciona con Crhome 1.15 o inferior (Actualmente Crhome está en 1.16), por lo que se requiere de la modificación de la instalación de Crhome en pc para que no se actualice de forma automática.

```
import requests
from bs4 import BeautifulSoup as bs
import random
import time
import pandas as pd
import numpy as np
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import undetected_chromedriver as uc
```

APERTURA AUTOMÁTICA DE NAVEGADOR

- Mediante el uso de la librería undetectd_crhomedriver podemos abrir y acceder a crhome y a la url que deseemos scrapear de forma automarica.

```
browser = uc.Chrome()
url = “Colocar aqui url scrap inicial”

browser.get(url)
```

AUTOCLICK COOKIES

- Se utiliza esta línea de código para localizar elemento en la página web utilizando XPath y hacer clic en él.
- Se recomienda comentar el código (#) ya que en caso de no aparecer el botón tiende a paralizarse el código dando error (Esta línea es opcional dependiendo del comportamiento de la propia web)

```
browser.find_element(“xpath”,’//*[@
id=”didomi-notice-agree-button”]’).
click()
```

OBTENCIÓN DEL HTML

```
html = browser.page_source
html
```

CONVERSIÓN DE HTML A ELEMENTO MANIPULABLE

```
soup = bs(html, ‘lxml’)
soup
```

EXTRACCIÓN DEL TÍTULO

```
titulo = soup.find(‘span’,{‘class’:’main-
info__title-main’}).text
titulo
```

EXTRACCIÓN DE LA LOCALIZACIÓN

```
localizacion = soup.
find(‘span’,{‘class’:’main-info__title-mi-
nor’}).text.split(‘,’)[0]
localizacion
```

EXTRACCIÓN DEL PRECIO

```
precio = int(soup.find(‘span’,{‘class’:’txt-
bold’}).text.replace(‘.’,’’))
precio
```

EXTRACCIÓN DE DATOS EN GRUPO

- En idealista, los datos como habitaciones, precio… se pueden extraer de forma individualizada o de forma conjunta.
- Para acelerar el proceso, se extrae en forma conjunta y se divide posteriormente (Puede usarse Excel o Python para la división / creación de nuevas columnas)
- Caracteristivas básicas (habitaciones, precio…)
- Caracteristicas extras (Ascensor, piscina…)

```
c1 = soup.find('div',{'class':'details-property'}).
find('div',{'class': 'details-property-feature-one'})
caract_basicas = [caract.text.strip() for caract in c1.find_all('li')]
c2 = soup.find('div',{'class':'details-property'}).
find('div',{'class': 'details-property-feature-two'})
caract_extra = [caract.text.strip() for caract in c2.find_all('li')]
ubicacion = soup.find('div',{'id':'headerMap'})
ubicacion_desg = [zona.text.strip() for zona in ubicacion.
find_all('li')]
```

CARGA DE NUEVA URL EN EL NAVEGADOR

- Ya se ha marcado los datos que se desean sacar por vivienda. Mediante el uso de la vivienda ejemplo anterior, se procede a cargar URL de lugar de extracción.

```
url_grande = 'colocar aquí url'
browser.get(url_grande)
```

NUEVA OBTENCIÓN DEL CÓDIGO FUENTE HTML Y CONVERSIÓN.

```
html = browser.page_source
soup = bs(html, 'lxml')
soup
```

SELECCIÓN DE ELEMENTOS

- Se busca la obtención de elementos dentro de elemento main en el HTML de la web, para poder iterar posteriormente por dichos elementos y extraer información.
- Establecemos variable

```
articles = soup.find('main',{'class':'listing-items'}).find_
all('article')
```

SELECCION ID DE INMUEBLES Y ELIMINACION DE NONES.

- Se busca la obtención de elementos del atributo data-adid de serie de elementos article y almacenados en la lista id_muebles
- Se configura codigo para la eliminación de datos Nones.

```
id_muebles = [article.get('data-adid') for article in articles]
id_muebles = [muebles for muebles in id_muebles if
muebles is not None]
```

EXTRACCION DE IDS.

- Esta parte del codigo es fundamental ya que debemos poner la url inicial de la que queremos sacar información.
- El elemento -{x} es esencial ya que es el iterador de pagina.
- La estructura de idealista es pagina-(numero). En esta caso x sera el numero de la pagina que vayamos a scrapear.
- Otro elemento esencial de este codigo es el time.sleep
- Nos permite simular el click humano para evitar que idealista detecte la acción del bot.
- Otro elemento indispensable es except Exception as e:
- Es útil ya que en caso de que idealista detecte que la acción llevada a cabo está siendo ejecutada por un bot, saltará el Captha. Entonces el código se detiene durante un periodo de tiempo de 10 segundos e intenta reconectar a idealista. Si la situación persiste entra la magia de HMA VPN.
- Este programa al cambiar de VPN/IP cada X minutos (según se programe) en el momento en que cambie de VPN/IP y el código lo detecte, continua extrayendo información ya que idealista detecta que es otra conexión.

```
busqueda = 'titulo'
busqueda = busqueda.replace(' ', '-')

x = 1
ids = []

while True:
    try:
        url = f"link_web_scrapear-{x}.htm"

        browser.get(url)

        time.sleep(random.randint(9, 11))

        try:
            # Manejar la ventana de cookies si aparece
            cookie_button = browser.find_element("xpath",
'//*[@id="didomi-notice-agree-button"]')
            cookie_button.click()
        except:
            pass

        html = browser.page_source
        soup = bs(html, 'lxml')

        pagina_actual = int(soup.find('main', {'class': 'listing-
items'}).find('div', {'class': 'pagination'}).find('li', {'class':
'selected'}).text)

        if x == pagina_actual:
            articles = soup.find('main', {'class': 'listing-items'}).
find_all('article')
        else:
            break

        for article in articles:
            id_muebles = article.get('data-adid')
            ids.append(id_muebles)
```

```
            time.sleep(random.randint(1, 3))
            print(id_muebles)

        ids = [muebles for muebles in ids if muebles is not
None]

    except Exception as e:
        print(f"Se produjo un error: {e}")
        print("Esperando 10 segundos antes de intentar nue-
vamente...")
        time.sleep(10)  # Esperar 10 segundos antes de
volver a intentar

    x = x + 1
```

TABLA DE IDS Y GUARDADO

```
ids_casas = pd.DataFrame(ids)
ids_casas.columns = ['id']
ids_casas.to_csv('ids_madrid_2300000m_a_3000000m.
csv',index = False)
```

CREACIÓN VACÍA DE SERIE

```
casas = pd.Series()
```

FUNCIÓN DE ANÁLISIS DE INFORMACIÓN

```
def parsear_inmueble(id_inmueble):

    print( '\n Casa numero: ' + id_inmueble)

    url = "https://www.idealista.com/inmueble/" + id_inmueble + "/"

    browser.get(url)

    html = browser.page_source

    soup = bs(html, 'lxml')

    titulo = soup.find('span',{'class':'main-info__title-main'}).text

    print('\n Titulo: ' + titulo)

    localizacion = soup.find('span',{'class':'main-info__title-
minor'}).text.split(',')[0]

    print('\n Localizacion: ' + localizacion)
    precio = int(soup.find('span',{'class':'txt-bold'}).text.
replace('.',''))

    c1 = soup.find('div',{'class':'details-property'}).
find('div',{'class': 'details-property-feature-one'})

    caract_basicas = [caract.text.strip() for caract in
c1.find_all('li')]

    #print('Caracteristicas basicas:' + caract_basicas)

    c2 = soup.find('div',{'class':'details-property'}).
find('div',{'class': 'details-property-feature-two'})

    caract_extra = [caract.text.strip() for caract in c2.find_all('li')]

    #print('Caracteristicas extras:' + caract_extra)

    casas['titulo'] = titulo

    casas['localizacion'] = localizacion



    casas['precio'] = precio

    casas['caracteristicas_basicas'] = caract_basicas

    casas['caracteristicas_extras'] = caract_extra

    df_casas = pd.DataFrame(casas)

    return(df_casas.T)
```

ITERACIÓN Y RECOPILACIÓN DE LA FUNCIÓN PARSEAR INMUEBLE

- Es esencial esta parte ya que es la que va a proceder a scrapear los datos individuales de cada vivienda.
- Al igual que en el paso 14 contiene Time.Sleep y una excepción en caso de perder conexión.

```
df_casas = parsear_inmueble(ids_casas.iloc[0].id)

for i in range(1, len(ids)):
    try:
        df_nuevo = parsear_inmueble(ids[i])
        df_casas = pd.concat([df_casas, df_nuevo])
    except Exception as e:
        print(f"Error en la iteración {i}: {e}")
        print("Esperando 10 segundos antes de reanudar...")
        time.sleep(10)  # Pausa durante 10 segundos
        continue

    time.sleep(random.randint(4, 6))
```

GUARDADO DEL DATAFRAME

```
df_casas.to_csv('nombre_del_dataframe.csv', index =
False, sep = ';', encoding = 'utf-16')
```

CIERRE DEL NAVEGADOR

```
browser.close()
```

HMA (HideMyAss) es una aplicación de VPN (Red Privada Virtual) que ofrece servicios de privacidad en línea y seguridad en internet. Una VPN es una herramienta que permite crear una conexión segura y cifrada entre tu dispositivo y un servidor remoto, lo que te permite navegar por internet de manera más segura y anónima al ocultar tu dirección IP real y cifrar tus datos.

Se ha usado continuamente en el scraping para evitar que la detección de idealista bloquease el scrapeo. Teniendo en cuenta las grandes medidas anti-scrapeo de la web, dicho programa ha sido de gran utilidad, además de simplificar el trabajo.

Aquí hay algunos aspectos importantes sobre HMA:

- **Privacidad y Anonimato:** HMA oculta tu dirección IP real y cifra tu tráfico de internet, lo que dificulta que terceros rastreen tu actividad en línea.

- **Ubicación Virtual:** HMA te permite elegir servidores en diferentes ubicaciones de todo el mundo.

- **Cifrado Seguro:** Utiliza protocolos de cifrado seguros para proteger tu información.

- **Acceso a Contenido Restringido:** Muchas veces, las VPN permiten a los usuarios acceder a servicios o contenido que pueden estar bloqueados.

- **Seguridad en Redes Públicas:** Cuando te conectas a una red Wi-Fi pública, tus datos pueden ser más vulnerables. Una VPN como HMA cifra tu tráfico, lo que lo hace más seguro incluso en redes no seguras.

- **Rotación de IP:** HMA es conocida por ofrecer rotación de direcciones IP, lo que significa que puedes cambiar tu dirección IP periódicamente. Esto puede ser útil para mantener un nivel adicional de anonimato, dificultar el seguimiento y la realización de Scraping.

- En este caso, se puso una rotación de entre 1 – 5 minutos, por lo que, si era detectado el scraping, pasado ese tiempo el codigo podía continuar scrapeando sin problemas.

- **Aplicaciones Multiplataforma:** HMA generalmente ofrece aplicaciones para una variedad de plataformas, como Windows, macOS, iOS, Android y más.

- **Rendimiento y Velocidad:** La velocidad de conexión y el rendimiento pueden variar según el servidor al que te conectes y la carga en ese servidor. Algunos usuarios pueden experimentar una ligera disminución en la velocidad debido al cifrado.

- **Política de Registro:** Es importante verificar la política de registros de cualquier proveedor de VPN. Algunos mantienen registros mínimos, mientras que otros pueden almacenar cierta información sobre tu actividad en línea.

- **Costo:** HMA generalmente ofrece diferentes planes de suscripción con distintos niveles de características y acceso a servidores. Los precios pueden variar según la duración de la suscripción.

- **En este caso se ha utilizado su versión gratuita de 7 días.**

DATA

3.1. DATOS OBTENIDOS

SE HA RECOPILADO y analizado una extensa cantidad de datos relacionados con el mercado inmobiliario en España. Se ha obtenido información detallada sobre más de 160,000 inmuebles en todo el país. Estos datos han incluido valiosa información sobre precios, ubicación, tipo de propiedad y otras variables clave que han sido fundamentales para haber comprendido el mercado inmobiliario (De f orma precisa en las ciudades mencionadas y general en el resto).

Los proyectos que se han desarrollado son:

Tasador de inmuebles de Malaga y Madrid
- Usando la población complete de cada ciudad a fecha de escrapeo

Tasador de inmuebles de todas las provincias de España
- Usando muestra de 1800 inmuebles por provincia

Comparador de descripciones
- Verifica la veracidad o fraude de anuncio inmobiliario

En el caso de Málaga y Madrid se han obtenido CSV cuya suma asciende aproximadamente 80.000 viviendas divididos en dos archivos CSV para un análisis independiente de cada provincia.

Cabe destacar la selección aleatoria de 1800 inmuebles como muestra representativa de cada provincia española, creando diversos CSV que en suma generan una cantidad aproximada de: 85.000 viviendas.

De esta manera, al diversificar los datos en diversos proyectos, ha sido mas sencillo poder manejarlos y ejecutar el código, consiguiendo agilizar el desarrollo y ejecución del proyecto en su conjunto.

3.2. CALIDAD VS CANTIDAD

Además, se ha desarrollado un modelo de NPL utilizando las descripciones obtenidas en el escrapeado, aprovechando todo el potencial de los datos para enriquecer el proyecto. (Se procede al desarrollo minucioso de los proyectos mencionados más adelante).

Uno de los principales problemas de la obtención de datos es la calidad de estos. Unos datos poco precisos o mal limpiados aun existiendo un número excesivo de estos generalmente producen un modelo de bajo rendimiento. Por tanto y en aras de crear modelos lo más efectivos posibles, se ha procedido a dos limpiezas de datos: una primaria (mediante Excel) y una secundaria (mediante uso de pandas y otras librerías).

Para enriquecer esta experiencia, los proyectos de Málaga y Madrid han sido filtrados de una forma más exhaustiva

por Excel y el proyecto enfocado a la tasación de todas las provincias tiene un mayor enriquecimiento en cuanto al filtrado con código, de esta manera se demuestra conocimiento mediante el uso de ambas herramientas.

Los proyectos de Málaga y Madrid al usar población completa de ambas ciudades son mas precisos y exhaustivos, obteniendo un buen rendimiento el modelo.
Usando el proyecto de tasación de todas las provincias 1800 viviendas como muestra, tu precisión es mas relativa, pero siendo muy útil para una concepción genérica de los valores de los inmuebles de dichas provincias.

En cuanto al modelo NPL de comparador de descripciones, se ha usado 1000 descripciones verdaderas (obtenidas del escrapeo previo) y otras 1000 falsas creadas a través de inteligencia artificial (Debido a la inexistencia de base de datos que contenga descripciones fraudulentas). Mediante un numero de 2000 se ha buscado que el modelo esté lo más balanceado posible.

MADRID - MÁLAGA

4.1. PORQUE ESTAS CIUDADES

LA ELECCIÓN DE ambas ciudades tiene dos razones principales:

1º: Son las ciudades de nacimiento de los miembros del equipo
2º: Son ciudades de gran relevancia inmobiliaria

Mediante la selección de estas, se procede al desarrollo tanto a la comparativa de precios (mediante el estudio de EDA) como el desarrollo de los modelos de Machine Learning más adecuados para poder desarrollar un tasador de viviendas en cada ciudad.

Los datos de los csv para el desarrollo del modelo han sido:

- **Columnas Categóricas:** Source.Name, Titulo, Tipo de inmueble, Descripción, Teléfono, Dirección, Subtitulo, Nombre del vendedor, Tipo de vendedor, Barrio, Municipio, Distrito, Ascensor (Sí/No), Obra nueva (Sí/No), Piscina (Sí/No), Terraza (Sí/No), Parking (Sí/No), Parking incluido en el precio (Sí/No), Aire acondicionado (Sí/No), Trastero (Sí/No), Jardín (Sí/No)
- **Columnas Numéricas:** Id del anuncio, Precio, Habitaciones, Precio antes de rebaja, euros/m2, Metros cuadrados construidos, Baños, Planta

4.2. ANALISIS GENERICO DEL SECTOR INMOBILIARIO EN LAS CIUDADES

Antes de iniciar el análisis, recordar que, para mayor precisión y conocimiento, se invita al lector a visitar los repositorios donde se han desarrollado los modelos, en el cual dispone con mayor detalle las conclusiones obtenidas.
A nivel comparativo, estas ciudades tienen grandes similitudes y grandes diferencias:

Similitudes:

- **Demanda Constante:** Tanto Málaga como Madrid son destinos populares para turistas, trabajadores y estudiantes, lo que genera una demanda constante de viviendas. La presencia de industrias como el turismo, la tecnología y los servicios financieros también contribuye a una demanda sostenida de propiedades.

- **Crecimiento Económico:** Ambas ciudades han experimentado un crecimiento económico significativo en los últimos años. Madrid, como la capital de España, es un centro financiero y de negocios importante, mientras que Málaga ha experimentado un auge en la industria tecnológica y de startups.

- **Inversión Extranjera:** La inversión extranjera en el sector inmobiliario ha sido un factor clave. Muchos inversionistas internacionales ven a España, y en particular a ciu-

dades como Madrid y Málaga, como lugares atractivos para invertir debido a la estabilidad política y económica.

- **Turismo**: Ambas ciudades atraen a una gran cantidad de turistas cada año, lo que impulsa la demanda de alquileres vacacionales y propiedades de inversión. El turismo ha mantenido un flujo constante de ingresos para el sector.

- **Calidad de Vida:** Málaga y Madrid ofrecen una alta calidad de vida, con buenos servicios públicos, atención médica de calidad, educación y una amplia gama de actividades de ocio y cultura. Esto atrae a residentes nacionales e internacionales.

- **Diversificación:** Ambas ciudades tienen una oferta inmobiliaria diversificada que incluye propiedades residenciales, comerciales y de oficinas. Esto permite al mercado inmobiliario adaptarse a diferentes ciclos económicos.

- **Políticas Gubernamentales:** Las políticas gubernamentales en España han sido favorables al mercado inmobiliario en términos de incentivos fiscales y facilidades para la inversión extranjera.

- **Infraestructura:** Las inversiones en infraestructura, como aeropuertos modernizados y sistemas de transporte público eficientes, hacen que ambas ciudades sean más atractivas para inversores y residentes.

- **Cultura de la Propiedad:** En España, la propiedad de vivienda es culturalmente importante, lo que fomenta la inversión y la estabilidad en el mercado.

- **Potencial de Crecimiento:** Ambas ciudades tienen un potencial de crecimiento económico y urbanístico, lo que aumenta la confianza de los inversores a largo plazo. Ubicación Geográfica: Málaga se encuentra en la Costa del Sol, en el sur de España, mientras que Madrid está en el centro del país. Esto afecta a la demanda de propiedades, ya que Málaga es un destino turístico costero, mientras que Madrid es un centro financiero y de negocios.

Diferencias:

- **Tipo de Propiedades:** Málaga tiene una gran cantidad de propiedades orientadas al turismo, como apartamentos vacacionales, mientras que Madrid tiende a tener una mayor variedad de propiedades, incluyendo residenciales, comerciales y de oficinas.

- **Precio de la Vivienda:** En general, los precios de la vivienda en Madrid tienden a ser más altos que en Málaga debido a su posición como capital y centro económico de España, salvo en la parte de lujo, donde Málaga se lleva la victoria atendiendo a la fortaleza de Marbella y zonas colindantes.

- **Demanda de Alquileres Vacacionales:** Málaga experimenta una fuerte demanda de alquileres vacacionales debido a su atractivo turístico, lo que puede influir en los precios y la disponibilidad de propiedades para alquilar.

- Dinámica de Inversión Extranjera: Madrid atrae una gran inversión extranjera debido a su importancia como centro financiero, mientras que Málaga tiende a recibir inversión extranjera relacionada con el turismo y la industria inmobiliaria.

- **Tamaño y Densidad:** Madrid es una ciudad mucho más grande y densamente poblada que Málaga, lo que influye en la oferta y la demanda de propiedades.

- **Perfil de Compradores y Residentes:** Málaga atrae a una población más diversa de compradores y residentes, incluyendo turistas, expatriados y jubilados extranjeros. Madrid tiene una población más cosmopolita y diversa en términos de negocios y empleo.

- **Crecimiento Económico Específico:** Madrid se destaca por su crecimiento económico en sectores financieros, tecnológicos y empresariales, mientras que Málaga ha experimentado un crecimiento relacionado con la tecnología y startups en los últimos años.

- **Oferta Cultural y de Ocio:** Madrid ofrece una amplia gama de opciones culturales y de entretenimiento, incluyendo teatros, museos y una vida nocturna vibrante. Málaga también ofrece una oferta cultural, pero su enfoque suele estar más relacionado con el turismo y la costa.

- **Acceso a Aeropuertos:** Madrid cuenta con el Aeropuerto Adolfo Suárez Madrid-Barajas, uno de los principales aeropuertos de Europa, mientras que Málaga tiene el Aeropuerto de Málaga-Costa del Sol. La conectividad aérea puede influir en la demanda de propiedades por parte de viajeros internacionales.

Este análisis nos permite que, cuando se observe los repositorios, entendamos tanto las similitudes como las diferencias existentes entre ambas ciudades.

4.3. MODELO UTILIZADO

Primero se debe destacar el contexto del proyecto. Se realizaron investigaciones y análisis para determinar la metodología más adecuada para lograr este objetivo.

Acto seguido se procedió a la elección del Modelo: Tras una revisión exhaustiva de las técnicas de modelado es-

tadístico y de aprendizaje automático disponibles, se optó por utilizar un modelo Random Forest debido a sus ventajas en términos de precisión, flexibilidad y capacidad para manejar una amplia variedad de características de las viviendas.

Destacar las características Relevantes: Se recopilaron y prepararon los datos. Estos datos incluían información sobre las viviendas, como su ubicación, tamaño, número de habitaciones, características adicionales (como piscina o garaje), y el precio de venta. Estos atributos se utilizaron como variables predictoras en el modelo. Se han utilizado 14 variables predictivas de las cuales, las de mayor relevancia son: precio/m2 y metros construidos. Las restantes tienen relación, pero menor significativamente.

Entrenamiento del Modelo: El modelo Random Forest se entrenó utilizando un conjunto de datos históricos de viviendas que incluía información detallada sobre propiedades y sus valores de venta reales. Durante esta fase, se ajustaron los hiperparámetros del modelo para optimizar su rendimiento.

Validación y Evaluación: El modelo se sometió a un proceso de validación cruzada y se utilizaron métricas de evaluación, como el error medio absoluto (MAE) o el error cuadrático medio (MSE), para medir su capacidad para predecir con precisión los valores de viviendas en el conjunto de prueba.

Implementación: Una vez que se confirmó la eficacia del modelo, se procedió a su implementación en un entorno de producción, lo que permitió su uso para tasar automáticamente viviendas.

Por tanto, los resultados: que se obtuvieron son muy positivos con el modelo Random Forest, que demostró ser capaz de proporcionar valoraciones de viviendas precisas y confiables. Esto contribuyó al éxito del proyecto y a su utilidad en el ámbito inmobiliario.

NLP FRAUDES

5.1. EL PORQUE DE ESTE MODELO

DESPUÉS DE HABER realizado el análisis inicial de las descripciones de las viviendas obtenidas a través del scraping, se identificaron patrones sospechosos en ciertos anuncios. Esto generó nuestro interés en abordar de manera más efectiva el problema de las estafas en el sector inmobiliario. La creación del detector de descripciones fraudulentas se ha convertido en un proyecto crucial para abordar esta creciente preocupación

5.2. EJECUCCION DEL MODELO

Para desarrollar el detector de fraudes, se procedió a recopilar un extenso conjunto de datos que incluía tanto descripciones legítimas como potencialmente engañosas (En este caso, han sido obtenidas mediante el uso de CHAT GPT debido a la ausencia de datos). Este conjunto de datos se ha utilizado para entrenar y refinar el modelo de procesamiento del lenguaje natural (NPL) que ha sido el núcleo de nuestro sistema de detección. El objetivo ha sido capacitar al modelo para identificar de manera precisa las características lingüísticas que indicaban posibles intentos de estafa en las descripciones de las viviendas.

Una vez que el modelo NPL se ha encontrado en una fase avanzada de desarrollo, hemos comenzado a llevar a cabo pruebas y experimentos exhaustivos. Se ha utilizado un conjunto de datos de 2000 descripciones (1000 reales y 1000 fraudulentas generadas por inteligencia artificial). De esta manera, se ha buscado la creación de un modelo equilibrado.

Este proyecto ha buscado enriquecer y demostrar el manejo y conocimientos adquiridos, a la par de enriquecer el presente proyecto final.

PROVINCIAS ESPAÑA

6.1. EL PORQUE DE ESTE MODELO

LA INTENCIÓN DE este proyecto se basa en la determinación de la diversidad y riqueza del mercado inmobiliario a nivel nacional. Tomando consciencia de la dificultad de escrapear en el tiempo disponible todas las provincias españolas, se ha tomado otra decisión: realizar 50 modelos usando muestras en vez de población.

6.2 REALIZACIÓN

En este proyecto, se recolectaron inicialmente alrededor de 90,000 datos de viviendas mediante scraping. Sin embargo, después de un riguroso proceso de limpieza de datos, se obtuvo un conjunto más depurado de aproximadamente 85,000 registros. Esta fase de limpieza de datos desempeñó un papel esencial para garantizar la calidad de la información utilizada en el posterior análisis.

Para abordar el objetivo principal del proyecto, se implementaron modelos de Random Forest con Gradient Boosting específicos para cada una de las 50 provincias en España, salvo Ceuta y Melilla debido que ha razón de la falta de datos se usa la población. Cada provincia se trató como un caso individual, y se recopilaron alrededor de 1,800 datos de viviendas por provincia para entrenar y desarrollar modelos personalizados. Estas 1,800 muestras se consideraron representativas de la población total de inmuebles en cada provincia.

El proceso de entrenamiento de estos modelos comenzó con una fase de optimización de hiperparámetros mediante GridSearch, lo que permitió afinar y mejorar el rendimiento de los modelos. Posteriormente, se aplicó la técnica de Gradient Boosting para incrementar aún más la capacidad predictiva de los modelos en cada provincia.

Es importante destacar que, en comparación con los modelos desarrollados para provincias más grandes como Málaga o Madrid, los modelos provinciales basados en un conjunto de datos de 1,800 muestras pueden tener un conjunto de variables predictoras más limitado debido a la disponibilidad de datos. Sin embargo, a pesar de esta limitación, el proyecto ha resultado en un conjunto valioso de herramientas que permiten realizar predicciones en el mercado inmobiliario a nivel provincial en España, contribuyendo así a una mejor comprensión y toma de decisiones en el sector inmobiliario.

LANDING PAGE

7.1. LA IDEA

LA CREACIÓN DE una Landing Page mediante la librería Streamlit surge como resultado del potencial y las habilidades dentro de nuestro equipo de trabajo. Uno de los miembros del equipo posee un amplio conocimiento en diseño, y todos nosotros hemos desarrollado una notable capacidad de aprendizaje durante el Bootcamp en el que participamos. Esta combinación de habilidades y recursos nos llevó a tomar la decisión de presentar todos los modelos desarrollados en nuestro proyecto a través de una plataforma de fácil acceso y visualmente atractiva: una Landing Page desarrollada utilizando la librería Streamlit.

Streamlit nos ha proporcionado la capacidad de mostrar de manera eficiente y atractiva para los usuarios todos los modelos que hemos creado durante el proyecto. Esto no solo facilita la interacción de los usuarios con nuestros resultados, sino que también demuestra nuestra habilidad para presentar de manera efectiva y profesional el trabajo realizado.

7.2. PORQUE STREAMLIT

La elección de utilizar Streamlit para crear nuestra Landing Page no solo ha sido una decisión técnica, sino que también responde a la importancia de hacer que nuestros modelos y resultados sean accesibles y comprensibles para un público que puede no estar familiarizado con la programación o la tecnología avanzada. Aquí hay algunas razones clave para destacar la importancia de esta elección:

Facilidad de Uso: No requiere que los usuarios tengan conocimientos de programación para interactuar con la plataforma. Esto significa que cualquier persona, independientemente de su nivel de habilidad técnica, puede navegar y comprender fácilmente la información y los modelos que presentamos.

Interfaz Intuitiva: La interfaz de usuario que Streamlit proporciona es altamente intuitiva. Los usuarios pueden navegar por la Landing Page de manera similar a como lo harían en un sitio web convencional, lo que facilita su interacción y comprensión de los resultados.

Visualización Clara: Streamlit permite presentar datos y modelos de manera visualmente atractiva. Gracias a esta capacidad, podemos representar gráficos, tablas y otros elementos de una manera que sea fácil de interpretar incluso para aquellos que no tienen experiencia técnica.

Transparencia y Comunicación Efectiva: Al crear una Landing Page accesible con Streamlit, se está priorizando la transparencia y la comunicación efectiva de los resultados. Esto es esencial cuando se trabaja en proyectos que involucran a partes interesadas que pueden no estar familiarizadas con la jerga técnica.

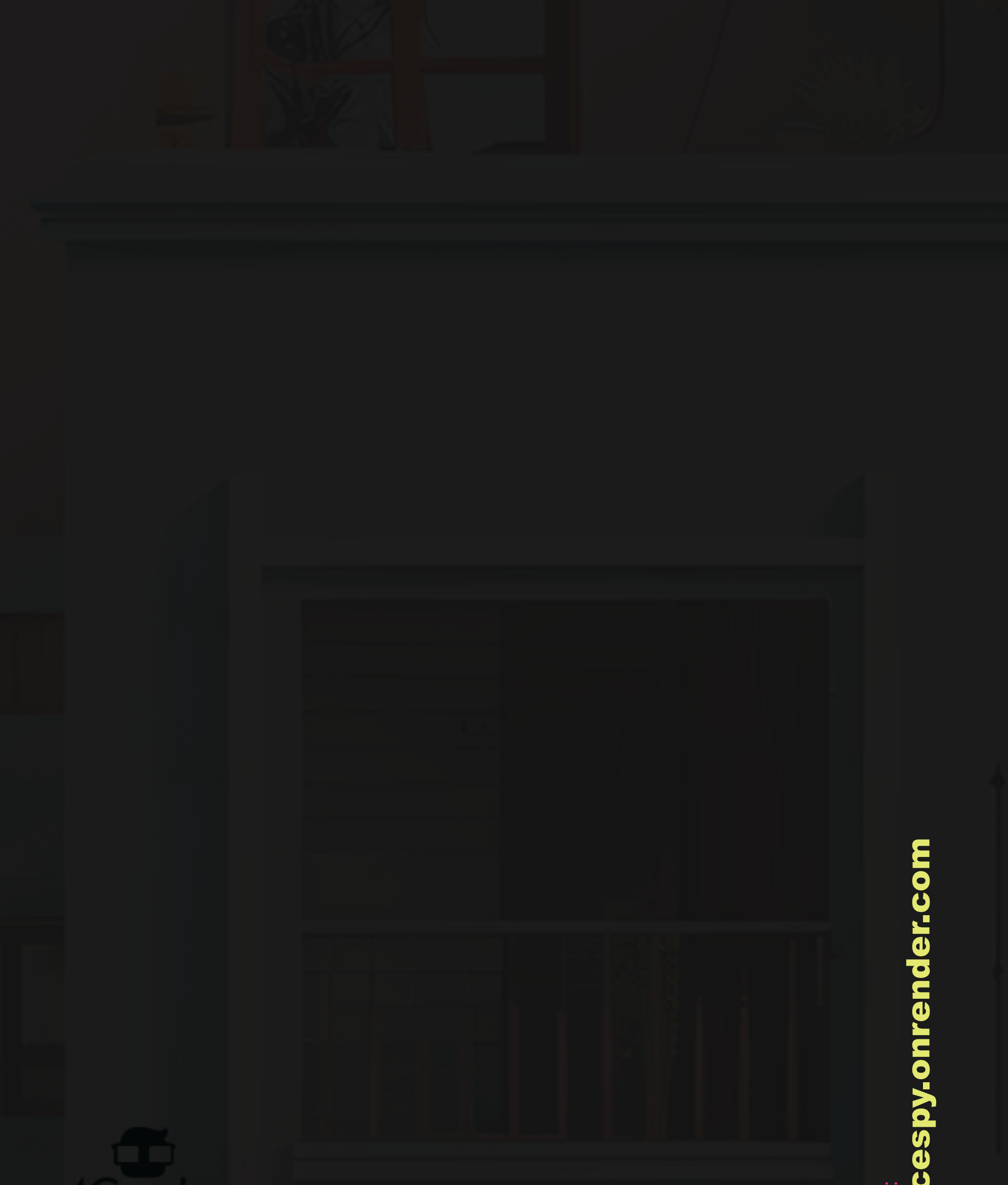
DIFICULTADES Y HABILIDADES

EN EL TRANSCURSO de su proyecto en el bootcamp, se han enfrentado diversas dificultades que se han transformado en oportunidades significativas de crecimiento personal y desarrollo de habilidades. Uno de los desafíos más notables ha sido el proceso de scraping de datos, que se ha convertido en un obstáculo debido a la limitación de tiempo en el bootcamp. A pesar de esto, se han embarcado en la tarea de investigar y aprender sobre nuevas librerías y técnicas para superar este obstáculo.

Además, la decisión de utilizar Streamlit para crear una Landing Page también ha presentado desafíos, principalmente porque era una herramienta relativamente desconocida en ese momento. Sin embargo, esta dificultad ha servido como una oportunidad para aprender sobre una nueva tecnología y comprender cómo utilizarla de manera efectiva para presentar sus modelos y resultados de manera accesible y visualmente atractiva.

Otra dificultad destacada ha sido la gestión de problemas técnicos, como los relacionados con el rendimiento y la visualización. A pesar de estos contratiempos, se han esforzado por encontrar soluciones y mejorar continuamente la calidad de su proyecto.

A través de estas dificultades, han adquirido valiosas lecciones. Se han aprendido a investigar y utilizar nuevas librerías, a gestionar el tiempo de manera eficiente, a lidiar con la frustración que a veces acompaña a los desafíos técnicos y a valorar aún más el trabajo en equipo y las habilidades humanas. Estas experiencias no solo les han permitido superar obstáculos técnicos, sino que también han fortalecido su capacidad para abordar futuros desafíos con confianza y determinación.



Diseño y maquetación: Jaime Dodero

Texto: Abel Montiel

Visita:
pricespy.onrender.com