Escuela Politécnica Superior

Ciencia de datos para determinar visitantes en un destino turístico inteligente

Máster Universitario en Ciencia de Datos

Trabajo Fin de Máster

Autor:
Javier Ignacio Belmar Tevar

Tutor/es:
Jose Norberto Mazón López

Jose Jacobo Zubcoff Vallejo

Septiembre 2021

Universitat d'Alacant
Universidad de Alicante

Índice

Tabla de ilustraciones

Capítulo 1: Introducción

En este capítulo se describe el contexto en el cual se desarrolla el presente trabajo de fin de máster (TFM). Cabe destacar que este TFM se ha desarrollado en un entorno real con el fin de solucionar un problema específico dentro del ámbito de los denominados destinos turísticos inteligentes.

Un destino turístico inteligente (DTI) se puede definir según la Sociedad de la Información a la Sociedad Estatal para la Gestión de la Innovación y las Tecnologías Turísticas (SEGITTUR) como "un espacio turístico innovador, accesible para todos, consolidado sobre una infraestructura tecnológica de vanguardia que garantiza el desarrollo sostenible del territorio, facilita la interacción e integración del visitante con el entorno e incrementa la calidad de su experiencia en el destino y la calidad de vida de los residentes" [6]. Por ello, se puede destacar que el aspecto clave de los DTI es la integración de las tecnologías de la información en el destino mediante la implantación de soluciones hardware y software innovadoras (desde sensores hasta herramientas de inteligencia de negocio) como se detalla en [4].

No obstante, la tecnología aplicada a un destino turístico no debe ser únicamente usada para mejorar la experiencia de las personas que visitan el destino (por ejemplo, para que puedan conectarse a Internet de manera gratuita mediante WiFi, para que puedan saber lo concurrida que está una playa o para que puedan conocer información de las atracciones turísticas por medio de una aplicación móvil), sino que también debe usarse para recopilar datos que permitan conseguir una mayor eficiencia en la gestión del destino [5].

En esta línea, se puede afirmar que un destino turístico se convierte en inteligente cuando hace un uso intensivo de la infraestructura tecnológica con el fin de (1) mejorar la experiencia del turista y (2) mejorar el proceso de toma de decisiones mediante la gestión de los datos.

Cabe destacar el modelo de DTI propuesto en [5] donde los autores reconocen las potencialidades de las tecnologías de la información para configurar destinos inteligentes, pero a partir de unas condiciones previas de carácter estratégico y relacional que determinan las acciones y el alcance de la estrategia. Su modelo se compone de tres niveles (ver Figura 1) que se detallan a continuación:

(1) Un nivel estratégico-relacional (enfocado hacia una gobernanza sustentada en la colaboración público-privada que garantice la sostenibilidad y un entorno abierto e innovador).
(2) Un nivel instrumental (centrado en la conectividad y la sensorización como precursora de un sistema de información que aporte inteligencia en el proceso de toma de decisiones).

(3) Un nivel aplicado (orientado al desarrollo de soluciones inteligentes para la mejora de la experiencia, marketing y gestión del destino).

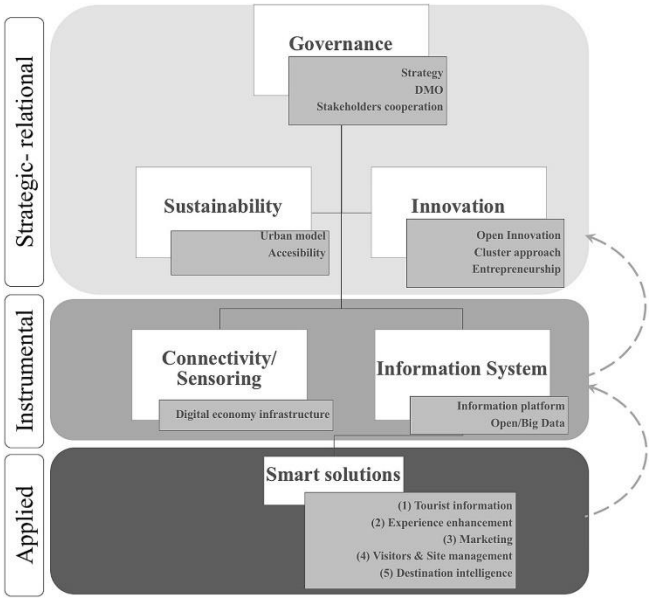


Figura 1. Modelo DTI procedente de [5]

Siguiendo este modelo de DTI, en este trabajo de fin de máster se sitúa en el nivel 3, es decir sobre la base de un problema específico de un destino turístico y una infraestructura hardware que nos proporciona ciertos datos, se propone la aplicación de técnicas de ciencias de datos para desarrollar una solución inteligente. Además, se debe destacar que este TFM realizará en un escenario real, en concreto en Alcoy, uno de los principales destinos turísticos de interior de la Comunitat Valenciana.

La localización geográfica de Alcoy, entre dos parques naturales, rango urbano, rico patrimonio histórico-cultural y tradición industrial le confieren interesantes atractivos que fueron objeto de una política y gestión turística a partir, fundamentalmente, de la creación de la Concejalía de Turismo y de la Oficina de Información Turística en 2000 y, sobre todo, del Plan de Dinamización Turística de Alcoi (2003), en cooperación con la Secretaría General de Turismo y la Agència Valenciana del Turisme. De hecho, las memorias anuales de la Oficina de Turismo de Alcoi[1] reflejan una gestión turística avanzada, patente en la certificación con la Q de calidad de la Tourist Info, la

[1] https://www.alcoyturismo.com/

participación del municipio en el Sistema Integral de Calidad Turística Española en Destinos (SICTED) o la reciente incorporación a la red de destinos turísticos inteligentes (red DTI) de la Comunitat Valenciana promovida por Invat·tur (Instituto Valenciano de Tecnologías Turísticas).

Este último hecho el que motiva que Alcoy sea el escenario del proyecto realizado en este trabajo de fin de máster. Alcoy trabaja desde hace tiempo en varios proyectos y estrategias centradas en la aplicación del concepto de ciudad inteligente (Alcoi Smart City[2]) donde destacan proyectos para convertir a Alcoi en un destino turístico inteligente. Uno de estos proyectos se denomina Alcoi Tourist Lab, en el seno del cuál se desarrolla este TFM. Alcoi Tourist Lab está subvencionado por la Agencia Valenciana de la Innovación y tiene como uno de sus principales objetivos, el poder desarrollar y/o probar soluciones y proyectos piloto funcionales hardware y software, que permitan obtener y analizar patrones y niveles de gasto de visitantes y ciudadanos de Alcoy que disfrutan de la oferta cultural, turística, tecnológica, gastronómica y de ocio de la ciudad. El objetivo que se pretende es analizar patrones de visitas y preferencias (siempre respetando la privacidad y la legislación de protección de datos). Desde Alcoy, se pretende analizar estos datos para mejorar la gestión del destino turístico, así como de la propia ciudad, incluyendo la mejora de los servicios ofrecidos tanto por la administración pública como para los sectores de comercio, turismo y hostelería.

Para ello, una necesidad crucial es conocer el número de visitantes que se encuentran en Alcoy en un momento determinado. Esta necesidad correspondería al nivel 1 del modelo DTI descrito anteriormente (Figura 1) ya que se corresponde con un objetivo estratégico. A partir de esta necesidad el proyecto Alcoi Tourist Lab define la infraestructura de sensorización necesaria (nivel 2 del modelo DTI de la Figura 1) con el fin de recolectar los datos necesarios para desarrollar una solución que pueda determinar los visitantes de un destino turístico y diferenciarlos de los residentes (nivel 3 del modelo DTI de la Figura 1).

Como se menciona anteriormente, este TFM se centra en desarrollar esta solución para entre discernir visitantes y residentes en el municipio de Alcoy sobre la base de la infraestructura desarrollada en el proyecto Alcoi Tourist Lab. Además, se realiza una validación y ajuste de la solución mediante el uso de encuestas que se llevan a cabo en la

[2] https://smartcity-alcoi.com/

Tourist Info (oficina de turismo) de Alcoy. De esta manera, el objetivo principal de este TFM es usar técnicas de ciencias de datos aprendidas durante el desarrollo del Máster en Ciencia de Datos de la Universidad de Alicante para poder hacer fiable la propuesta en este escenario real de la ciudad de Alcoy.

Capítulo 2: Escenario de partida

Al formar parte de un proyecto real, este TFM se inserta dentro de un escenario concreto que se describe en este capítulo, con el fin de describir el contexto en el cual se desarrolla y el punto de partida.

Dentro del anteriormente mencionado proyecto Alcoi Tourist Lab, se ha desarrollado un sistema inteligente de recolección y procesamiento de datos de dispositivos móviles que permite obtener la afluencia de personas que se encuentran en Alcoy en un momento determinado. Este sistema se basa en el uso de SSID (de sus siglas en inglés Service Set Identifier) de los enrutadores (routers), es decir el nombre de una red WiFi. Cada SSID es una cadena de caracteres que consta de un máximo 32 caracteres. El objetivo del sistema es identificar a aquellos dispositivos móviles (para o cual se detecta también la MAC de cada dispositivo) cuya WiFi almacenada como preferida para la conexión (en concreto el SSID de esa WiFi) esté ubicado en Alcoy y diferenciarlos de aquellos dispositivos cuya SSID preferida no sea de Alcoy.

Con este fin se desarrollan dos artefactos software diferenciados, cada uno haciendo uso de un hardware específico, a saber:

- Software que recopila tramas de datos de dispositivos móviles mediante sensores de afluencia de personas (dispositivos Smart Spots de HOPU que se describen más adelante). Estas tramas se procesan con el fin de determinar cuál es el SSID de la WiFi de conexión preferida de cada dispositivo.
- Software que recopila los SSID de las redes WiFi existentes en Alcoi. Este software se implanta y ejecuta en un dispositivo hardware desarrollado ad-hoc basado en una Raspberry Pi.

Se debe hacer notar que la recolección de SSID se realiza por medio de dispositivos hardware donde no se almacena ningún dato. Por otro lado, las tecnologías empleadas son innovadoras pero lo suficientemente maduras para evitar cualquier riesgo de seguridad. En concreto se usa el protocolo MQTT, base de datos MongoDB y servidor que cumple todos los requisitos de seguridad según normativa europea.

Este es el escenario de partida sobre el que se desarrolla el presente TFM. Este escenario se describe a continuación.

La figura 2 muestra un diagrama de la arquitectura propuesta, cuyos elementos principales se describen a continuación:

- Un recolector de SSIDs de los puntos de acceso WiFi existentes en una ubicación específica (por ejemplo, una ciudad). Este recolector tiene como objetivo crear

una base de datos precisa de los SSID disponibles en un destino turístico (parte superior izquierda de la figura, en naranja).

- Un recolector de SSID de puntos de acceso WiFi preferidos procedentes de tramas de datos de solicitud de sondeo de dispositivos móviles detectados por HOPU Smart Spots [7] (parte superior derecha de la figura, en verde).
- A partir de los datos recolectados en el escenario de partida (recolector de SSIDs de la ciudad y recolector de SSIDs preferidos asociados a MACs de dispositivos móviles), se requiere el desarrollo de un algoritmo para determinar la cantidad diaria de visitantes en diferentes lugares alrededor del destino (parte inferior de la figura, en azul). Esta parte corresponde con el objetivo principal de este TFM.

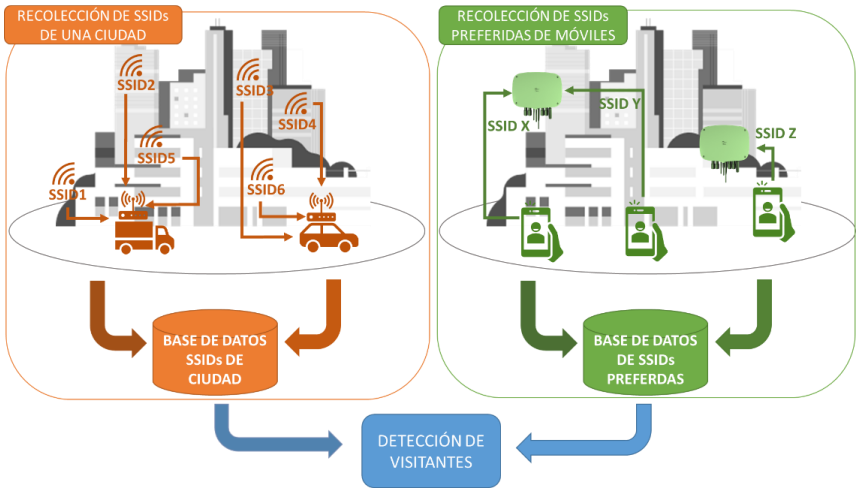


Figura 2. Escenario de partida para el desarrollo de la solución que determine el número de visitantes en diversos puntos de la ciudad de Alcoy de manera diaria.

2.1. Recolección de SSIDs

Los puntos de acceso WiFi suelen exponer sus SSID mediante el uso de tramas de balizas periódicas. Se trata de un anuncio periódico para informar a los dispositivos que están a la escucha de que ese SSID está disponible y tiene unas características determinadas. Los dispositivos clientes dependen de estas tramas de baliza para descubrir qué redes están disponibles (escaneo pasivo), y para asegurarse de que las redes a las que están asociadas están realmente presentes y disponibles.

Estos SSID pueden tener de 0 a 32 bytes de longitud y son, en general, una cadena de lenguaje natural de caracteres arbitrarios. Un análisis de los patrones expuestos en todo el mundo muestra que estas cadenas tienden a ser localmente identificables y únicas.

La recopilación de SSID es bastante inmediata y ha sido empleada masivamente por empresas como Google, Mozilla, etc. para mejorar sus sistemas de geolocalización de dispositivos móviles. Una tabla bastante completa de empresas que recogen esta información para el llamado WPS (WiFi Positioning System) y su disponibilidad se puede encontrar en [8]. La captura de SSIDs no está exenta de polémica, siendo destacable el escándalo en torno a la recogida subrepticia de datos WiFi mientras se capturaban imágenes de vídeo y datos cartográficos para el servicio Street View de Google [9]. A raíz de este escándalo, se han utilizado propuestas como la de añadir la etiqueta ”opt_out” a los SSID para optar por la captura de este servicio.

La legalidad de la recogida de esta información, especialmente en el espacio europeo, no está clara. En el escenario donde se desarrolla este TFM, el proyecto forma parte de un servicio público municipal, por lo que se han extremado las medidas de legalidad y privacidad, de acuerdo con los servicios de asesoría jurídica del municipio. Básicamente, el enfoque que se ha seguido es el de encriptar inmediatamente cada uno de los SSID capturados y almacenar únicamente las marcas de tiempo y la ciudad en la que se ha capturado (no se requiere una localización más específica).

Aunque la implementación más inmediata del sistema de captura de SSID sería el uso de una aplicación para teléfonos móviles como la de Wigle [10], se decidió desarrollar un recolector de SSID a medida para simplificar el desarrollo y mejorar el rendimiento de la captura de tramas SSID utilizando antenas de alta ganancia.

Para probar la propuesta, se utilizó una Raspberry Pi 4 Modelo B junto con una antena WiFi USB de doble banda modelo TP-Link Archer T3U Plus con una sensibilidad de unos -75 dBm para la banda de 2,4 Ghz y -70 dBm para la banda de 5 GHz. La elección del tipo de antena para el escáner de SSIDs ha sido un elemento clave porque se pretendía captar los SSIDs de los edificios adyacentes desde el nivel del suelo. En este sentido, se ha seleccionado una antena omnidireccional con una ganancia de unos 5dBi para ofrecer un compromiso razonable entre alcance y un patrón de recepción adecuado para captar los SSID de los routers WiFi instalados en los edificios adyacentes.

La Raspberry Pi recoge los datos de los SSID de las redes WiFi de una ciudad y se aplica el algoritmo hash SHA1 al SSID de cualquier punto de acceso Wi-Fi encontrado, así como la fecha de captura en formato *timestamp* y el nombre de la ciudad donde se produjo.

Los datos recopilados se almacenan en una base de datos MongoDB donde son accesibles con el fin de desarrollar el presente TFM.

2.2. Recolección de SSIDs preferidos de dispositivos móviles

La recopilación de los SSID preferidos se realiza mediante una versión modificada del dispositivo HOPU Smart Spot que se muestra en la figura 3. Se trata de un dispositivo IoT multisensor dedicado a aplicaciones de ciudades inteligentes. En nuestro caso, nos beneficiamos de su capacidad de monitorización de multitudes por WiFi que ha sido adecuadamente adaptada para conseguir las características deseadas que se describen a continuación.

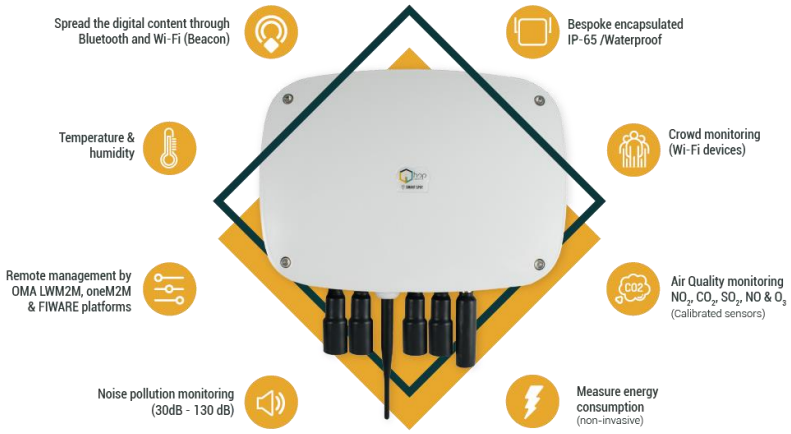


Figura 3. Dispositivo HOPU Smart Spot IoT.

El dispositivo HOPU Smart Spot escucha los paquetes de *probe request* enviados por los dispositivos móviles. Estos paquetes corresponden a los enviados por los teléfonos móviles cuando buscan un punto de acceso WiFi al que conectarse. Normalmente, la búsqueda de un punto de acceso WiFi puede ir acompañada del nombre del punto de acceso WiFi concreto que se busca. Una vez que se ha detectado un dispositivo móvil, dentro del propio HOPU Smart Spot, se aplica un algoritmo hash SHA1 a la MAC y al SSID (de la lista de redes preferidas o PNL) detectados, lo que imposibilita volver a

obtener la MAC y el SSID, anonimizando así la información del usuario en el propio dispositivo después de haberla recibido.

Las MAC y los SSID anonimizados mediante el hash SHA1 se almacenan en la memoria volátil y se cuentan en intervalos de 1, 5 y 10 minutos internamente en el dispositivo. Las MAC (y el SSID correspondiente de la PNL) se descartan/borran de la memoria cuando no son detectadas por el dispositivo y están fuera del intervalo de tiempo máximo (10 minutos).

Cada minuto, el dispositivo HOPU Smart Spot cuenta los dispositivos detectados en cada uno de los intervalos (1, 5 y 10 minutos) y envía esta información a través de los protocolos de integración disponibles en el dispositivo (que puede tratarse de los siguientes: LwM2M/MQTT/LoRa/SENTILO/FIWARE).

Todo este proceso de hashing de MACs y SSIDs dentro del propio dispositivo y sin almacenarlos en la memoria no volátil se realiza para proteger los datos de los ciudadanos de acuerdo con la actual legislación de protección de datos.

Cabe destacar que la recogida de SSID se realiza mediante dispositivos hardware en los que no se almacenan datos. Por otro lado, las tecnologías utilizadas son innovadoras pero lo suficientemente maduras como para evitar cualquier riesgo de seguridad. En concreto, se utiliza FIWARE y, en concreto el protocolo MQTT para el *broker* de mensajes a través de la implementación Mosquitto, y bases de datos y servidores MongoDB que cumplen con todos los requisitos de seguridad según los estándares europeos.

Capítulo 3: Objetivos

En este capítulo se describen los objetivos a cumplir por este TFM, que emanan directamente del escenario real considerado de Alcoy como destino turístico inteligente

A partir de la infraestructura que suministra el proyecto Alcoi Tourist Lab, se recolectan una serie de datos procedentes de los sensores de los Smart Spot de HOPU tal y como se ha descrito anteriormente. A partir de estos datos, el objetivo principal de este TFM es comprobar si es posible desarrollar un sistema de detección de visitantes a partir de estos datos.

Además, nos centraremos en uno de los sensores (el que se encuentra colocado en la oficina de turismo de Alcoy). Esto nos permitirá contar con datos obtenidos por encuestas manuales realizadas a las personas que acuden a la oficina de turismo (residentes y visitantes). Estas encuestas recogen información cómo el lugar de residencia, la cantidad de personas que viajan juntas, o si tiene el Wifi del móvil activado o no. En concreto, se ha creado un formulario de Google Forms con el fin de obtener estos datos de usuarias y usuarios de la Tourist Info de Alcoy que nos permiten validar el sistema desarrollado.

Capítulo 4: Tecnologías y herramientas

En este capítulo se describen las tecnologías y herramientas que se han usado para el desarrollo de este TFM y que complementan al escenario de partida (infraestructura tecnológica del proyecto Alcoi Tourist Lab) que se ha descrito en el capítulo 2 de este documento.

Para llevar a cabo toda la gestión de datos necesaria en este TFM, primero se ha elaborado una ETL mediante pentaho data integration para el volcado de datos desde la base de datos de MongoDB a los ficheros Excel y csv.

Después, se ha usado Python, lo que nos ha permitido la elaboración de visualizaciones y el desarrollo de los algoritmos necesarios. En concreto, se ha hecho uso de Visual Studio Code[3] desarrollado por Microsoft para la edición del código. Además, este programa soporta una extensión de Jupyter notebook también desarrollado por Microsoft recientemente actualizada (v2021.8.2041215044).

A continuación, se muestran las librerías que se han considerado como herramientas empleadas:

```
#### Librerías generales
import pandas as pd # manejo de datasets
from pathlib import Path # relativización del path de los archivos
import numpy as np
import datetime as dt # para pandas gestion datetime
from statsmodels.tsa.seasonal import seasonal_decompose

#### Visualizaciones
import plotly.express as px # gráficas interactivas 1
import plotly.graph_objects as go # gráficas interactivas 2
import matplotlib.pyplot as plt
import plotly.figure_factory as ff

#### Clustering / Dendrogramas
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage # dendrogramas
```

Como se observa se ha divido en tres partes las librerías utilizadas. Estas librerías, en su mayoría han sidp de gran importancia para llevar a cabo el TFM. En las subsecciones siguientes se detallarán cada una de ellas.

[3] https://code.visualstudio.com/

4.1. Pandas

Para el manejo de los conjuntos se ha utilizado `pandas`[4], que es la librería por excelencia de Python para la gestión de datos en forma de tablas. Con pandas ha sido posible importar los ficheros Excel y csv en el tipo de objeto Dataframe y Series propios de la librería. Soporta una gran cantidad de formatos de archivo. A lo largo de la investigación se ha utilizado diversas funcionalidades que ofrece la librería. Algunas de ellas son: Filtrado de datos, renombrado de columnas, agregación de valores con su correspondiente operación de grupo. Además, también ofrece la posibilidad de realizar operaciones estadísticas básicas como la media, un simple conteo, etc. También se ha utilizado la funcionalidad de `value_counts` para obtener rápidamente la distribución de valores en un campo.

4.2. Pathlib

Se ha utilizado `pathlib`[5] para trabajar con las rutas de los archivos y relativizar la búsqueda de estos a un nivel por encima de la carpeta donde se ejecuta este cuaderno.

Este módulo ofrece clases que representan rutas del sistema de archivos con la semántica apropiada para diferentes sistemas operativos (Windows, macOS, etc). Las clases de las rutas se dividen entre rutas puras, que proporcionan operaciones sin flujo de entrada y salida (E/S), y rutas concretas que heredan el comportamiento de las rutas puras pero también proporcionan métodos que realizan operaciones de E/S.

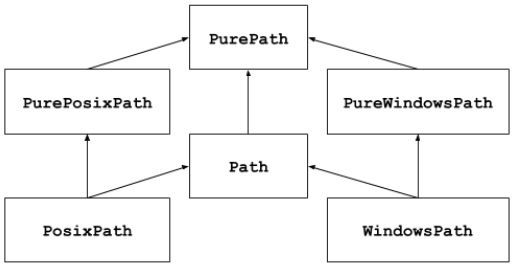


Ilustración 4. Jerarquía de las clases rutas que maneja pathlib

[4] https://pandas.pydata.org/
[5] https://docs.python.org/3/library/pathlib.html

4.3. Numpy

Se ha utilizado `numpy`[6] para hacer sobre todo transformaciones de datos como por ejemplo el uso de `np.nan` o `np.inf`.

Numpy es la librería fundamental de Python para realizar operaciones científicas. A parte de lo utilizado en el TFM, también nos permite la estructuración de datos en arrays y matrices, ofreciendo funcionalidades como las de operar, cambiar de dimensión, ordenación, y más operaciones para este tipo de objetos. Es posible usar funciones que crean directamente los famosos objetos del tipo `ndarray` como por ejemplo `np.arange` que ha sido utilizado en el TFM.

4.4. statsmodels.tsa.seasonal

Para la extracción de las estacionalidades se ha usado `seasonal_decompose`[7], función contenida en el módulo de statsmodels.tsa.seasonal. Este módulo proporciona clases y funciones para la estimación de muchos modelos estadísticos diferentes, así como para la realización de pruebas estadísticas y la exploración de datos estadísticos. De entre funciones y clase que aporta el módulo, se ha utilizado `seasonal_decompose` que a parte del uso mencionado, también se ha utilizado para descomponer las series temporales de manera aditiva analizar su tendencia, residuos y estacionalidad.

4.5. Matplotlib

Se ha utilizado maplotlib para elaborar dos tipos de gráficas. El primer tipo es una gráfica en la que se ha coloreado el fondo para señalar las fechas en las que la potencia

[6] https://numpy.org/doc/stable/user/whatisnumpy.html
[7] https://www.statsmodels.org/stable/generated/statsmodels.tsa.seasonal.seasonal_decompose.html

del sensor era distinta a las demás. El otro tipo de gráfica es una gráfica simple pero que ha servido para representar la calidad de las correlaciones presentadas.

4.6. plotly.graph_objects y plotly.express

Se ha aprovechado la interactividad de las gráficas que ofrece estos paquetes para obtener representaciones de mayor calidad. Por ejemplo, se puede hacer zoom a zonas en las que los valores parecen similares. Plotly es la desarrolladora de estos paquetes de gráficas que son ampliamente utilizados en el mundo de la ciencia de datos.

4.7. Sklearn.cluster

Por último para la detección de bajas y altas frecuencias se ha utilizado `Kmeans` de `sklearn.cluster`. Se ha hecho uso de linkage de scipy.cluster.hierarchy para apoyar la división de grupos de frecuencias. Skalearn.cluster ofrece variados algoritmos de no supervisados para clustering. A parte de ofrecer otros métodos para clustering como DBSCAN, también ofrece conjuntos de datos para practicar con ellos.

Capítulo 5: Trabajo relacionado

En este capítulo se describe brevemente trabajos anteriores relacionados con el desarrollo de este TFM. En concreto, se pone el foco en propuestas de detección de e afluencia de personas.

En los últimos años el estudio de la afluencia de personas mediante sensores ha ido creciendo indudablemente. Este tipo de mediciones se solía hacer mediante visión por computador, pero la llegada de los sensores capaces de reconocer dispositivos con señal WiFi revolucionó la investigación de este campo, pues estos pueden ser capaces de cubrir una mayor área y además sin necesidad de tener contacto visual. Esto fue debido a que los dispositivos que más llevaba consigo la gente (los móviles) tenían conexión WiFi, y por ello surgieron investigaciones que aprovechaban dicha conexión (como las de [1], [2] y [3]) para determinar la afluencia de personas en un determinado lugar.

En el estudio [1] se analiza la afluencia de personas en aeropuertos mediante el uso de dos monitores. Estos monitores están situados a ambos extremos del control de seguridad, de esta manera en esta investigación el "ground-truth" serían los datos del propio control de seguridad y desde los monitores se obtendrían los valores aproximados. El presente TFM presenta cierta similitud en la metodología empleada en [1] ya que nuestro "ground-truth" serían los datos recogidos por los cuestionarios de la Oficina de Turismo (Tourist Info) de Alcoy.

Al igual que se realiza en este TFM, en [2] se hace uso de la lista de SSIDs de los dispositivos detectados. Otras investigaciones tal y como se apunta en [3] han hecho uso de este tipo de registros para medir la duración que tienen las colas de espera, para estimación de trayectorias, relaciones sociales y estimaciones de densidad de población en una zona concreta. Este recurso de las SSIDs será empleado en esta propuesta de TFM para poder detectar los visitantes en un destino turístico.

En el ámbito del turismo, de hecho, son varias las propuestas que se han desarrollado con el fin de poder detectar a turistas con especial énfasis en tratar de determinar cómo se mueven por el destino turístico. A continuación, se describen brevemente algunas de ellas (se han seleccionado en dependencia del tipo de dato que utilizan y se ha realizado una breve comparación con la propuesta desarrollada en este TFM).

Por ejemplo, en [11] se utilizan los denominados datos móviles pasivos (es decir, datos de eventos registrados por los operadores de redes móviles en el curso del uso de los teléfonos móviles por parte de un consumidor conectado a las redes públicas de voz y datos) con el fin de realizar un seguimiento de turistas en un destino turístico. La ventaja de usar estos datos es que pueden registrarse en la red sin ninguna actividad por parte del

usuario: se generan automáticamente en cuanto se comunican un dispositivo móvil, la torre de comunicaciones y el sistema del operador de red de telefonía. Sin embargo, su principal inconveniente es que estos datos son propiedad de las empresas de telefonía por lo que si se quiere contar con esta información es necesario adquirirla.

Otras propuestas, como la desarrollada en [12] hacen uso de otras tecnologías como NFC (Near Field Communication), que es una tecnología inalámbrica para la transferencia de datos sin contacto físico. En concreto en este trabajo se estudia cómo aplicar NFC en el ámbito del turismo haciendo hincapié en las oportunidades pero analizando también las amenazas y problemas de su uso. En concreto, se puede destacar que NFC es de corto alcance, lo que limita su uso a situaciones concretas donde la persona que lo use realiza una actividad (un pago o el uso del transporte público).

Existen otros trabajos, como el descrito en [13] que hace uso de otro tipo de datos diferente, como son los datos de las transacciones bancarias. En este caso, el objetivo es complementar las estadísticas oficiales basadas en encuestas para suplir esta falta de información fiable. Estas estadísticas se complementan por medio del uso de los datos de las transacciones con tarjeta (tanto pagos como retiradas de efectivo en cajeros automáticos). La idea subyacente detectar patrones en las transacciones bancarias con el fin de determinar si la persona que las ha realizado es visitante o residente. Cabe destacar que este trabajo presenta aplicaciones prácticas mediante el uso de datos de BBVA. No obstante, el problema sigue siendo similar al uso de datos procedentes de empresas de telefonía: los datos de transacciones bancarias pertenecen a empresas y, para poder usarlos, es preciso adquirirlos, lo que tiene un coste asociado muchas veces no asumible por un destino turístico.

Un trabajo que usa datos procedentes de la señal WiFi es [14] donde se presenta un método para rastrear a las personas en eventos masivos de ciudades turísticas sin necesidad del uso de una aplicación móvil o de códigos QR o similares mecanismos que necesiten de la participación activa de las personas. Esta propuesta se basa en el escaneo en varios lugares los paquetes enviados por la interfaz Wi-Fi de los smartphones de los visitantes, y correlacionando los datos capturados en estos diferentes lugares. El método propuesto se implementa mediante una Raspberry Pi. Esta propuesta tiene semejanza con la que se desarrolla en este TFM ya que la base tecnológica es similar (uso de señales WiFi y de Raspberry Pi para conseguir los datos), sin embargo, el objetivo que se persigue es diferente, ya que lo que se pretende con este TFM es poder obtener el número de

visitantes de un destino turístico (no se pretende realizar un rastreo de los mismos). Además, este trabajo también conlleva implicaciones para la privacidad de las personas.

Capítulo 6: Detectando visitantes en un destino turístico mediante el uso de SSIDs

En este capítulo se detalla la propuesta desarrollada para poder detectar el número de visitantes que se encuentra en un destino turístico mediante el uso de la detección del listado de las SSIDs preferidas de cada dispositivo móvil.

6.1. Valor añadido de la propuesta

Como se ha comentado anteriormente, existe gran cantidad de trabajos relacionados que miden la afluencia de personas en un lugar concreto mediante la detección de dispositivos móviles mediante el uso de señales WiFi. En la propuesta de este TFM se va un paso más allá, cumpliendo el objetivo planteado por la Oficina de Turismo de Alcoy, y se propone el uso de las señales WiFi para discernir los visitantes que se encuentran en la ciudad de Alcoy.

Es innegable que, en España, el turismo forma parte de un porcentaje bastante significativo del PIB por lo que este tipo de sistemas que puedan conocer de primera mano la cantidad de visitantes en una zona de terminada tienen mucho interés. Estos datos, de hecho, se pueden usar para mejorar la gestión de un destino turístico y poder poner el foco de atención en aquellas zonas más visitadas.

La costa alicantina es un lugar al que acuden muchos turistas de forma periódica y, en este tipo de destinos de costa, normalmente existe una multitud de nacionalidades que visitan el destino. Es por ello, que es importante poder determinar la nacionalidad de cada visitante ya que la toma de decisiones al realizar ciertas estrategias, por ejemplo, de marketing, depende mucho de las nacionalidades (no es lo mismo una campaña de marketing para Reino Unido que para Francia). Sin embargo, este no es el caso de una ciudad de interior como Alcoy, que no requiere de estos datos tan concretos por nacionalidades, sino que sus necesidades de datos son otras, tal y como se muestra a continuación.

Alcoy es una ciudad de tamaño medio con una población de 58.994 habitantes (2019), la mayoría de los cuales viven en el área urbana. La arquitectura del casco urbano incluye principalmente edificios de 3 y 4 plantas. Alcoy es una ciudad de larga tradición industrial en un proceso de diversificación económica en el que la economía de los visitantes está jugando un papel emergente gracias a su rico patrimonio, tanto natural (dos parques naturales ubicados en el término municipal) como cultural (un sitio que forma parte de la lista de Patrimonio Mundial entre otros atractivos singulares de diferentes períodos históricos). A pesar de la escasa oferta de alojamiento (585 camas), según las encuestas realizadas por la oficina de turismo, el número de visitantes muestra un crecimiento

constante. De acuerdo con esta tendencia, desde la administración local de Alcoy, muy comprometida con las iniciativas de destino inteligente, está tratando de desarrollar nuevos sistemas de afluencia de personas, un objetivo reforzado por la necesidad de garantizar el distanciamiento social para asegurar una experiencia segura del visitante en el contexto de Covid-19. En concreto, con el fin de seguir tomando decisiones informadas en el ámbito del turismo, Alcoy requiere distinguir diariamente a los visitantes de los residentes en varios puntos de la ciudad, evitando las deficiencias de las gravosas encuestas tradicionales u otros enfoques intrusivos, como una aplicación móvil específica de destino turístico, al tiempo que se preserva la privacidad.

Figura 5. Vista de la ciudad de Alcoy (fuente: Jordi Miró).

Por ello, la propuesta realizada en este TFM aporta el valor que necesita la ciudad de Alcoy ya que, mediante el uso de la infraestructura disponible, se desarrolla y prueba un mecanismo para detectar la afluencia de personas en una determinada zona, poniendo énfasis en determinar cuántas de estas personas son visitantes (es decir, no son personas que residen en la ciudad de Alcoy). En concreto, la zona objeto de estudio es la oficina de turismo de Alcoy.

6.2. Recolección de los datos

Los datos se encuentran en una base de datos MongoDB recolectados según la infraestructura disponible en el proyecto Alcoi Tourist Lab (descrito en capítulos anteriores de este documento). Con el fin de acceder a los datos, proceder a su limpieza e integrarlos convenientemente, se ha desarrollado un proceso ETL (de sus siglas en inglés Extraction/Transformation/Load) mediante el uso de la herramienta Pentaho Data Integration.

La base de datos MongoDb original contiene tres colecciones, denominadas *ssidCollect*, *crowdLevel*, *places* y *dataSSIDCollector*. La finalidad de estas colecciones se describe a continuación:

- *ssidcollect*: contiene los datos procedentes de cada sensor de los dispositivos HOPU Smart Spot. En concreto, esta colección contiene los siguientes atributos:
    - *sensor* (string): almacena el id del sensor del que ha obtenido los datos.
    - *fecha* (timestamp): la fecha y hora exactas en la que se ha capturado la información.
    - *mac* (SHA1): información obtenida por el sensor, concretamente la MAC del dispositivo, o más bien la función de resumen de esa MAC tras aplicar el algoritmo SHA1.
    - *ssid* (SHA1): información obtenida por el sensor, concretamente el SSID de la PNL, o más bien la función de resumen de ese SSID tras aplicar el algoritmo SHA1.
- *places*: almacena información sobre el lugar donde están instalados los sensores. En concreto, esta colección contiene los siguientes atributos:
    - *sensor* (string): almacena el id del sensor que ha obtenido la información.
    - *nombre* (string): nombre del lugar donde se encuentra el sensor.
    - *desc* (string): breve descripción del lugar donde se encuentra el sensor.
    - *coords* (*coordinates*): coordenadas de la ubicación del sensor.
    - *ciudad* (string): nombre de la ciudad donde se encuentra el sensor. En este caso será siempre Alcoy.
- *dataSSIDCollector*: contiene los datos SSID recogidos por la Raspberry Pi. Los atributos son los siguientes:
    - *fecha* (timestamp): fecha exacta en la que se han capturado los datos.

- *ssid* (SHA1): información obtenida por la Raspberry Pi, concretamente el SSID, o más bien la función de resumen de ese SSID tras aplicar el algoritmo SHA1.
- *ciudad* (string): nombre de la ciudad donde se han obtenido los datos. En este caso será siempre Alcoy.

Con el fin de recolectar y obtener los datos necesarios para poder desarrollar y testear el algoritmo de detección de visitantes, se han desarrollado procesos ETL en la herramienta Pentaho Data Integration como el que se detalla en la Figura 5 donde nos conectamos a la base de datos de MongoDB (a la colección *ssidcollect*) para obtener los datos necesarios de SSIDs y MACs procedentes del sensor que se encuentra en la oficina de turismo de Alcoy (en concreto, este sensor se denomina CW08).

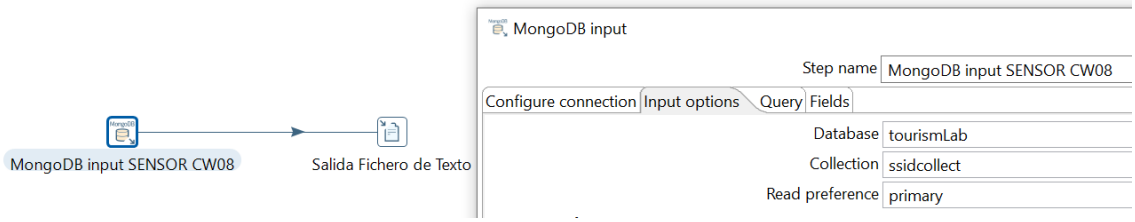


Figura 6. Proceso ETL que recolecta datos de los SSIDs

6.3. Descripción de los datos

Para este TFM se ha utilizado archivos en formato CSV y archivos de Microsoft Excel (XLSX). En concreto, los datos que provienen del sensor, una vez ejecutado el proceso ETL correspondiente se almacenan en un fichero CSV llamado *resultados_cw08.csv*. Este fichero contiene los datos que registran los sensores. En concreto, un registro de este archivo se compone de los campos siguientes:

- fecha: fecha y hora del registro.
- sensor: identificador del sensor al que pertenece el registro.
- ssid: ssid favorita del dispositivo detectado.

- mac: MAC (Media Access Control) del dispositivo.

- visitor: primera estimación de si la persona detectada es visitante o no. Contiene el valor Y si se asume que es una persona visitante y N si se asume que es una persona residente. Estos datos son lo que se pretende determinar correctamente con el desarrollo del presente TFM.

Este fichero contiene 83087 registros tomados desde el 21 de diciembre del 2020 hasta el 8 de julio del 2021 (7 meses aproximadamente).

Por otra parte, en un fichero de Microsoft Excel (XLSX) se encuentran los resultados de las encuestas realizadas por la oficina de turismo de Alcoy durante el mismo periodo. Cada registro se compone de la fecha y hora en la que completa el cuestionario y las respuestas a las preguntas del cuestionario que se detallan a continuación:

- Timestamp: Fecha y hora de realización del cuestionario.

- ¿Cúal es su lugar de residencia?: Las personas contestan el lugar del que proceden (Alcoi, Comunidad Valenciana, España o internacional). Esta pregunta permite determinar si la persona es residente en Alcoy o visitante.

- En el caso de que haya marcado anteriormente que su lugar de residencia NO es Alcoi. ¿Podría especificar de dónde es? (país, provincia, municipio...). Esta pregunta permite conocer de donde procede la persona visitante. Si bien esta información no se usa en este TFM, ya que sólo se necesita conocer si la persona es visitante o no.

- ¿Con cuántas personas viaja?: Número de personas con las que viaja el encuestado.

- ¿Con cuántas personas de las que viaja han entrado a la Tourist Info (oficina de turismo) ?: Personas que entran en la oficina junto con la persona encuestada. Esta información es relevante ya que la idea es que el sensor que se encuentra en la

oficina de turismo detecte a todas las personas que se encuentran en ella en un momento determinado.

- Con el fin de mejorar nuestros servicios de información turística. ¿Me podría indicar si tiene el Wifi de su teléfono móvil abierto para conectarse a redes WiFi disponibles? Se indicará Sí o No en el caso en el que encuestado tenga o no la WiFi activada. Estos datos se recopilan ya que una de las limitaciones de la infraestructura propuesta para la detección de visitantes es que depende de que los dispositivos móviles de las personas estén encendidos. Realizando esta pregunta en el cuestionario, se puede saber de manera manual, el porcentaje de personas que tienen activa la WiFi.

Este fichero contiene 1411 registros tomados desde el 27 de noviembre del 2020 hasta el 7 de julio del 2021 (7 meses aproximadamente). La franja de tiempo en la cual se toman estos datos es mayor que la franja anteriormente comentada en la que se toman los datos del sensor porque se estuvo varios días probando el procedimiento de realización de encuestas.

Por último, en un fichero CSV se tienen los registros de las SSIDs recogidas en Alcoy. En este fichero, hay 118972 registros que contienen los SSIDs de las redes WiFi que se encuentran en Alcoy. Este archivo se utilizará para hacer una primera discriminación entre visitantes y residentes. Cuenta con un único campo llamado SSID.

6.4. Análisis exploratorio

Se ha realizado un análisis exploratorio para comprender y comprobar el contenido y la calidad de los datos. A lo largo del análisis se ha ido haciendo algunos arreglos para obtener unos datos de calidad. Además, se ha observado la distribución de los valores para los campos de los distintos conjuntos y se ha estudiado formas para la agregación de los datos, y así, obtener otros conjuntos que puedan sernos de gran ayuda para las soluciones que se presentarán más adelante.

Se ha iniciado el análisis con la carga de los conjuntos del cuestionario, sensor y SSIDs de Alcoy. Para el cuestionario, se hizo un arreglo para el número de personas en la oficina de turismo. Este arreglo consiste sumar 1 a dicho campo (`num_personas_oficina`),

pues en el origen no se cuenta a la persona que realiza el cuestionario. En nuestro caso se contará, ya que será el campo que se utilice para contabilizar las personas que entran a la oficina de turismo.

Para futuras agregaciones (diarias) se ha creado un nuevo campo `date` que no estaba presente en los orígenes de datos. Esta operación se realiza tanto para los datos procedentes del cuestionario realizado en la oficina de turismo de Alcoi, como para los datos procedentes del sensor.

Realizados estos dos arreglos iniciales, los dos conjuntos principales (sensor y cuestionario) tienen el siguiente aspecto:

1. Porción del conjunto del cuestionario:

	timestamp	residencia	origen	num_personas	num_personas_oficina	wifi_abierta_tlf	date
0	2020-11-27 12:22:02.791	Alcoi	NaN	1	2	Sí	2020-11-27
1	2020-11-27 17:17:22.383	Alcoi	NaN	1	2	Sí	2020-11-27
2	2020-11-27 18:02:33.662	Comunidad Valenciana	Valencia	2	3	No	2020-11-27
3	2020-11-28 10:16:21.630	Comunidad Valenciana	Valencia	5	4	Sí	2020-11-28
4	2020-11-28 11:05:25.625	Alcoi	NaN	2	3	No	2020-11-28

2. Porción del conjunto del sensor:

	timestamp	sensor	ssid	mac	visitor	date
0	2020-12-21 12:30:26.336	CW08	cada330f25b98b5e0d44da5fc3decb161d337b4a	b627731c6c589cdabd7b02e979f7019c1d1efcf4	Y	2020-12-21
1	2020-12-21 12:30:26.336	CW08	cada330f25b98b5e0d44da5fc3decb161d337b4a	b627731c6c589cdabd7b02e979f7019c1d1efcf4	Y	2020-12-21
2	2020-12-21 12:30:27.443	CW08	cada330f25b98b5e0d44da5fc3decb161d337b4a	b627731c6c589cdabd7b02e979f7019c1d1efcf4	Y	2020-12-21
3	2020-12-21 12:30:27.443	CW08	cada330f25b98b5e0d44da5fc3decb161d337b4a	b627731c6c589cdabd7b02e979f7019c1d1efcf4	Y	2020-12-21
4	2020-12-21 12:30:31.944	CW08	cada330f25b98b5e0d44da5fc3decb161d337b4a	b627731c6c589cdabd7b02e979f7019c1d1efcf4	Y	2020-12-21

Por otro lado, tenemos el conjunto de las ssid’s recogidas de Alcoi que nos servirá para hacer una primera aproximación a los visitantes recogidos por el sensor.

- Porción del conjunto de ssid’s:

	ssid
2	00ccba5c6f7925e1fc70b2e0893852bf596c88c2
3	0143a4fa9f588990add630753c676a2c5961232d
4	015075bd5ebf650fa28d9016af6a4a50e25cb153
5	01593e9505926488c67e8cc24b8fc3de53f7dace
6	0185884ec4753ec27be08b3ee0bc7086b4db5ee4

Figura 7. Conjunto de ssid's de Alcoi

Más adelante se hablará de como este conjunto es crucial para determinar las personas que son visitantes de entre todas aquellas personas detectadas por el sensor.

La siguiente parte del análisis es la gestión de los valores nulos. Para empezar, se hace un barrido por los campos de los conjuntos y se encuentra que en el **cuestionario** el campo de origen tiene 520 nulos, en este caso no hay problema, pues que sea nulo depende directamente de la pregunta anterior del cuestionario. Concretamente, si en la anterior pregunta (segunda pregunta): "¿Cuál es su lugar de residencia?" se contesta Alcoi, la respuesta a la siguiente pregunta (tercera: “En el caso de que haya marcado anteriormente que su lugar de residencia NO es Alcoi. ¿Podría especificar de dónde es? (país, provincia, municipio...)”) figurará vacía, ya que se deberá especificar el origen en ese caso. Por tanto, lo que se va a realizar es un relleno de los campos nulos con "Alcoi".

Por otro lado, en el conjunto inicial del **sensor**, se observa que gran parte de los nulos cae sobre el campo `visitor`, de lo cual no hay de que preocuparse porque es nuestra tarea principal la estimación de este. También se ha encontrado algunos registros (3 en total) en los que en los campos de `ssid` y `mac` contienen algún nulo. Se procederá a eliminar dichos registros, la repercusión que genera esta eliminación no se puede comparar con la que generaría mantener los nulos.

Siguiendo con el análisis, nos damos cuenta de que es necesario realizar un filtro horario tanto para el sensor como para el cuestionario. El filtrado se realiza en dos pasos que consisten en:

- **Paso 1:** Se adaptan ambos conjuntos para que coincidan en fecha inicio y fecha fin. En este caso las fechas son 21 de diciembre del 2020 y 7 de julio del 2021

- **Paso 2:** Se realiza un filtrado de ambos conjuntos para que coincidan en horario. Esto se realiza porque el sensor siempre está activo y la oficina de turismo no siempre está abierta, esta tiene sus **horarios**:
    - Lunes a viernes: 10-14h.
    - Sábados, domingos y festivos: 10-14h.
    - Los días 4 de enero y 5 de enero abre de 10-14 y 17-19h.
    - Festivos cerrados: 1 y 6 de enero, 1 de mayo y 25 de diciembre.

En todo momento se adapta el conjunto menos restrictivo (en lo que a fechas se refiere) al más restrictivo.

Hacer un análisis exploratorio también nos ayuda a conocer la distribución de los valores de los campos. Esto nos pueda ayudar a saber qué nos podremos encontrar más adelante e incluso nos pueda ayudar a detectar alguna anomalía.

En Python es bastante sencillo obtener la distribución de los valores de un campo, se realiza de siguiente forma: `df[campo].value_counts().sort_values()`. La última función sirve para ordenar los valores según la frecuencia en la que aparecen. A continuación, se comentarán algunas de las distribuciones de los campos más relevantes:

- En cuanto a los datos procedentes del cuestionario realizado en la oficina de turismo de Alcoy:
    - `origen`: Se observa una clara mayoría del número de registros cuyo origen Alcoi o Valencia (351 y 250 respectivamente), en tercer lugar, está Alicante con 65 registros.
    - `num_personas_oficina`: Nos damos cuenta de que las personas suelen entrar a la oficina en pareja o en solitario (522 y 454 respectivamente)
    - `wifi_abierta_tlf`: Con esta distribución sabremos cuantas personas de las que rellena el cuestionario tenían la wifi activa o no. Como se detallará más adelante, este campo formará parte del algoritmo discriminador. En este campo tiene 3 valores posibles, `Sí` (tiene wifi activa), `No` (no la tiene) y `No entiendo la pregunta.` La distribución de los valores es 675, 321 y 118 respectivamente

- En cuanto a los datos procedentes del sensor situado en la oficina de turismo de Alcoy:
    - `visitor_v2`: Forma parte del a **primera aproximación hacia el algoritmo**. Se detallará más adelante. La distribución es la siguiente: 18816 (`Y`) y 28995 (`N`)
    - `ssid`: campo que contiene la SSID asociada a cada MAC en un instante determinado.
    - `mac`: Este es el campo que nos permitirá contar las personas detectadas por el sensor.

Antes de pasar con el algoritmo, se mostrarán dos gráficas que forman parte de este análisis, y que han ayudado a la decisión de realizar un clustering como siguiente paso del algoritmo.

Con la primera se observan las SSIDs ordenas en el eje x (donde figuran sus id's) en función de sus frecuencias. Nótese que en las últimas SSIDs las frecuencias empiezan a subir de manera acentuada. **Probablemente** sea debido a que existen personas que estén constantemente en presencia del sensor, por lo que podrían trabajadores de la propia oficina, o si tuvieran MACs dinámicas podrían pertenecer al mismo dispositivo que se conecta a la misma SSID.

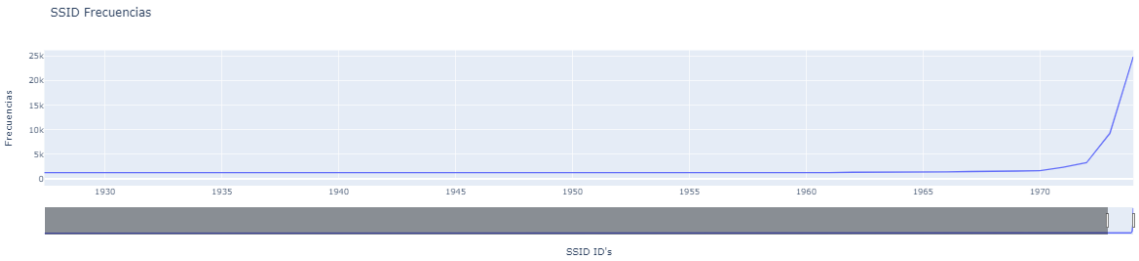


Figura 8. Gráfica de frecuencias de las ssid's

La siguiente gráfica se ha realizado con la misma idea que la anterior pero esta vez con las MAC. Como se podrá observar también existen unos pocos registros que muestran una alta frecuencia.

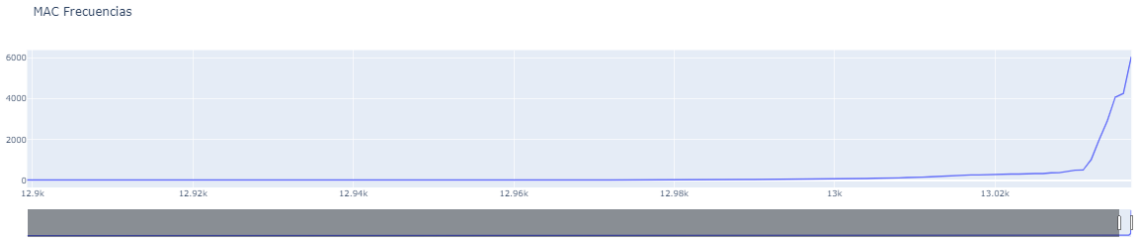


Figura 9. Gráfica de frecuencias de las mac's

6.4.1. Análisis diario

La manera de medir los sucesos de los conjuntos va a ser mediante el uso de series temporales. Estas series usarán como unidad tiempo el día. Para este proyecto estas series son una herramienta muy útil para saber lo que nos cuentan los datos.

Para empezar, se ha de adaptar los datos para que sean aptos para su incorporación a las series. Como ya se ha comentado antes, se va a medir principalmente las MACs del sensor y las personas que entran en la oficina de turismo para rellenar el cuestionario.

Aunque inicialmente (como objetivo exploratorio) se va a tener en cuenta tanto residentes como visitantes, el objetivo fundamental y final de las series es la medición de los visitantes.

Antes de seguir, se va a mostrar las consideraciones que se han tomado a la hora de confeccionar las series. Estas dos series podría considerar un boceto de las que vendrán continuación.

Esta ha sido la forma de realizar el conteo de las personas detectadas por el sensor. Se ha realizado un agregado diario del número de MACs **distintas** detectadas para día. No se va a considerar las MACs repetidas en un mismo día porque se pretende contar tan solo los visitantes nuevos que acudan a la oficina.

```
"""
Con mac's únicas
"""
df_sensor_d2_dist = df_sensor_d.groupby('date').mac.nunique()
df_sensor_d2_dist = df_sensor_d2_dist.to_frame()
df_sensor_d2_dist.reset_index(level=0, inplace=True)
df_sensor_d2_dist.head(5)
```

Se observa un pequeño fragmento de la serie inicial del sensor. En ella se observa las MACs agregadas por día.

	date	mac
0	2020-12-21	66
1	2020-12-22	251
2	2020-12-23	208

Figura 10. Serie del sensor inicial

Por otro lado, la implementación para crear la serie del cuestionario se muestra en la porción código de abajo. Como se verá más adelante se ha cogido tan solo personas oficina, porque el estudio se centrará tan solo en las personas que entraron a esta.

```
"""
Se cuenta el número de personas en la oficina.
Como se verá más adelante, se va a escoger la potencia mínima e intermedia del sensor por lo que
no detectará a personas de fuera de la oficina.
"""
df_cuest_d2 = df_cuest.copy()
df_cuest_d2 = df_cuest_d2[['date', 'num_personas_oficina']].groupby(['date'], as_index=False).su
m()
```

Una parte de la serie inicial del cuestionario es la siguiente:

	date	num_personas_oficina
0	2020-12-21	10
1	2020-12-22	15
2	2020-12-23	16

Figura 11. Serie del cuestionario inicial

Aunque no lo parezca, estas dos series presentadas son muy importes para la presente investigación, pues conforman las bases de la lógica aplicada a la contabilización de campos objeto de estudio.

La decisión de escoger para el cuestionario tan solo las personas que entran en la oficina, viene condicionada por la siguiente gráfica y estudios posteriores a esta.

Figura 12. Rangos potencia del sensor

Las áreas de distinto color representan diferentes niveles de potencia del sensor estudiado. Como se aprecia, estas áreas cambian a lo largo del tiempo.

Roja: Alta potencia. El sensor estaba configurado de tal manera que detectaba la más mínima señal WiFi de los dispositivos. Por ello, aunque hubiera dispositivos en un rango amplio de metros, el sensor los detectaba.

Azul: Baja potencia. El sensor estaba configurado de tal manera que sólo detectaba señales WiFi potentes de los dispositivos. Por ello, sólo detectaba dispositivos en un rango muy limitado de metros.

Verde: Potencia media. Es decir, se ajustó el sensor para poder detectar señales WiFi en un rango intermedio.

Con esta visualización ya se puede escoger descartar aquellos registros recogidos con la potencia del sensor alta. Es decir, para los siguientes estudios se tomará como fecha mínima de serie el 10 de mayo (inicio zona azul). Esto se debe a que los datos del cuestionario se han recolectado de las personas que estaban dentro de la oficina de turismo, por lo que si se quiere usar estos datos del cuestionario con el fin de probar el algoritmo de detección de visitantes procedentes de los datos del sensor, entonces se deben usar sólo aquella potencia que no detecte personas más allá de la oficina de turismo.

6.5. El algoritmo: Pasos comunes

A lo largo de la investigación se han desarrollado distintas soluciones (algoritmos) para poder aproximar las aportadas por el cuestionario. A pesar de obtener resultados distintos, todas las soluciones comparten una serie de pasos que veremos a continuación. Más adelante se explicará con detalle el algoritmo completo escogido.

6.5.1. Primer paso: discriminación con la lista de ssid's

Anteriormente se habló del conjunto de SSIDs de Alcoi, este conjunto es una pieza muy importante para la construcción del algoritmo de detección de visitantes del sensor. Recordemos que el conjunto contiene todas la SSIDs de Alcoi, y que como mediante el sensor también se recogen las SSIDs favoritas a las que se conectan, entonces es posible aproximar si un registro es visitante o no. Esto se realiza comprobando si la SSID que

aparece en un registro pertenece a lista de SSIDs recogidas correspondientes a Alcoi (conjunto de SSIDs de Alcoi). Para señalar visitantes y residentes se ha creado un campo `visitor_v2` con `Y` si es visitante y `N` si es residente. Esto se podría implementar fácilmente de la siguiente forma:

```
list_ssid = df_ssid_Alcoi['ssid'].tolist()

def f(row):
if row['ssid'] in list_ssid:
val = 'N'
else:
val = 'Y'
return val

df_sensor['visitor_v2'] = df_sensor.apply(f, axis=1)
```

6.5.2. Segundo paso: Clustering de frecuencias

Para las frecuencias de ambos campos (SSID y MAC), se ha realizado el clustering mediante KMeans. Según la regla del codo implementada a partir de la obtención del SSE (Sum of squared error) o la inercia. El método es sencillo: se prueban distintas combinaciones en el parámetro K (número de clústeres) y por cada una se calcula la in ercia. Esta inercia es la distancia o el error de cada punto a su centroide.

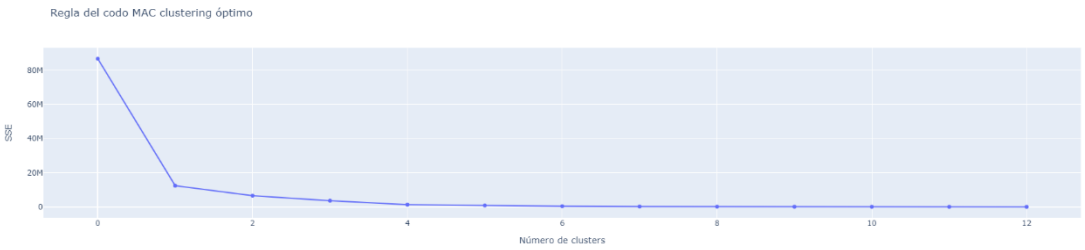


Figura 13. Método del codo (SSE)

Como se observa, por el SSE obtenido y la naturaleza del problema (potenciales trabajadores o no) se escogen dos clústeres. Como se verá más adelante se crearán dos clústeres: Uno con frecuencias altas y otro con bajas frecuencias. El método del codo se utiliza en estos casos para determinar el valor óptimo de K, este es el punto anterior a la disminución progresiva de la pendiente.

Este clustering se ha realizado simplemente para que no sea una persona la que escoja las frecuencias, a partir de las cuales se debe considerar desechar las MACs o las SSIDs que potencialmente sean empleados de la oficina o alrededores.

A continuación, se expone la implementación del clustering y una posterior implementación manual del aumento de la tolerancia. Esto es para que el clúster de altas frecuencias sea más permisivo:

```
km2 = KMeans(n_clusters=2, init='k-means++', n_init=10, algorithm='auto')
y_km2 = km2.fit_predict(df_aux2[['counts']])
df_y_km2 = pd.DataFrame(y_km2, columns = ['counts_clust'])

# clusters
df_y_km2['counts_clust'].value_counts().sort_values()
df_y_km2['index'] = np.arange(df_y_km2.shape[0])
df_y_km2.tail(10)
num = len(df_y_km2.query('counts_clust != 0')['index'].to_list())

################ Para relajar el cluster #################
tol = 0 # mayor tol, más macs eliminadas
##########################################################
num = num + tol
lista_ids_reps = df_y_km2.tail(num)['index'].to_list()
lista_macs_reps = df_aux2[df_aux2['mac_id'].isin(lista_ids_reps)].mac.to_list()
```

En el ejemplo de arriba, la lista `lista_macs_reps` contiene las MACs que se eliminarán del conjunto del sensor. Esta acción consigue rebajar lo suficiente (cantidad mínima) las cifras del sensor para que ahora se ajusten más a lo recogido por el cuestionario. Al estar el conjunto de frecuencias ordenado de menor a mayor, el resultado del clustering para las MACs y las SSIDs resulta sencillo de comprobar, pues las id's de las MACs y las SSIDs con frecuencia elevada está situadas el final del conjunto y pertenecen al clúster 1.

Las siguientes figuras muestra la asignación por el clustering realizado:

	counts_clust	index
13028	0	13028
13029	0	13029
13030	0	13030
13031	0	13031
13032	0	13032
13033	0	13033
13034	1	13034
13035	1	13035
13036	1	13036
13037	1	13037

Figura 15. Resultado del clustering para las mac's

	counts_clust	index
1965	0	1965
1966	0	1966
1967	0	1967
1968	0	1968
1969	0	1969
1970	0	1970
1971	0	1971
1972	0	1972
1973	0	1973
1974	1	1974

Figura 14. Resultado del clustering para las ssid's

Con estos datos se procede a elaborar una lista de exclusión de MACs y SSIDs Para eliminar los registros que sean trabajadores potenciales de la oficina.

6.6. El algoritmo: Pasos de la solución escogida.

En esta sección se va a desarrollar los dos últimos pasos de la solución escogida. Se seleccionó esta aproximación por la naturalidad de la misma, su justificación lógica y sus resultados.

6.6.1. Tercer paso: Obtención de un conjunto con conocimiento a priori.

A continuación, se va a detallar las pequeñas etapas que componen este paso. El objetivo es la obtención de un conjunto que incorpore conocimiento a priori que permita aproximar el número de personas que no han podido ser detectadas por el sensor porque no tenían la Wifi activa. Este conocimiento trata de las cifras de los visitantes mensuales recogidos por la oficina de turismo de Alcoy en 2019.

Tal y como se observa en la Tabla 1, en el año 2019, la oficina de turismo (Tourist-Info) de Alcoy atendió un total de 36.387 visitantes (de las cuales 31% eran locales, 57% del resto de España y 12% del extranjero) En este sentido, a nivel mensual, la afluencia de visitantes fue superior en los meses de abril, coincidiendo con la festividad más característica del lugar mayo, septiembre, y finalmente diciembre.

Por el contrario, durante la temporada estival la afluencia de visitantes disminuyó considerablemente ya que el destino no destaca por ofrecer unos productos turísticos orientados al perfil del visitante de verano.

TABLA 1. NÚMERO DE VISITANTES TOURIST-INFO. (2019)		
INTERNACIONAL	NACIONAL+LOCAL	TOTAL VISITANTES

	Nº	*% del total internacional*	*% del total de visitantes*	Nº	*% del total nacional + local*	*% del total de visitantes*	Nº	*% del total*
ENE	403	*9,23%*	*19%*	1686	*5,27%*	*81%*	2089	*6%*
FEB	184	*4,21%*	*11%*	1557	*4,86%*	*89%*	1741	*5%*
MAR	447	*10,23%*	*15%*	2448	*7,65%*	*85%*	2895	*8%*
ABR	325	*7,44%*	*6%*	5382	*16,81%*	*94%*	5707	*16%*
MAY	1071	*24,52%*	*20%*	4382	*13,69%*	*80%*	5453	*15%*
JUN	254	*5,82%*	*11%*	2031	*6,34%*	*89%*	2285	*6%*
JUL	255	*5,84%*	*15%*	1500	*4,68%*	*85%*	1755	*5%*
AGO	205	*4,69%*	*21%*	763	*2,38%*	*79%*	968	*3%*
SEP	470	*10,76%*	*9%*	4773	*14,91%*	*91%*	5243	*14%*
OCT	443	*10,14%*	*24%*	1401	*4,38%*	*76%*	1844	*5%*
NOV	137	*3,14%*	*10%*	1276	*3,99%*	*90%*	1413	*4%*
DIC	174	*3,98%*	*3%*	4820	*15,05%*	*97%*	4994	*14%*
TOTAL	**4368**	***100%***	***12%***	**32019**	***100%***	***88%***	**36387**	***100%***

Fuente: Memoria Touris-Info Alcoi 2019

Como ya se ha visto anteriormente, se va a escoger los meses de mayo, junio y julio.

- Primera etapa: se obtienen las fechas de dos años atrás (2019), así como el número de mes de las fechas de ambos años.

- Segunda etapa: **Agregación mensual de los visitantes recogidos por el cuestionario,** esto se puede realizar por la etapa anterior. Recordemos que inicialmente el conocimiento nos proporciona los visitantes mensuales del año 2019.

- Tercera etapa: **Corrección de las medias mensuales del conjunto de 2019.** Se ha de adaptar aproximadamente a los días en los que realmente se recogen visitantes para el conjunto de 2021. Recalcar que estas series empiezan el 10 de mayo y acaban el 7 de julio, por lo que habría que realizar una pequeña corrección en 2019, pues en esos años las cifras pertenecen esos 3 meses completos.

- Cuarta etapa: **Uso del factor covid.** Es lógico pensar en la influencia que tiene la incidencia de la pandemia de covid en las visitas. En esta etapa se trata de usar el factor mensual proporcionado por la oficina de turismo de Alcoy que pueda relacionar las visitas del 2019 mensuales con las 2021 mensuales. Este es 0.05, 0.1 y 0.1 en los meses de mayo, junio y julio.

- Quinta etapa: **Añadido de la componente seasonal.** Mediante la función de Python `seasonal_decompose`, se obtiene la componente estacionaria del cuestionario. Como se ha realizado mediante una descomposición aditiva, mediante algunos arreglos, el sensor se podrá beneficiar de ella. En la siguiente gráfica se muestra la descomposición aditiva de la serie del cuestionario. Se utilizará tan solo la componente seasonal. Se observa una clara estacionalidad semanal.

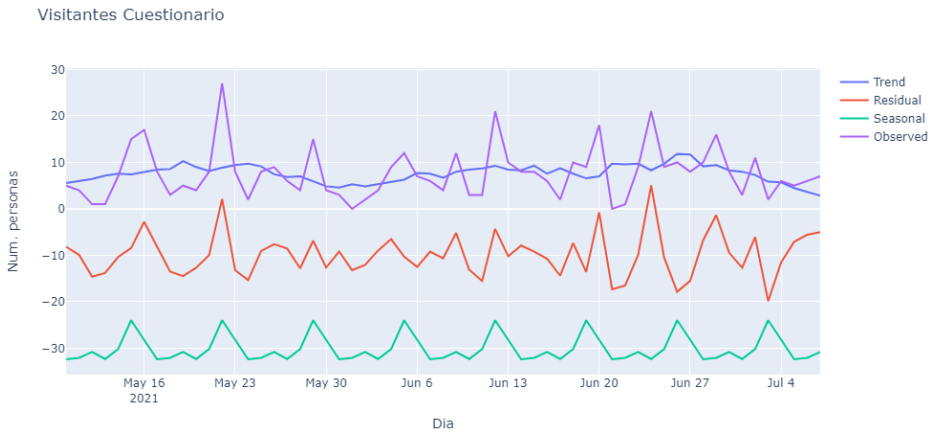


Ilustración 16. Descomposición de la serie del cuestionario

- Sexta etapa: **Suma de la estacionalidad y arreglo diario del 2019 por el factor covid.** El arreglo diario no es más que la división entre la media mensual corregida (por el f. covid) y el número de días considerados para ese mes, a eso se le suma la componente estacionaria. Llamaremos a este campo obtenido `visitantes_dia_media_season_corr`.

6.6.2. Cuarto paso: Aplicación a los datos del sensor el conocimiento a priori

Estas dos etapas aplican el conocimiento a priori a los registros del sensor.

- Séptima etapa: **Aplicación del ratio de las personas sin wifi.** En secciones anteriores se comentó que el sensor solo detectaba los dispositivos cuya conexión wifi estuviera activa. Se ha obtenido el ratio (a partir del cuestionario) aproximado

de personas que usan y no usan dicha conexión. Gracias a etapas anteriores, tenemos una estimación de los visitantes del cuestionario con el campo `visitantes_dia_media_season_corr`. Por lo que se procede obtener el valor aproximado de las personas que entraron sin wifi activa a la oficina, y así sumárselo al sensor. A esta corrección se le suma los visitantes registrados por el sensor. A continuación, se observa las proporciones obtenidas en el campo `wifi_abierta_tlf`:

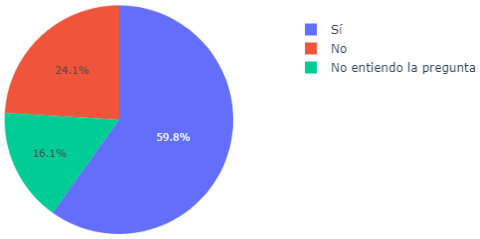


Figura 17. Proporción de personas con wifi en la oficina

- Octava etapa: **Corrección mediante factores externos.**

  Se puede ajustar la curva de los valores aproximados mediante su multiplicación por un factor de ajuste sin alterar la correlación entre dicha curva y el "ground-truth". Se puede conseguir dicha corrección de ajuste a partir de la media de los cocientes de la división entre los visitantes reales y los últimos valores devueltos por el algoritmo. Como se verá más adelante, al ser una media, la correlación entre los valores resultado (antes y después de la operación) y los valores reales no cambia.

Capítulo 7: Resultados

En este capítulo se describen los resultados obtenidos al aplicar el algoritmo desarrollado según las etapas explicadas en el capítulo anterior.

En las secciones anteriores se han detallado los pasos y cada una de sus etapas que se han seguido para obtener los resultados que se verán continuación. Se realizará una comparación visual de ajuste y una comparación entre las correlaciones de las curvas obtenidas antes y después de los pasos seguidos. Las siguientes gráficas muestran los resultados alcanzados en el rango de fechas en el que el sensor estaba configurado con la potencia baja.

En esta gráfica se observa como el sensor no recibe apenas información de los dispositivos (más adelante se **discutirá** esto):

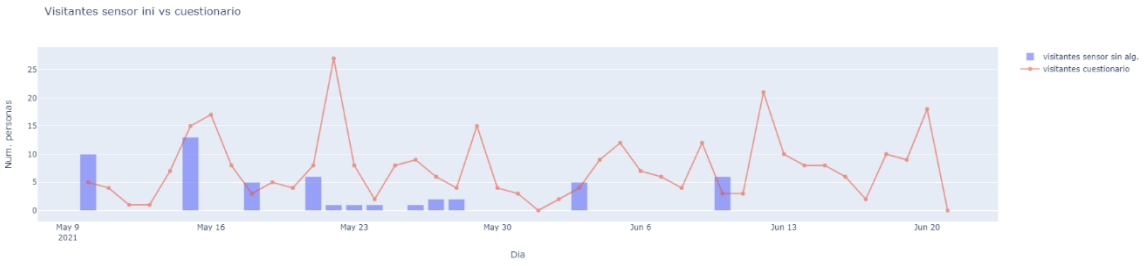


Figura 18. Resultados Antes de algoritmo

La implementación realizada del algoritmo inicial con el que se obtiene los visitantes del sensor de la figura de arriba es la siguiente:

```
visitantes_ini = df_sensor[['date', 'visitor_v2', 'mac']].query('visitor_v2 == "Y"') \
                 .groupby(['date']).mac.nunique()

visitantes_ini = visitantes_ini.to_frame()
visitantes_ini.reset_index(level=0, inplace=True)
visitantes_ini['date'] = pd.to_datetime(visitantes_ini['date'])
visitantes_ini = visitantes_ini[visitantes_ini['date'] >= '2021-5-10']
```

Las dos primeras líneas son las que discriminan entre visitantes y residentes a partir del campo `visitor_v2`, recordemos que este campo adquiría sus valores de la siguiente forma:

```
list_ssid = df_ssid_alcoy['ssid'].tolist()

def f(row):
    if row['ssid'] in list_ssid:
        val = 'N'
    else:
        val = 'Y'
    return val
df_sensor['visitor_v2'] = df_sensor.apply(f, axis=1)
```

En la siguiente visualización, vemos cómo se consigue lidiar con esa falta de sensibilidad del sensor en el **mismo** rango de fechas (algoritmo escogido).

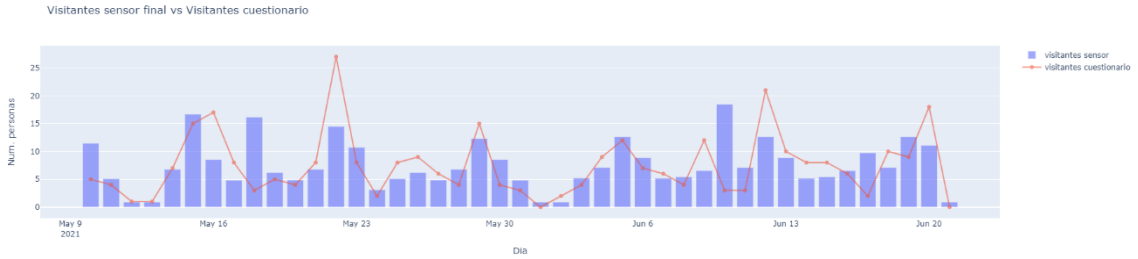


Figura 19. Resultados después de algoritmo

La implementación realizada del algoritmo final con el que se obtiene los visitantes del sensor de la figura de arriba es la siguiente:

1. Se crea un dataframe para preparar los cálculos que se verán más adelante:

```
df_comparacion = pd.DataFrame({'date':pd.date_range(start = dt.datetime(2021,5,10), \

                end = dt.datetime(2021,7,7))})

df_comparacion = pd.merge(df_comparacion, visitantes, on=['date'], how='left')

df_comparacion = pd.merge(df_comparacion, visitantes_c, on=['date'], how='left')

df_comparacion.rename(columns={'mac':'visitantes_s', 'num_personas_oficina':'visitantes_c'},inplace=True)

df_comparacion = pd.merge(df_comparacion, residentes, on=['date'], how='left')

df_comparacion = pd.merge(df_comparacion, residentes_c, on=['date'], how='left')

df_comparacion.rename(columns={'mac':'residentes_s', 'num_personas_oficina':'residentes_c'},inplace=True)

df_comparacion.fillna(0,inplace=True)
```

2. En el mismo dataframe se crean las fechas y los números de mes del 2019

```
# quitar 2 años de la fecha actual
df_comparacion['2019_dates'] = df_comparacion['date'] - pd.DateOffset(years=2)

# extraer numero de mes
df_comparacion['mes_num_2019'] = pd.to_datetime(df_comparacion['2019_dates']).dt.month

# num dias en un mes
```

```
df_comparacion['dias_mes_2019'] = df_comparacion['2019_dates'].apply(lambda t: pd.Period(t, freq
='S').days_in_month)
```

3. Se obtienen los números de los meses para 2021, por otro lado, se calcula la suma mensual de los visitantes del cuestionario.

```
df_comparacion['mes_num_2021'] = pd.to_datetime(df_comparacion['date']).dt.month
df_agregacion_2021_visitantes_mes = df_comparacion[['mes_num_2021', 'visitantes_c']].groupby(['m
es_num_2021'], as_index=False).sum()
```

4. Se hace el arreglo de los días para los meses incompletos, se instancia una serie con los números de los días usados tanto para el sensor como para cuestionario. Recordemos que en el mes de mayo se analizan los datos a partir del día 10, de junio se tiene el mes entero y de julio hasta el día 7. Al final se obtien un dataframe con los números correspondientes a los meses de 2019, los visitantes mensuales arreglados de 2019 y los días de los meses para 2021 (21, 30 y 7).

```
meses = pd.Series(['2019-05-01', '2019-06-01', '2019-07-01'])
visitantes_2019 = pd.Series([5453*2/3, 2283, 1755/4]) # Arreglo de mes incompleto
mes_dias_2021 = pd.Series([21, 30,7])
cuest_v_2019_mensual_frame = {'mes_num_2019': meses, 'visitantes_2019': visitantes_2019, 'mes_dias_2021': mes_
dias_2021}
df_cuest_v_2019_mensual = pd.DataFrame(cuest_v_2019_mensual_frame)

df_cuest_v_2019_mensual['mes_num_2019'] = pd.to_datetime(df_cuest_v_2019_mensual['mes_num_2019']).dt.month
```

	mes_num_2019	visitantes_2019	mes_dias_2021
0	5	3635.333333	21
1	6	2283.000000	30

Ilustración 20. Con numero de mes, número de días por mes y visitantes del 2019

5. Descomposición de la serie del cuestionario y guardado de la estacionalidad en una variable. Además se crea una dataframe a partir del anterior que albergará más columnas producto de operaciones futuras.

```
df_comparacion_aux_2 = df_comparacion.copy()
result_add_v_c_aux_2 = seasonal_decompose(df_comparacion_aux_2['visitantes_c'], \
                        model='additive', extrapolate_trend='freq', period=7)
seasonal = result_add_v_c_aux_2.seasonal
```

6. Se agrega al último dataframe creado una columna que contiene los visitantes del cuestionario mensuales. Además añade la columna resultado de aplicar el factor covid a las medias mensuales (días corregidos) de los visitantes del 2019.

```
df_visitantesC_mensuales = df_comparacion_aux_2[['visitantes_c', 'mes_num_2021']]. \
                           groupby(['mes_num_2021'], as_index=False).sum()

df_visitantesC_mensuales.rename(columns={'visitantes_c': 'visitantes_c_mensuales'}, inplace=True
)
df_comparacion_aux_2 = pd.merge(df_comparacion_aux_2, df_visitantesC_mensuales, on=['mes_num_202
1'], how='left')

df_comparacion_aux_2['vistantes_dia_media_2019_corrCovid'] = df_comparacion_aux_2['vi
sitantes_2019'] * f_covid
```

7. Se divide la media media mensual de visitantes entre el número días que tiene el mes correspondiente. Se reparte la cantidad entre los días del mes.

```
df_comparacion_aux_2['visitantes_dia_media_season_corr'] = \
(df_comparacion_aux_2['vistantes_dia_media_2019_corrCovid'] \
/ df_comparacion_aux_2['mes_dias_2021']) + df_comparacion_aux_2['seasonal']
```

8. Uso de la proporción de personas que entran a la oficina con el Wifi inactivo. Este ratio es multiplicado por la columna obtenida en el paso anterior. Después se le suma las personas detectadas por el sensor (Wifi activa).

```
ratio_wifi = 0.402
df_comparacion_aux_2['visitantes_dia_media_season_corr_wifi'] = \
(df_comparacion_aux_2['visitantes_dia_media_season_corr']*ratio_wifi) + \
 df_comparacion_aux_2['visitantes_s']
```

9. Se realiza la corrección de ajuste entre series que no afecta la correlación entre ambas.

```
coef_corr_v_s = (df_comparacion_aux_2['visitantes_c'] / \
df_comparacion_aux_2['visitantes_dia_media_season_corr_wifi']).mean()
df_comparacion_aux_2['visitantes_dia_media_season_corr_wifi_corr'] = \
df_comparacion_aux_2['visitantes_dia_media_season_corr_wifi']*coef_corr_v_S
```

Para comprobar los resultados de forma numérica, se ha utilizado la **correlación** de Pearson. La que se ha obtenido los datos la figura 13 entre lo detectado por el cuestionario y lo detectado por el sensor es de 0.013.

Mientras que la correlación obtenida de la figura 14. entre lo detectado por el cuestionario y lo detectado por el sensor es de **0.55**. Las gráficas que muestran estos resultados son:

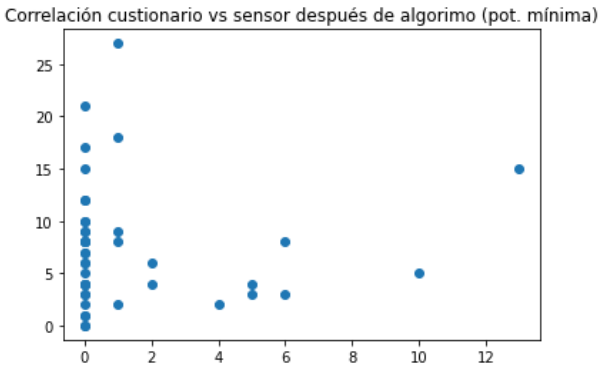


Figura 15. Plot de la correlación antes del algotimo. Pot. mínima

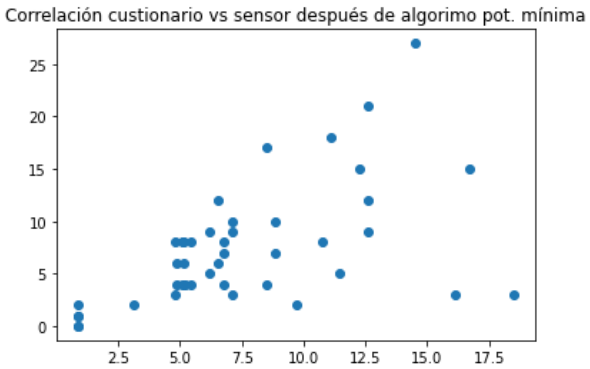


Figura 16. Plot de la correlación después del algotimo. Pot. mínima

Como se observa, la correlación es más acertada después del algoritmo. Ahora se mostrarán los resultados obtenidos si se añadiera la potencia intermedia (del 22 de junio al 7 de julio). Al añadir este rango de fechas en las visualizaciones se observa un gran cambio en esa parte en la gráfica análoga a la figura 13. De esta forma el sensor es capaz de detectar más información, pero existen unos inconvenientes que se **discutirán** más adelante.

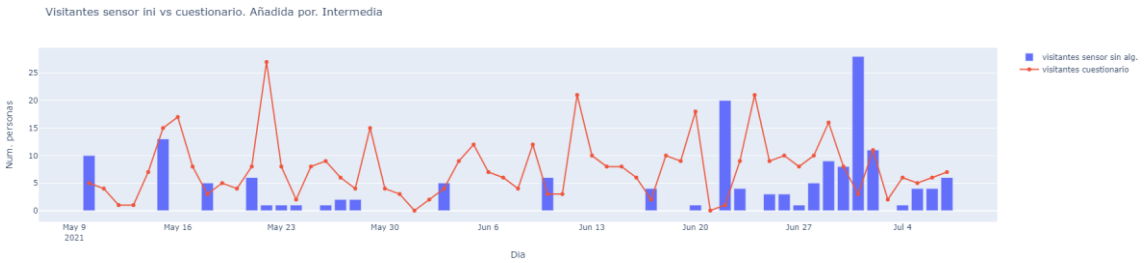


Figura 21. Resultados antes de algorimo. Pot. Intermedia añadida

En la gráfica mostrada se observa un crecimiento de los visitantes detectados por el sensor en el rango de fechas en el que la potencia es intermedia.

Si añadimos también ese rango de fechas los datos conseguidos por el algoritmo se obtendría la gráfica siguiente:

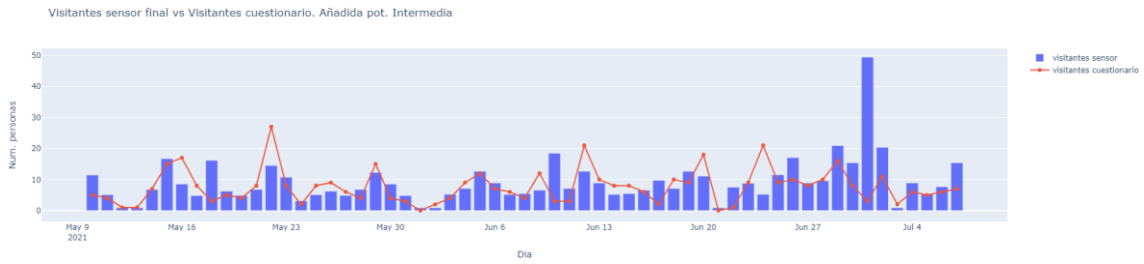


Figura 22. Resultados después de algoritmo. Pot. Intermedia

Los resultados de esta representación sugieren la idea de que existe una excedencia (de algunos valores) de visitantes detectados por el sensor tomando como referencia los recogidos por el cuestionario. Esto se **discutirá** también en la sección correspondiente.

En el primer rango de fechas (10 de mayo a 22 junio) parece existir días en los que no se detecte ningún dispositivo, pero en realidad sí que se están detectando. Inicialmente (antes de algoritmo) no existe una discriminación entre visitantes y residentes elaborada y esto también puede ocasionar las cifras tan bajas obtenidas en el primer rango. A continuación, se mostrará una gráfica con los dos rangos de fecha en la que también se contabilizan los residentes iniciales:

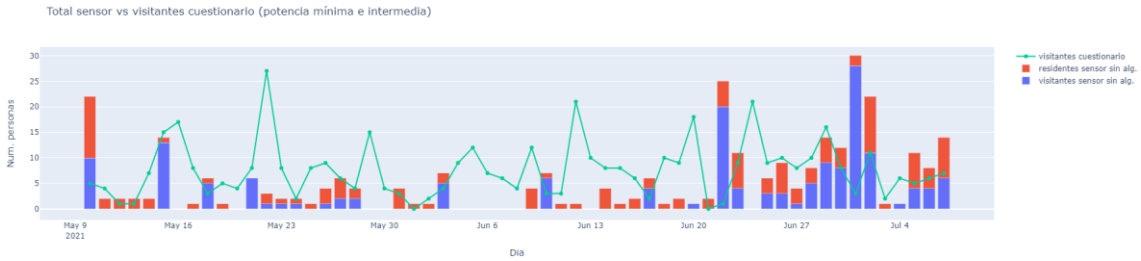


Figura 23. Resultados antes de algoritmo. Pot. Intermedia

Como se observa, en el primer rango hay una falta de detección de visitantes por parte del sensor, y es por esto por lo que nos decantamos por emplear el conocimiento a priori para compensar esta falta de información.

Al utilizar la correlación para medir los resultados de después de aplicar el algoritmo añadiendo la potencia intermedia, esta correlación baja su valor. Es decir, al aplicar la correlación entre los datos de la figura 18 obtenemos un valor de 0.24, y si se aplica solamente al rango en el que se obtienen los datos con potencia intermedia es de -0.062. Esto es debido a la excedencia de las cifras al usar dicha potencia. Más adelante habrá una **discusión** sobre una posible solución y la posibilidad de elección de un rango de potencia mayor en la **propagación** presente proyecto.

7.1. Discusión

A partir de los resultados anteriores se elabora una discusión en la que también se mencionan posibles soluciones que podrían mejorar los resultados obtenidos en la elección final.

La falta de información en el primer rango sugiere la elección del segundo, pero la excedencia en las cifras del sensor respecto a las del cuestionario, disminuyen la calidad de la correlación. Como es de esperar, si añadimos el segundo rango al primero la correlación, es decir si tenemos en cuenta el periodo desde el 10 de mayo hasta el 7 julio,

la correlación un aumenta lo suficiente. Para atajar el problema de la ya comentada excedencia se ha pensado en que se podría hacer de dos formas:

Forma 1: Eliminación de outliers en las cifras recogidas por el sensor, esto consiste en eliminar registros completos (días) en los que el sensor obtiene valores que podrías ser catalogados como anomalías.

Forma 2: Se podría añadir en el algoritmo que los outliers detectados en los datos del se remplacen por la media del día anterior y el siguiente.

Con estas dos formas se podrían eliminar o rebajar esos valores que exceden con creces los del cuestionario.

Una de las razones principales por la que se ha escogido el primer rango como solución final, es la de que al aplicar el algoritmo sólo en ese rango se obtiene una correlación aceptable. Otra de las razones es por el hecho de la practicidad que aporta no superar los valores del cuestionario, y es la obtención de una **cota inferior** de lo que realmente está sucediendo. Es decir, de esta forma se puede afirmar que como mínimo se va a obtener un número de visitantes determinado.

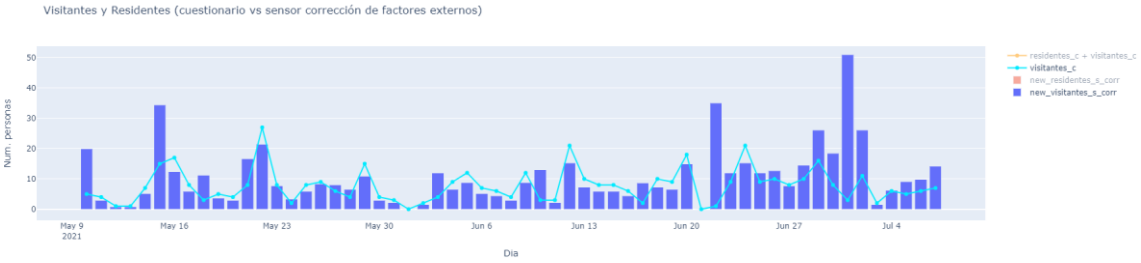
7.2. Otra forma de solventar la wifi inactiva

De entre a las aproximaciones implementadas, merece la pena destacar una en la que nos hicimos la siguiente pregunta: ¿Qué pasaría si todas las personas tuvieran la WiFi de su dispositivo móvil activada? En teoría, si todos los individuos tuvieran wifi, el sensor sería capaz de detectarlos. Por ello, se hizo una aproximación que simulara este suceso. Esto se simuló simplemente de la siguiente forma:

```
df_comparacion['ratio_inc_wifi_vs'] = df_comparacion['visitantes_c'] * 0.402
df_comparacion['new_visitantes_s'] = df_comparacion['visitantes_s'] + df_comparacion['ratio_inc_
wifi_vs']
```

Como se ve, y se **discutirá** más adelante, se recogen los encuestados que no tienen la wifi activa. Después, estas cifras se añaden a las resultantes del sensor. De esta forma, la estacionalidad no hace falta añadirla, pues ya forma parte de la adición realizada. Los

pasos seguidos son los dos primeros: **Discriminación con lista de ssid's y Clustering de frecuencias**. Después de estos pasos, se implementa la porción de código mostrada más arriba, y más adelante una corrección de ajuste usando la media de los cocientes de nuevo (como en otras aproximaciones). Esto se hace para simular la perturbación en los datos que puedan acarrear otros factores externos.



Como se ve los resultados son muy buenos, pero tiene su lógica. Discutiendo lo comentado anteriormente, en esta sección decíamos que, se usa una porción de las personas encuestadas que no tienen la Wifi activa para suplir esa falta de detección por parte del sensor. Realmente se está añadiendo parte de la realidad a nuestras aproximaciones por lo que generaría un sobreajuste y no podría ser reproducible a otras localizaciones.

8. Bibliografía

[1] L. Schauer, M. Werner, and P. Marcus, “Estimating crowd densities and pedestrian flows using Wi-Fi and bluetooth,” 2014, doi: 10.4108/icst.mobiquitous.2014.257870.

[2] X. Liu, J. Wen, S. Tang, J. Cao, and J. Shen, “City-Hunter: Hunting Smartphones in Urban Areas,” 2017, doi: 10.1109/ICDCS.2017.148.

[3] U. Singh, J. F. Determe, F. Horlin, and P. De Doncker, “Crowd Monitoring: State-of-the-Art and Future Directions,” *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*. 2020, doi: 10.1080/02564602.2020.1803152.

[4] U. Gretzel, M. Sigala, Z. Xiang, and C. Koo, “Smart tourism: foundations and developments,” *Electron. Mark.*, 2015, doi: 10.1007/s12525-015-0196-8.

[5] J. A. Ivars-Baidal, M. A. Celdrán-Bernabeu, J. N. Mazón, and Á. F. Perles-Ivars, “Smart destinations and the evolution of ICTs: a new scenario for destination management?,” *Curr. Issues Tour.*, 2019, doi: 10.1080/13683500.2017.1388771.

[6] “Libro Blanco de Destinos Turísticos Inteligentes SEGITTUR.” https://www.segittur.es/destinos-turisticos-inteligentes/proyectos-destinos/libro-blanco-destinos-turisticos-inteligentes/ (accessed Sep. 5, 2021).

[7] HOPU Smart Spots. HOP Ubiquitous S.L. https://smartcities.hopu.eu/smart-spot.html

[8] Wi-Fi positioning system. Wikipedia. https://en.m.wikipedia.org/wiki/Wi-Fi_positioning_system

[9] Burdon, M., & McKillop, A. (2014). The Google street view Wi-Fi scandal and its repercussions for privacy regulation. Monash University Law Review, 39(3), 702-738.

[10] WiGLE: Wireless Network Mapping. https://wigle.net/

[11] Reif, J., & Schmücker, D. (2020). Exploring new ways of visitor tracking using big data sources: Opportunities and limits of passive mobile data for tourism. Journal of Destination Marketing & Management, 18, 100481.

[12] J. Pesonen, E. Horster Near field communication technology in tourism. Tourism Management Perspectives, 4 (2012), pp. 11-18

[13] J.D. Romero Palop, J.M. Arias, D.J. Bodas-Sagi, H.V. Lapaz. Determining the usual environment of cardholders as a key factor to measure the evolution of domestic tourism. Information Technology & Tourism, 21 (1) (2019), pp. 23-43

[14] B. Bonné, A. Barzan, P. Quax, W. Lamotte. WiFiPi: Involuntary tracking of visitors at mass events. IEEE 14th international Symposium on “A world of wireless, mobile and multimedia networks” (WoWMoM) 4–7 june 2013 (2013), pp. 1-6