Universidad del País Vasco
Euskal Herriko Unibertsitatea

Máster Universitario en

Ingeniería **C**omputacional y **S**istemas **I**nteligentes

Konputazio Zientziak eta Adimen Artifiziala Saila –
Departamento de Ciencias de la Computación e Inteligencia Artificial

K I S A

I C S I

Tesis de Máster

Detección de fraude fiscal en alquiler de pisos turísticos mediante técnicas de clasificación positive-unlabeled

Ibon Merino Bermejo

Tutor(a/es)

Dr. Iñaki Inza
Departamento de Ciencia de la Computación e Inteligencia Artificial
Facultad de Informática

Dr. Jerónimo Hernández
Departamento de Ciencia de la Computación e Inteligencia Artificial
Facultad de Informática

informatika fakultatea
facultad de informática

KZAA /CCIA

Septiembre 2018

Resumen

El objetivo principal de estetrabajo final de master consiste en la identificación de alojamientos turísticos fraudulentos a partir de datos extraídos de webs de alojamiento turístico. Se trata de un problema de clasificación semisupervisada o, más concretamente, aprendizaje a partir de datos positivos y no etiquetados. Además de un modelo capaz de detectar el fraude fiscal, también es necesario un método de evaluación del modelo fiable para este tipo de clasificación particular.

The main objective of this master's thesis consists of the identification of fraudulent tourist accommodation from data extracted from tourist accommodation websites. It is a problem of semi-supervised clasification or, specifically, learning from positive and unlabeled data. In addition to a model that detects tax fraud, we also need a reliable evaluation method for this particular classification type.

Master amaierako lan honen helburu nagusia webguneetatik ateratako datuak erabilita iruzurrezko ostatu turistikoen identifikazioan datza. Erdi-gainbegiratutako sailkapen arazo bat da edo, zehazkiago, datu positibo eta etiketatu gabeetatik lortutako ikasketa. Zerga-iruzurra atzeman dezakeen ereduaz gain, eredua balioztatu dezakeen metodo fidagarri bat ere beharrezkoa da sailkapen berezi honetan.

Índice general

Índice de figuras

Indice de tablas

1. CAPÍTULO

Introducción

Estamos en una era donde todo está digitalizado y donde el bien que más generamos son los datos. Se estima que en 2014 el mundo almacenó unos 5 zettabytes. Dada tal cantidad de datos y que las técnicas existentes hasta el momento no eran capaces de procesar toda esa cantidad de datos surgió lo que hoy en día llamamos Big Data para enfrentarse a los retos de almacenamiento, procesamiento y análisis de las nuevas fuentes de datos. Este término que está últimamente en constante auge agrupa técnicas de análisis de datos que nos permiten analizar datos meteorológicos, genómicos, complejos procesos físicos, motores de búsqueda en internet,... [Howe et al., 2008, Chen et al., 2013, Chen et al., 2014, Batty, 2013]

Muchas empresas informáticas hacen uso de estas tecnologías para almacenar esta cuantiosa cantidad de datos y procesarlos para obtener medidas o información más comprensible para el ser humano. Una de estas empresas y que tiene especial relevancia en el sector de la administración pública es Eurohelp Consulting S.L. Este Trabajo de Fin de Master (TFM) se ha hecho en colaboración con esta empresa.

1.1. Eurohelp Consulting S.L

Eurohelp es una empresa líder en el ámbito de las Tecnologías de la Información que ha experimentado el mayor crecimiento de todo el sector en los últimos años. Cuenta con una plantilla próxima a los 200 profesionales y está fuertemente especializada en proporcionar soluciones de alto nivel tecnológico en diferentes ámbitos:

Figura 1.1: Eurohelp Consulting S.L.

- e-Administración
- Facturación electrónica
- Gestión documental
- Buscadores web
- Software libre
- GIS
- Firma digital, PKI
- Sistemas de movilidad
- Web 2.0
- Seguridad informática
- I+D+I

1.2. Proyecto URBEX

El proyecto URBEX o Plataforma de análisis predictivo de datos de los entornos urbanos tiene como objetivo desarrollar una plataforma capaz de dar servicios a los nuevos modelos de ciudad. Se proponen nuevos sistemas de información para la toma de decisiones con la integración de técnicas de inteligencia artificial como aprendizaje automático y minería de datos. Éstos se usan para la analítica no sólo descriptiva, sino incorporando analítica predictiva y preescriptiva para facilitar la toma de decisiones en las ciudades inteligentes.

El proyecto ha sido cofinanciado por el Ministerio de Energía, Turismo y Agenda Digital dentro del Plan de Investigación Científica y Técnica y de Innovación 2013-2016 modificada para la Resolución de la Secretaría de Estado para la Sociedad de la Información y la Agenda Digital de fecha 26 de octubre de 2017 (BOE núm. 279, de 17 de noviembre) en el marco de la Acción Estratégica de Economía y Sociedad Digital.

1.3. Objetivo

Dentro del proyecto URBEX, el objetivo principal de este estudio se central en el alquiler de pisos turísticos del País Vasco. La creación de páginas de alquiler de pisos turísticos como Booking, Airbnb, Wimdu o Homeaway ha aumentado exponencialmente el número de éstos. Estas páginas hacen de intermediarios entre las dos partes, huéspedes y anfitriones, tanto en el momento del contacto como a la hora del pago. Aun así es deber de las partes el debido cumplimiento de sus obligaciones fiscales. Uno de los principales problemas detectados es la necesidad de estudiar la cantidad de estos pisos que eluden sus responsabilidades fiscales. De esta forma, aunque queda fuera de mi TFM, una vez podamos detectar los establecimientos alegales, sería factible estimar la cantidad de impuestos que dejan de recaudar las haciendas de las respectivas provincias.

Inicialmente partimos de que la empresa Eurohelp realizó un web scraping de estas tres webs (Airbnb, Wimdu y Homeaway) para otro proyecto por lo que la parte de obtención y almacenamiento de datos queda fuera de éste. Este proyecto tiene como objetivo principal el análisis de esos datos. La razón por la que se obtienen datos de diversas fuentes es para obtener un mayor cantidad de datos y cubrir la mayor oferta de alojamientos. Esto conlleva otro problema: identificar el mismo alojamiento en diferentes webs. Poder detectar una misma vivienda en las diferentes páginas nos otorga mayor información sobre ese alojamiento, a la vez que se evitan problemas de sobre-representación de casos repetidos.

A continuación se enumeran los objetivos de este proyecto y las diferentes tareas necesarias para lograrlos:

1. Macheo[1] de un mismo alojamiento en distintas webs.

    a) Análisis descriptivo de los alojamientos turísticos ofertados en cada una de las diferentes webs: comparar las distribuciones de los atributos de los alojamientos de cada web.

[1] Macheo: Término anglosajón que se significa identificar dos o más elementos como el mismo

b) Diseño del modelo de macheo de alojamientos.

c) Análisis descriptivo de los alojamientos turísticos una vez macheados.

d) Evaluación del modelo

2. Detección de fraude fiscal.

a) Análisis descriptivos de los precios de los alojamientos turísticos. Esto nos va a permitir hacernos una idea global sobre como varían los precios de los alquileres turísticos.

b) Construcción de un modelo mediante técnicas de aprendizaje automático para la detección de fraude fiscal en alquiler de alojamientos turísticos.

c) Evaluación del modelo.

2. CAPÍTULO

Contexto

La extracción de datos de estas webs (Airbnb, Wimdu y Homeaway) se realiza mediante un proceso de *web scraping*[1]. Este proceso de *web scraping* surge de un proyecto en cooperación con EUSTAT[2]. En líneas generales el proceso de *web scraping* sigue la estructura de la Figura 2.1

Todos los alojamientos que se escrapean son de Euskadi. La mayor parte de los alojamientos se aglutinan en tres ciudades principalmente: Vitoria-Gazteiz (1.31 % de los datos), Bilbao (7.8 %) y San Sebastian-Donostia (51.1 %), es resto de alojamientos están repartidos por los demás pueblos del País Vasco. Además, todos los alojamientos se miran a 120 días (se ha fijado así porque a 4 meses de distancia la mayoria de pisos aun están sin alquilar), esto es, el *web scraper* realiza una consulta de todos los alojamientos que se puedan alquilar a 120 días de la fecha en el que se hace el escrapeo reservando sólo para una única noche.

Este proceso de *web scraping* se realiza para las 4 webs por lo que el scrapeo es distinto para cada una de ellas pero siguen la misma estructura.

[1] El *web scraping* es una técnica utilizada mediante programas de software para extraer información de sitios web.

[2] Por razones de confidencialidad los detalles concretos de este proyecto quedarán excluidos

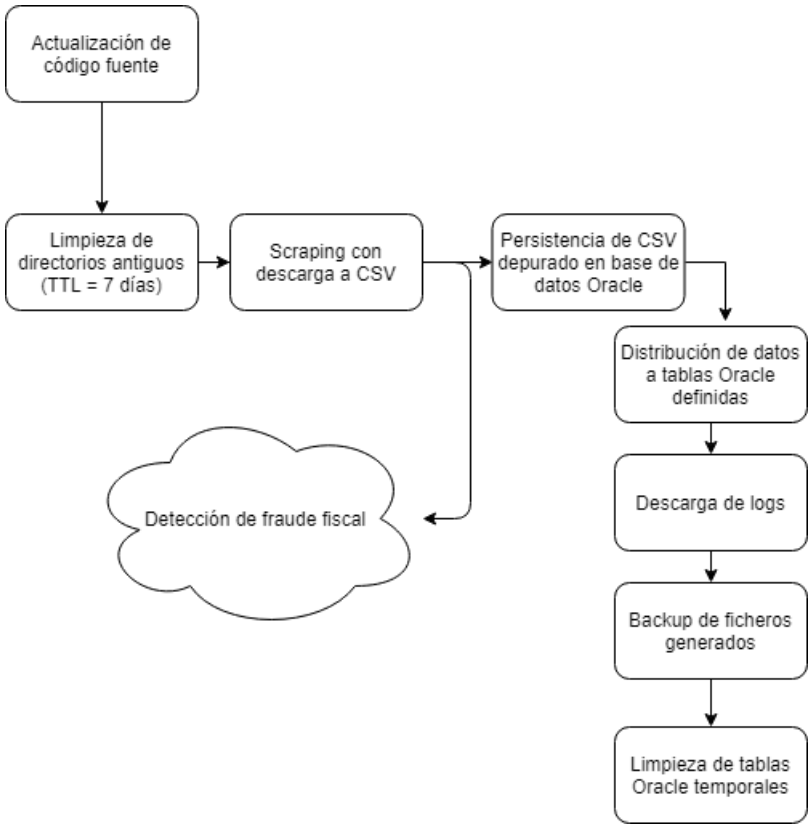

Figura 2.1: Pipeline web scraping

2.1. Airbnb

Es una empresa y una plataforma de software dedicada a la oferta de alojamiento a particulares y turísticos. Esta empresa fue creada en noviembre de 2008 por Brian Chesky, Joe Gebbia y Nathan Blecharcyk en San Francisco, California.

La particularidad de esta plataforma es que la mayoría de las ofertas son habitaciones que tienen libres los anfitriones y que las alquilan a la par que ellos siguen viviendo en la casa. Esta peculiaridad ha generado mucha controversia. La Figura 2.2 es una captura de la página principal.

Por un lado, inicialmente no existía un seguro por el cual cualquier destrozo realizado por parte de los huéspedes fuera sufragado. Debido al gran revuelo que esto generó, actualmente Airbnb dispone de un programa de cobertura el cual cubre hasta un millón de dólares en concepto de responsabilidad por lesiones personales o por daños en la propiedad que se produzcan durante una estancia en un alojamiento en Airbnb en caso de reclamación.

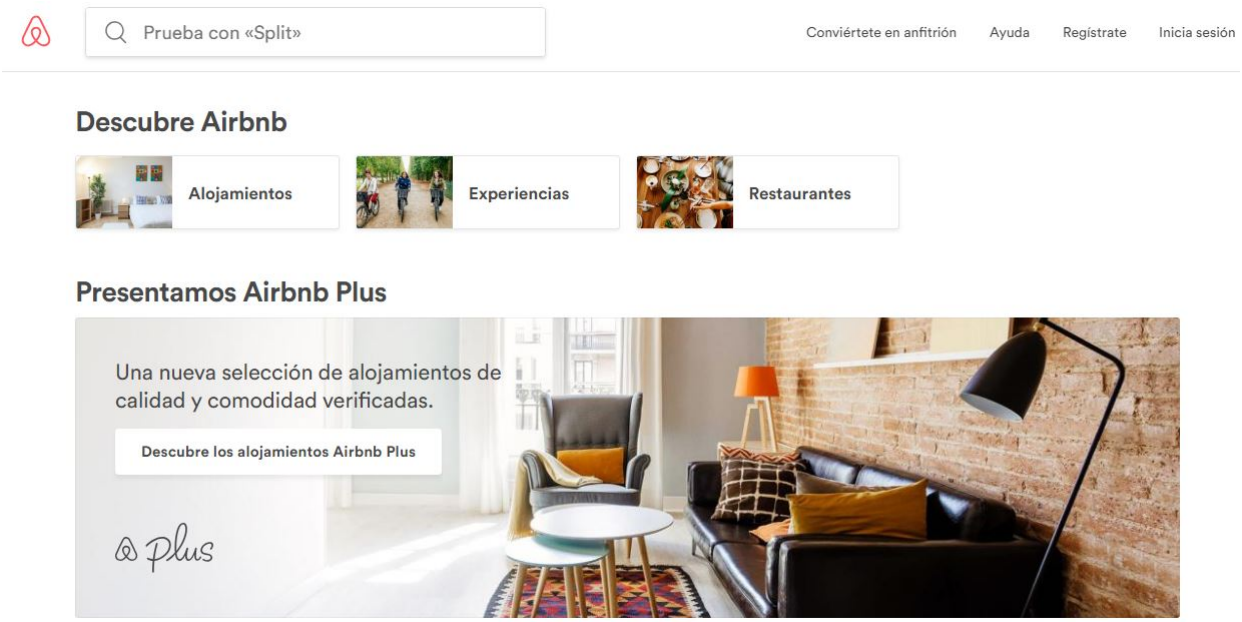

Figura 2.2: Página principal de Airbnb

Por otro lado, vale la pena mencionar la situación jurídica en la que se encuentra Airbnb. Bajo las condiciones de uso de Airbnb se exige que se cumplan los debidos pagos de impuestos locales. Para algunas autoridades locales Airbnb dispone de un acuerdo con la autoridad tributaria correspondiente para recaudar y liquidar impuestos en nombre de los anfitriones. Esto no modifica las obligaciones en materia fiscal de los anfitriones, únicamente simplifica y ayuda a automatizar el proceso para todas las partes. Aquí es donde radica nuestro interés.

En el scrapeo que se realiza para este proyecto, los atributos que se extraen de cada alojamiento son:

- precio/noche: El precio por noche que tiene el alojamiento en esa fecha (120 días desde el momento del escrapeo).
- identificador: Identificador interno de Airbnb presente en la URL que además nos va a permitir relacionar la misma instancia de un scrapeo a otro.
- nº huéspedes: Nº de huéspedes que permite el alojamiento.
- nº dormitorios: Nº de dormitorios de que dispone el alojamiento.
- nº camas: Nº de camas de que dispone el alojamiento.

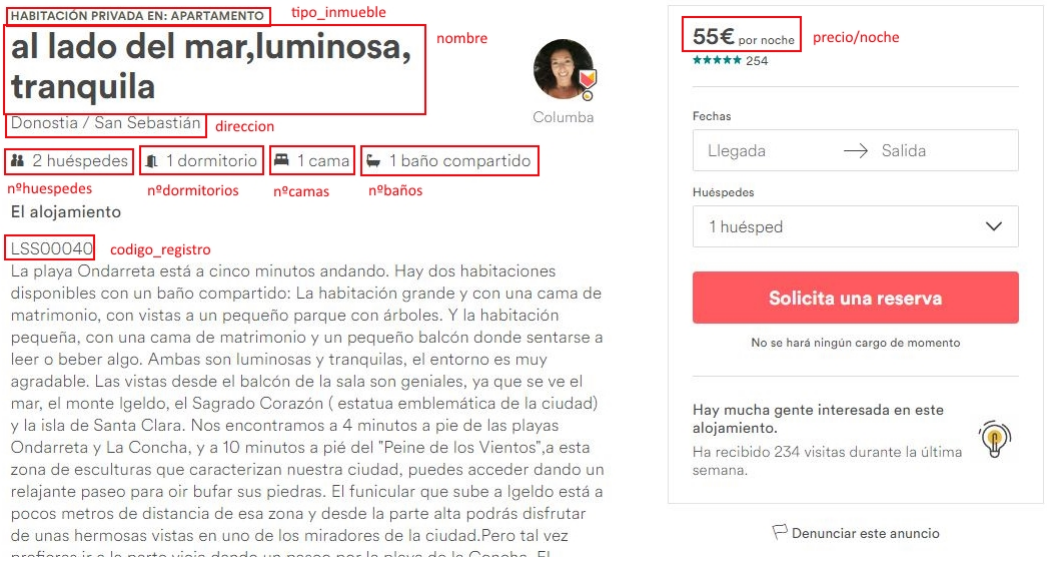

Figura 2.3: Atributos de un alojamiento en Airbnb

- nº baños: Nº de baños de que dispone el alojamiento.
- código_registro: Nº de licencia turística (solo en aquellos casos en que aparezca).
- tipo_inmueble: existen tres tipos de alojamientos:
  - Alojamiento entero: Se dispone de un alojamiento entero.
  - Habitación privada: Se dispone de habitación propia y se comparten algunas zonas comunes.
  - Habitación compartida: Se comparte habitación con otras personas.
- provincia: Provincia en la que se encuentra el alojamiento.
- fecha_scraping: Fecha en la que se realiza el scrapeo.
- checkin: Fecha a 120 días de la fecha de scrapeo, es decir, momento en el que se alquilaría el alojamiento.
- chekout: Fecha posterior a checkin en la que se abandonaría el alojamiento (esta fecha es un día posterior a la fecha de checkin).
- dirección: Ciudad en la que se encuentra el alojamiento.
- nombre: Nombre que los propietarios le dan al alojamiento

En la Figura 2.3 se encuentran marcados los atributos que se extraen de cada uno de los alojamientos de Airbnb.

El identificador se puede obtener a partir de la URL del alojamiento. En la Figura 2.4 podemos observar un ejemplo.

Figura 2.4: URL de un alojamiento en Airbnb

El resto de atributos son características del scraper como el identificador del proceso de scrapeo, por lo que no nos otorgan información sobre los alojamientos de la página de Airbnb.

Airbnb es de las tres páginas la que más alojamientos tiene. Aun así, se estima que es la que, con mayor probabilidad, tiene alojamientos con fraude fiscal debido al gran número de habitaciones privadas y compartidas de que dispone.

2.2. Homeaway

HomeAway, Inc es una compañía de oferta de alquileres turísticos. Fue fundada en 2004 como CEH Holdings. La empresa adquirió diferentes páginas y las consolidó en una única entidad lanzando HomeAway.com en junio de 2006.

Homeaway es la única página web de la que no se realiza un web scraping ya que ésta dispone de una API que nos sirve para obtener los datos que tiene disponible en la web.

La API de HomeAway es un servicio web RESTful el cual se accede a través de la URL "https://ws.homeaway.com/public". Todas las respuestas se devuelven como un archivo JSON. En nuestro caso utilizamos el método search de la API para acotar el rango de búsqueda en la web seleccionando solamente los alojamientos pertenecientes al País Vasco. Un extracto de lo que devuelve la API está recogido en el Anexo A

Los datos se devuelven paginados por lo que hay que llamar a la API varias veces para obtener todos los registros. Los pirmeros atributos del archivo JSON de cada página son los atributos de la paginación:

- nextPage: URL a la siguiente página o NULL si no la hay.
- pageSize: Tamaño de la página.

Figura 2.5: Página principal HomeAway

- pageCount: Número total de páginas
- page: Número de la página actual.
- size: Tamaño total de todas las páginas.
- refinements: Conjunto de links para esta búsqueda, pero refinado para cada uno de los posibles refinamientos.

El último atributo se llama “entries”, es decir, entradas. Este atributo es una lista de los alojamientos para la página actual con estos atributos:

- listingId: Identificador del alojamiento en la web.
- headline: Título del alojamiento.
- description: Descripción del alojamiento.
- accommodations: Listado de habitaciones
- minStayRange: Rango de estancia mínima
  - minStayHigh: Índice superior de estancia mínima
  - minStayLow: Índice inferior de estancia mínima

- thumbnail: Datos sobre la miniatura.
  - height: Altura de la miniatura.
  - imageSize: Tamaño de la imagen.
  - secureUri: Uri segura de la imagen.
  - uri: Uri de la imagen.
  - width: Ancho de la miniatura.
- priceQuote: Datos sobre la cotización.
  - amount: Cantidad.
  - averageNightly: Media por noche.
  - currencyUnits: Unidades actuales/moneda.
  - rent: Renta.
- priceRanges: Lista de detalles sobre los precios del alojamiento.
  - currencyUnits: Unidades actuales/moneda.
  - from: Precio desde.
  - periodType: Tipo de periodo (entre semana o fin de semana).
  - to: Precio hasta.
- location: Detalles sobre la localización.
  - lat: Latitud.
  - lng: Longitud.
  - city: Ciudad.
  - state: Estado/Provincia.
  - country: País.
- regionPath: Path de las regiones en las que está el apartamento.
- reviewCount: Cantidad de opiniones sobre este alojamiento.
- reviewAverage: Media de puntos de este alojamiento en las opiniones.
- bookWithConfidence: Verdadero si este alojamiento esta cuvierto por la garantía "Book with confidence" de HomeAway.

- detailsUrl: URL a los detalles sobre este alojamiento.
- bathrooms: Nº de baños.
- bedrooms: Nº de habitaciones.
- listingUrl: URL para ver el alojamiento en la web de HomeAway.

2.3. Wimdu

Wimdu es una empresa de venta online y servicio de hospedaje en la que permite a los turistas alquileres de corto plazo. Es una empresa que empezó en Marzo de 2011 como startup y fue financiada con 90 millones de dolares, la cual fue la mayor inversión en una startup hasta el momento.

Figura 2.6: Página principal Wimdu

La lista de atributos de los alojamientos de esta web son:

- precio: El precio por noche que tiene el alojamiento en esa fecha (120 días desde el momento del escrapeo).
- linkid: Identificador interno de Wimdu presente en la URL que además nos va a permitir relacionar la misma instancia de un scrapeo a otro.
- aforo: Nº de huéspedes que permite el alojamiento.
- dormitorios: Nº de dormitorios de que dispone el alojamiento.
- aseos: Nº de baños de que dispone el alojamiento.

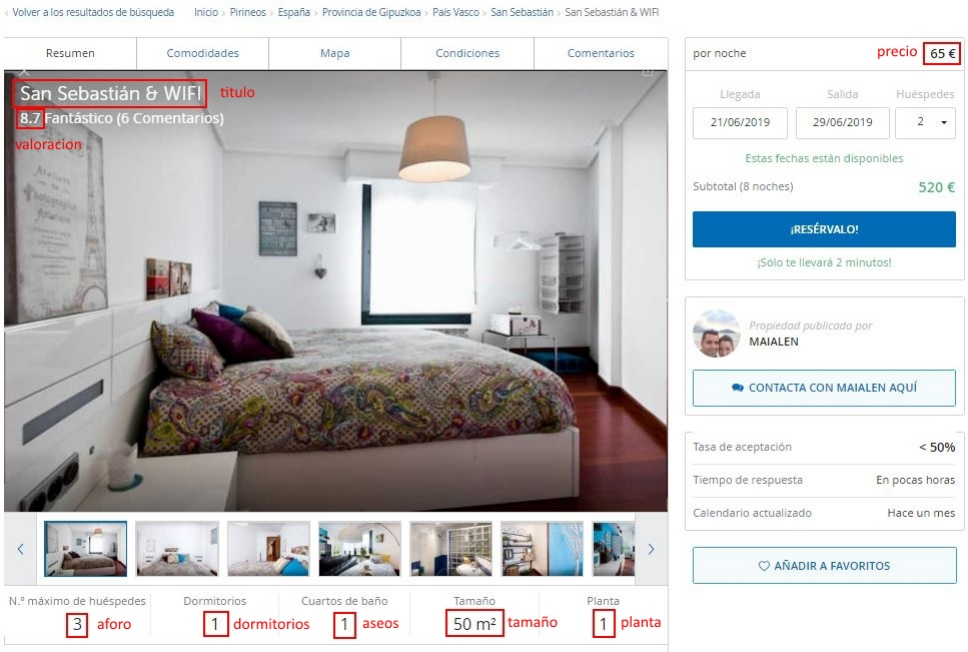

Figura 2.7: Atributos de un alojamiento de wimdu 1

Figura 2.8: Atributos de un alojamiento de wimdu 2

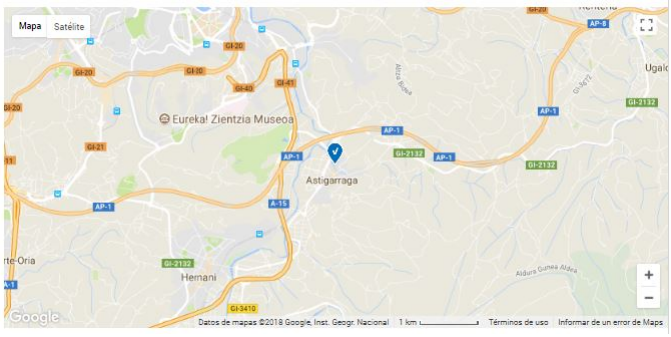

Figura 2.9: Mapa de la localización de un alojamiento de wimdu

- tamaño: Nº de metros cuadrados disponibles del alojamiento.
- código_registro: Nº de licencia turística (solo en aquellos casos en que aparezca).
- tip_propiedad: Existen 10 tipos de alojamientos (Apartamento, Casa, Habitación privada, Chalet, Cabaña, Casa de campo, Castillo, Casa en un árbol, Barco, Automóvil)
- checkin: Fecha a 120 días de la fecha de scrapeo, es decir, momento en el que se alquilaría el alojamiento.
- chekout: Fecha posterior a checkin en la que se abandonaría el alojamiento (esta fecha es un día posterior a la fecha de checkin).
- dirección: Dirección en la que se encuentra el alojamiento.
- titulo: nombre que los propietarios le dan al alojamiento.
- enlace: URL para acceder a la página del alojamiento de Wimdu.
- longitud: Longitud en la que se encuentra el alojamiento.
- latitud: Latitud en la que se encuentra el alojamiento.
- extras: Lista de extras que dispone el alojamiento.
- planta: Planta en la que se encuentra el alojamiento.
- descripción: Descripción que dan los propietarios sobre el alojamiento. Puede estar en varios idiomas.
- valoración: Media de las valoraciones obtenidas por los usuarios que han pasado una estancia en ese alojamiento.
- min_noches: Mínimo de noches que hay que alquilar el alojamiento.

Una cuestión interesante en los alojamientos de esta web es la localización del alojamiento. En la Figura 2.9 podemos observar la localización de uno de los alojamientos. Esta localización en muchos de los casos no es exacta pero sirve a modo de orientación a la par que otorga información sobre el alojamiento que se puede utilizar para el macheo entre las fuentes.

3. CAPÍTULO

Macheo de fuentes: identificación del mismo alojamiento en distintas webs

Antes de empezar a explicar el macheo de fuentes, en la Figura 3.1 tenemos un listado comparativo de los atributos que obtenemos de cada uno de los apartamentos de las distintas webs.

Dado que no existen muchos atributos que estén presentes y coincidan en las 3 fuentes y partiendo de que la identificación del mismo elemento procedente de distintas fuentes no es trivial, se han realizado varias aproximaciones.

Estas aproximaciones son las diferentes combinaciones entre algoritmos de clasificación y distintos subconjuntos de variables seleccionadas.

3.1. Código de registro

Este código es el número de licencia turística que se obtiene al registrar un alojamiento como apartamento turístico en el Organismo Oficial correspondiente(ej., en Donostia, en Bilbao...). Este número es independiente de la fuente de datos por lo que el mismo apartamento tendrá (si está incluido) el mismo número de licencia.

Este macheo no precisa de ninguna técnica de machine learning; es un macheo directo, determinista, que nos sirve para machear todos los alojamientos que dispongan de este atributo. Aun así, como el mismo código de registro puede estar escrito de distintas

airbnb	**wimdu**	**homeaway**
precio/noche	precio	precio
identificador	linkid	id
nºhuespedes	aforo	aforo
nºdormitorios	dormitorios	dormitorios
nºcamas		
nºbaños	aseos	aseos
	tamaño	
codigo_registro	codigo_registro	codigo_registro
tipo_inmueble		
provincia		estado
fecha_scraping		
checkin	checkin	checkin
chekout	checkout	checkout
direccion	direccion	direccion
nombre	titulo	titulo
	enlace	enlace
	longitud	longitud
	latitud	latitud
	extras	
	planta	
	descripcion	descripcion
	valoracion	
	min_noches	
	tip_propiedad	
		ciudad
		disponible
		pais

Tabla 3.1: Comparativa de los atributos de las fuentes

maneras, con o sin guiones y espacios, se eliminan todos los espacios y guiones de este campo para que el macheo sea más fiable.

3.2. Algoritmos

3.2.1. Vecino más cercano

El algoritmo K-NN[1] o k vecinos más cercanos [Bhatia et al., 2010] es un método de clasificación supervisada basado en un entrenamiento mediante ejemplos cercanos en el espacio de los elementos.

Cada instancia está descrita en términos de p atributos considerando todas las clases $c \in C$ para la clasificación.

$$\boldsymbol{x_i} = (x_{1i}, x_{2i}, \ldots, x_{pi}) \in X$$

Siendo $d(\boldsymbol{x_i}, \boldsymbol{x_j})$ una distancia[2] definida, un punto es asignado a la clase c si ésta es la clase más frecuente entre los k ejemplos de entrenamiento más cercanos. Asumamos que $f(\boldsymbol{x})$ es una función que devuelve el valor real de la clase para la instancia $\boldsymbol{x}$.

Dado un ejemplar $\boldsymbol{x_q}$ que debe ser clasificado, sean $\boldsymbol{x_1}, \ldots, \boldsymbol{x_k}$ los k vecinos más cercanos a $\boldsymbol{x_q}$ entre los ejemplos de aprendizaje, el clasificador devuelve

$$\hat{f}(\boldsymbol{x_q}) = argmax_{c \in C} \sum_{i=1}^{k} \delta(c, f(\boldsymbol{x_i}))$$

donde δ es la delta de Kronecker.

[1] K-Nearest Neighbors

[2] Generalmente esta distancia es la distancia euclídea

$$d(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sqrt{\sum_{r=1}^{p} (x_{ri} - x_{rj})^2}$$

En el caso de varialbes cualitativas se utilizará una delta de Kronecker

$$\delta(a,b) = \begin{cases} 1 & si \quad a = b \\ 0 & si \quad a \neq b \end{cases}$$

El valor $\hat{f}(\boldsymbol{x_q})$ devuelto por el algoritmo como estimador de $f(\boldsymbol{x_q})$ es solo el valor más común de f entre los k vecinos más cercanos a $\boldsymbol{x_q}$.

Para machear dos alojamientos se utilizará un 1-NN donde el atributo identificador de una de las fuentes es la clase y se predicen las instancias de la otra fuente.

3.2.2. Radius NN

El algoritmo Radius NN [Bentley et al., 1977] es muy similar al KNN . La principal diferencia reside en que en vez de coger los k vecinos más cercanos, dada una distancia fija Δ y un caso a clasificar $\boldsymbol{x_q}$, este caso se clasifica como la clase más frecuente entre todos los vecinos dentro de la distancia Δ.

Gracias a este método podemos ser más exigentes en cuanto a la hora de clasificar. Si no existe ningún caso dentro de la distancia Δ, dejaremos $\boldsymbol{x_q}$ sin clasificar, cosa que en nuestro caso puede ocurrir. Un alojamiento no tiene porqué estar en todas las webs.

3.3. Selección de variables

3.3.1. Aforo, aseos y dormitorios

Estos tres atributos tienen que ser completamente iguales entre dos alojamientos para que se puedan clasificar como iguales. Gracias a esto se pueden descartar muchos casos. Para incluir esto en el algoritmo KNN tenemos que modificar como calculamos la distancia.

$$d(\boldsymbol{x_i},\boldsymbol{x_j}) = \sqrt{\sum_{r=1}^{p} diff(x_{ri},x_{rj})}$$

$$diff(x_{ri},x_{rj}) = \begin{cases} 0 & si \quad \text{r es aforo, aseos o dormitorios \& } x_{ri} = x_{rj} \\ \text{ínf} & si \quad \text{r es aforo, aseos o dormitorios \& } x_{ri} \neq x_{rj} \\ (x_{ri} - x_{rj})^2 & \text{en otro caso.} \end{cases}$$

3.3.2. Precio

El precio de un mismo alojamiento puede ser distinto en las cada una de las fuentes por lo que habrá que testear si esta variable es importante a la hora de machear los alojamientos

o no.

3.3.3. Dirección y descripción

La dirección y la descripción están compuestas por texto por lo que existen muchas formas de transformar un texto a variables. Se han planteado dos formas:

1. Conteo de palabras: Una variable nueva por cada palabra distinta que aparezca en el texto y su valor es la cantidad de veces que aparezca esa palabra.
2. TFIDF[3]: es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección [Joachims, 1996]. Primero hay que realizar un conteo de palabras y tras eso calcular el TFIDF de cada palabra.

Una vez transformados los textos a variables cuantitativas (mediante el uso del conteo de palabras o el TFIDF) podemos utilizarlas con cualquiera de los algoritmos previamente presentados.

De cara a entender mejor la variable descripción y tener una visión general del contenido de ésta se ha generado la Figura 3.1. En esta figura se muestra un *word cloud* o nube de palabras creado a partir del contenido de la variable descripción de los datos escrapeados en un día en Homeaway. El *word cloud* es un gráfico que sirve para representar la cantidad de veces que aparece cada término en un texto mediante el tamaño de letra. En nuestro caso el texto es la concatenación de todas las descripciones. Para darle estilo al *word cloud* se ha utilizado una mascara con la forma de Euskal Herria.

Podemos apreciar que muchos de los términos más grandes son "FeelFree Rentals", "people rentals", . . . estas son agencias que gestionan viviendas vacacionales.

3.3.4. Geolocalización

Esta aproximación sólo puede utilizarse para machear instancias de Wimdu y Homeaway ya que las de Airbnb no disponen de este atributo. Está compuesto por dos variables: longitud y latitud. Como la localización es aproximada y muchos alojamientos se parecen en cuanto a sus coordenadas, se aplica la función Sigmoide. Al utilizar esta función se

[3]Term frequency - Inverse document frequency o Frecuencia de término - Frecuencia inversa de documento

Figura 3.1: *Word cloud* de la variable descripción

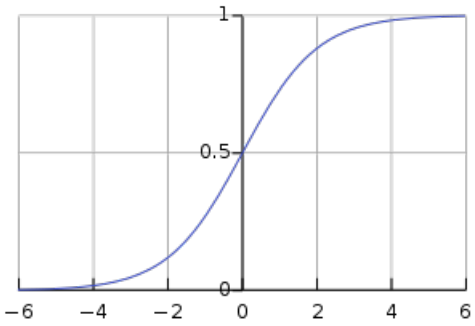

Figura 3.2: Función Sigmoide

consigue dar mayor importancia a cambios pequeños, es decir, a pequeñas diferencias se obtiene ampliar esas diferencias.

$$d(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sqrt{\sum_{r=1}^{p} diff(x_{ri}, x_{rj})}$$

$$diff(x_{ri}, x_{rj}) = \begin{cases} sigmoid((x_{ri} - x_{rj})^2) & si \quad \text{r es longitud o latitud} \\ (x_{ri} - x_{rj})^2 & \text{en otro caso.} \end{cases}$$

La función sigmoide tiene la forma de la Figura 3.2 y viene definida por la siguiente formula:

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

3.4. Macheo

Las variables predictoras que servirán de base a los algoritmos son las diferentes combinaciones de los atributos que se han expuesto en la sección 3.3. Por ejemplo, uno de las combinaciones que se van a utilizar sería la geolocalización, aseos, aforo y dormitorios para entrenar un modelo 1-NN. Se entrenarán modelos con todas las combinaciones posibles y se calculará cuál es el mejor modelo para machear dos fuentes.

4. CAPÍTULO

Positive Unlabelled Learning: Aprendizaje de modelos predictivos desde datos positivos y no etiquetados

Los métodos de clasificación son muy utilizados actualmente para automatizar tareas de categorización que requerirían de supervisión humana. Existen distintos tipos de escenarios de clasificación [Hernández-González et al., 2016]: clasificación supervisada, clasificación semi-supervisada, clasificación no supervisada, positive unlabelled learning, one class classification, etc.

El tipo de clasificación más tradicional y a priori el más simple porque se dispone de más aplicaciones reales es la clasificación supervisada [Richards, 2006]. Muchas veces, no es posible utilizar este tipo de clasificación ya que los datos en la realidad no son ni tan uniformes ni tan fáciles de supervisar. Aún así, estos algoritmos se suelen utilizar como base e inspiración para otros algoritmos más elaborados que o mejoran el rendimiento de éstos o posibilitan utilizar datos incompletos. Un dataset de entrenamiento en clasificación supervisada estaría compuesto por n instancias con p atributos y etiquetadas cada una con una clase $c \in C$. El objetivo es aprender un modelo que sea capaz de anotar una nueva instancia $\boldsymbol{x}$ sin etiquetar.

En la clasificación semi-supervisada una pequeña porción de los datos de entrenamiento están etiquetados (estando todas las clases posibles presentes) y una gran parte de los datos sin etiquetar. Lo que se pretende es no solo aprender de los datos etiquetados sino también de los que no lo están.

El estudio central de este TFM radica en el Positive Unlabelled learning (PU) o aprendi-

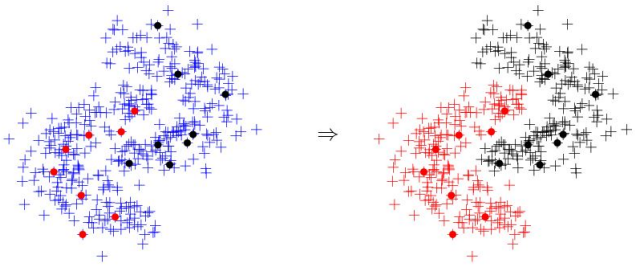

Figura 4.1: Semi-supervised clasification

zaje a partir de datos positivos y no etiquetados. En este tipo de problemas disponemos de dos conjuntos de datos de entrenamiento: un conjunto P de casos positivos y un conjunto U de no etiquetados. Esto se debe a la incertidumbre en la clase a anotar ya sea por el gasto que conlleva etiquetar los casos o por la ausencia de un experto capaz de hacerlo.

Las soluciones al problema de PU learning se podrían, grosso modo, dividir en tres grupos. En el primer grupo tendríamos algoritmos que tratan las instancias no etiquetadas como negativas. Para mejorar estos algoritmos se combinan diferentes clasificadores como en [Calvo et al., 2007b] y en [Sriphaew et al., 2009] o usando pesos para los no etiquetados como en [Elkan and Noto, 2008], [Liu and Lee, 2003], [Liu et al., 2003] y [Zhang and Lee, 2005].

El segundo grupo se basa en algoritmos en dos pasos. Entre ellos están el spy-EM de [Liu et al., 2002] basado en el EM (Expectation-Maximization) de [Dempster et al., 1977]. Otros algoritmos pueden encontrarse en [Li and Liu, 2003], [Pan et al., 2012] y [Yu et al., 2004].

El último grupo se basa en extraer información directamente de la distribución de probabilidad de las instancias no etiquetadas. Entre estos algoritmos están [Denis, 1998] basado en el modelo de aprendizaje PAC (Probably Approximately Correct learning o aprendizaje correcto probablemente aproximado) de [Valiant, 1984], clasificadores redes Bayesianas positivas (PBCs) [Calvo, 2008, Calvo et al., 2007a, Denis et al., 2002] como el Naive Bayes de [Minsky, 1961] o el tree-augmented Naive Bayes (TAN) de [Friedman et al., 1997].

Recientemente, la clasificación a partir de datos positivos y no etiquetados está recibiendo cada vez más atención [Elkan and Noto, 2008, Du Plessis et al., 2014, Du Plessis et al., 2015, Jain et al., 2016]. [Sakai et al., 2016] proponen un método que combina PU con métodos tradicionales de clasificación supervisada, el cual no necesita de las restrictivas asunciones sobre distribuciones: se asume que los datos siguen unas distribuciones

concretas. Se ha demostrado que funciona excelentemente. Por el mismo hilo, proponen un método de optimización de la AUC que no requiera de asunciones restrictivas [Sakai et al., 2018].

En cuanto a nuestro problema, disponemos de casos etiquetados (aquellos casos que tienen código de registro) y no etiquetados (el resto de casos). Debido a esto nos encontramos en un problema de PU y decidimos abordarlo mediante varias técnicas.

4.1. Gaussian Naive Bayes

El clasificador Naive Bayes [John and Langley, 1995] es una aproximación simple pero que, por lo general, suele obtener buenos resultados en aplicaciones reales. Este clasificador es un tipo de red Bayesiana llamada "naive" porque tiene en cuenta dos importantes asunciones de simplificación. Estas dos asunciones son:

1. Las variables predictoras son condicionalmente independientes entre ellas dada la clase.

2. Que no existen atributos ocultos o latentes que influencien el proceso de predicción.

Con este modelo reducimos el número de parámetros que tenemos que estimar. Ésta no es una técnica de PU pero va a ser necesaria para las técnicas que se van a explicar a continuación.

Sean C la variable aleatoria que denota la clase de una instancia, $\boldsymbol{X}$ un vector de variables aleatorias, c la representación de una etiqueta de C y $\boldsymbol{x}$ la representación de un vector particular de los atributos observados. Dado un caso de test $\boldsymbol{x}$ a clasificar utilizando la regla de Bayes podemos calcular las probabilidades de cada clase.

$$p(C = c|\boldsymbol{X} = \boldsymbol{x}) = \frac{p(C = c)p(\boldsymbol{X} = \boldsymbol{x}|C = c)}{p(\boldsymbol{X} = \boldsymbol{x})} \tag{4.1}$$

De esta forma podemos predecir cuál es la clase más probable. $\boldsymbol{X} = \boldsymbol{x}$ representa $X_1 = x_1 \wedge X_2 = x_2 \wedge \ldots \wedge X_k = x_k$ y dado que el NB asume que los atributos son condicionalmente independientes entre ellos una vez conocida la clase, obtenemos:

$$p(\boldsymbol{X} = \boldsymbol{x}|C = c) = p(\bigwedge_i X_i = x_i|C = c) = \prod_i p(X_i = x_i|C = c)$$

En cuanto al denominador de la ecuación 4.1 normalmente no se suele calcular, se ignora, ya que no depende de C y es el mismo para todas las etiquetas c. Luego se normaliza de forma que la suma de $p(C = c|X = x)$ de todas las clases sea uno.

Naive Bayes trata de forma distinta las variables continuas de las discretas. Para cada atributo discreto, $p(X = x|C = c)$ es modelado por un único valor entre 0 y 1 que representa la probabilidad de que el atributo X vaya a tener el valor particular x cuando la clase es c. Por el contrario, cada atributo continuo es modelado por una distribución de probabilidad continua para todo el rango de los valores del atributo.

Asumimos que las variables numéricas siguen una distribución normal (o Gaussiana) para que de esta forma podamos calcular las probabilidades condicionales mediante la probabilidad de la función de densidad g. [John and Langley, 1995]

$$p(X = x|C = c) = g(x; \mu_c, \sigma_c) \tag{4.2}$$

$$g(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{4.3}$$

donde x es el valor de la variable predictora, μ la media de la distribución , μ_c la media de la distribución que sigue la variable para la clase c, σ la desviación típica de la distribución y σ_c la desviación típica de la distribución que sigue la variable predictora para la clase c.

4.2. EM

El algoritmo Expectation-Maximization (EM) [Dempster et al., 1977, Liu et al., 2002] es un popular algoritmo iterativo para estimaciones de máxima verosimilitud en problemas con datos incompletos. El algoritmo EM consiste en dos pasos, el paso de Esperanza y el paso de Maximización. El paso de Esperanza básicamente se encarga de imputar los datos faltantes. La estimación de los parámetros se realizan en el paso de Maximización dados los datos completos tras la imputación del paso anterior. Este algoritmo garantiza que siempre se mejora la verosimilitud y que converge a un máximo, ya sea local, global o un punto de silla.

El algoritmo EM es un mecanismo de optimización que trabaja sobre un clasificador y que ayuda a aprender los parámetros del mismo. En nuestro caso hemos decidido usar el Naive Bayes Gaussiano que hemos explicado en el apartado 4.1 como base de este algoritmo.

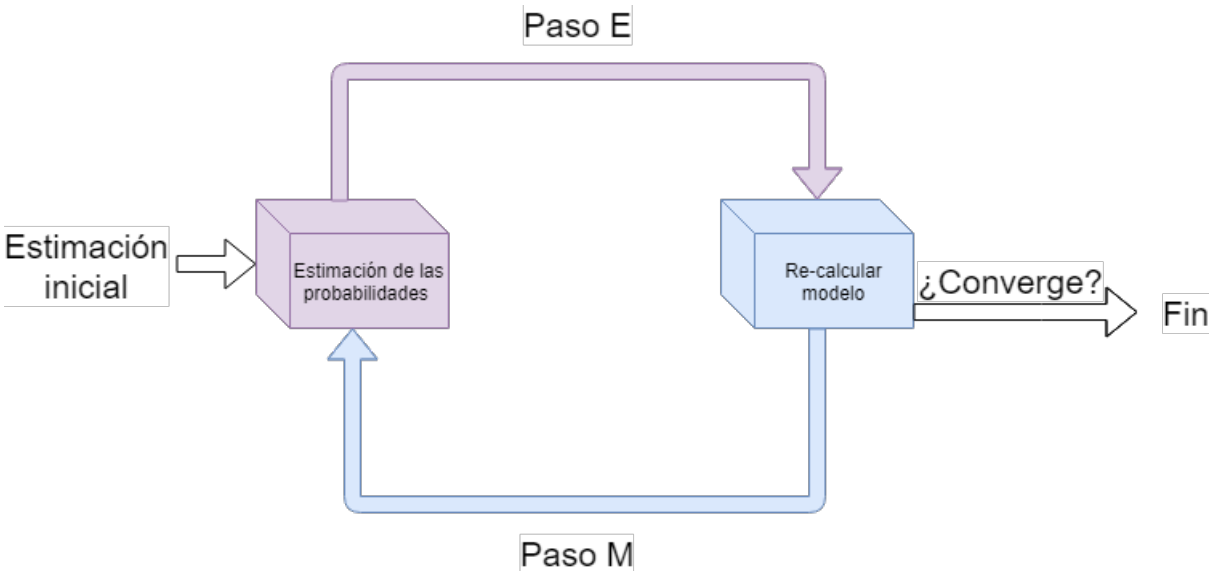

Figura 4.2: Estructura general del algoritmo EM

Lo que nos interesa de este algoritmo es la habilidad que tiene de trabajar con datos faltantes, en este caso, las clases del conjunto de datos U.

I-EM(U, P):
Datos: conjunto de casos no etiquetados U, conjunto de casos positivos P
Construir el clasificador inicial Naive Bayes NB utilizando los conjuntos de datos U y P;
mientras *la función de máxima verosimilitud mejore* **hacer**
  para $x_j \in U$ **hacer**
    Calcular $p(C = c_1|x_j)$ utilizando el clasificador NB actual;
    //$p(C = c_2|x_j) = 1 - p(C = c_1|x_j)$
    Actualizar $p(x_j|C = c_1)$ y $p(C = c_1)$ utilizando los valores de $p(C = c_1|d_j)$ obtenidos y P;
  fin
  Construir un nuevo clasificador utilizando como probabilidades las obtenidas al clasificar los casos x_j;
fin

Algoritmo 1: I-EM con un clasificador Naive Bayes.

Inicialmente, para cada caso $\boldsymbol{x_i}$ en P se le asigna la clase c_1 ($p(c_1|x_i) = 1$ y $p(c_2|x_i) = 0$) y para cada $\boldsymbol{x_j}$ en U se le asigna $p(c_1|x_j) = 0.5$ y $p(c_2|x_j) = 0.5$

Un dato que es necesario para obtener un buen resultado utilizando el algoritmo EM es aportar una buena probabilidad a priori de las clases. Esto es crítico ya que no disponemos de la forma de calcularla debido a la ausencia de etiquetas en algunos de los ejemplos.

Por ello, éste será un parámetro que habrá que pasar al algoritmo.

4.3. Gaussian Mixture

Un modelo de mezcla de Gaussianas [Reynolds et al., 2000] es un modelo probabilístico que sirve para representar la presencia de varias subpoblaciones normalmente distribuidas dentro de una población general. En la Figura 4.3 podemos ver un ejemplo de una mezcla Gaussiana de 3 componentes.

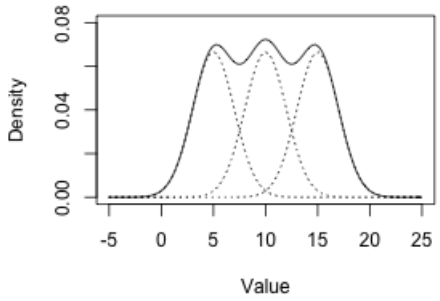

Figura 4.3: Mezcla de 3 Gaussianas

Un modelo de mezcla de Gaussianas está parametrizado por dos tipos de valores: los pesos de cada componente de la mezcla, y las medias y varianzas/covarianzas de los componentes. Para un modelo de mezcla de Gaussianas con K componentes, la componente k-esima tiene una media μ_k y una varianza σ_k para el caso univariado y una media $\vec{\mu}_k$ y una matriz de covarianzas Σ_k para el caso multivariado. Los pesos de las componentes de la mezcla se definen como ϕ_k para el componente C_k, con la condición que $\sum_{i=1}^{K} \phi_i = 1$.

El modelo univariado se explica a través de las siguientes ecuaciones:

$$p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x|\mu_i, \sigma_i) \tag{4.4}$$

$$\mathcal{N}(x|\mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} exp(-\frac{(x-\mu_i)^2}{2\sigma_i^2}) \tag{4.5}$$

$$\sum_{i=1}^{K} \phi_i = 1 \tag{4.6}$$

mientras que el modelo multivariante se explica de la siguiente manera:

$$p(\vec{x}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\vec{x}|\vec{\mu}_i, \Sigma_i) \tag{4.7}$$

$$\mathcal{N}(\vec{x}|\vec{\mu}_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^K |\Sigma_i|}} exp(-\frac{1}{2}(\vec{x} - \vec{\mu}_i)^T \Sigma_i^{-1}(\vec{x} - \vec{\mu}_i)) \tag{4.8}$$

$$\sum_{i=1}^{K} \phi_i = 1 \tag{4.9}$$

Si el número de componentes K es conocido, es común utilizar el algoritmo EM para estimar los parámetros del modelo de la mezcla.

En este caso el paso E consiste en calcular la esperanza de las asignaciones de las componentes C_k para cada valor de $\boldsymbol{x} \in X$ dados los parámetros del modelo ϕ_k, μ_k y σ_k:

$$\hat{\gamma}_{ik} = \frac{\hat{\phi}_k \mathcal{N}(x_i|\hat{\mu}_k, \hat{\sigma}_k)}{\sum_{j=1}^{K} \hat{\phi}_j \mathcal{N}(\boldsymbol{x_i}|\hat{\mu}_j, \hat{\sigma}_j)}, \tag{4.10}$$

donde $\hat{\gamma}_{ik}$ es la probabilidad de que $\boldsymbol{x_i}$ sea generado por el componente C_k. Por ello, $\hat{\gamma}_{ik} = p(C_k|\boldsymbol{x_i}, \hat{\phi}, \hat{\mu}, \hat{\sigma})$.

El paso M consiste en actualizar los valores de los parámetros del modelo ϕ_k, μ_k y σ_k. Utilizando las $\hat{\gamma}_{ik}$ calculadas en el paso de esperanza, $\forall k$ se calcula:

$$\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N} \tag{4.11}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}} \tag{4.12}$$

$$\hat{\sigma}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} (x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \hat{\gamma}_{ik}} \tag{4.13}$$

Cuando el número de componentes K es desconocido a priori, se intenta averiguar aprendiendo modelos con distintos valores de K y eligiendo el mejor.

4.4. Spy-EM

El algoritmo EM funciona muy bien para datasets "fáciles" donde los casos positivos y negativos son fácilmente separables. En el caso de tener datasets "difíciles", con solapamiento de clases, no funciona tan bien. Esto se debe a que la inicialización está altamente sesgada por los casos positivos. Para ello, [Liu et al., 2002] proponen una técnica llamada Spy-EM para hacer frente a este problema.

Para solventar este problema plantean identificar primero los casos más probablemente negativos del conjunto de datos U para tener una inicialización más robusta. Tras utilizar el I-EM (Algoritmo 1) , nos encontramos en una buena posición para identificar aquellos que son con alta probabilidad negativos. El problema reside en cómo obtener información de confianza para identificarlos. Se envían unos "espías" pertenecientes al conjunto P hacia el conjunto U. Seleccionamos aleatoriamente $s\%$ de los casos del conjunto P (en [Liu et al., 2002] utilizan un 10%). Estos casos son los espías, denotados como S. Los espías se tratan idénticamente como los no etiquetados y nos ayudan a inferir cómo se comportan los casos positivos desconocidos.

Inicialmente se utiliza el Algoritmo 1 I-EM con el conjunto U el cual contiene ahora tiene algunos casos espías. Una vez acabe el algoritmo I-EM, con las probabilidades obtenidas se decide qué casos son los más probables de ser negativos. Se calcula un valor límite t para tomar esta decisión. Los casos que tengan una probabilidad $p(C = c_1|d_j)$[1] menor que t se consideran los más probables de ser negativos y se denotan como N. Los casos de U (excluidos los espías) que tengan una probabilidad mayor que t seguirán considerándose no etiquetados y de ahora en adelante los denotaremos como U_{new}. En el algoritmo 2 se muestra cómo obtener estos tres nuevos conjuntos P, N y U_{new}.

Spy-EM-1(P, U):

[1] Consideraremos la clase c_1 como la clase positiva y la clase c_2 como la negativa

Datos: U, P
$N = U_{new} = \phi$;
$S = muestra(P, s\%)$;
$US = U \cup S$;
$P = P - S$;
Asignar a cada caso d_j en P la clase c_1;
Asignar a cada caso d_j en US la clase c_2;
Lanzar I-EM(US, P);
Clasificar cada caso d_j en US;
Determinar el límite de probabilidad t usando S;
para $d_j \in U$ **hacer**
  si *la probabilidad* $p(C = c_1|d_j) < t$ **entonces**
    $N = N \cup \{d_j\}$;
  en otro caso
    $U_{new} = U_{new} \cup \{d_j\}$;
  fin
fin

Algoritmo 2: Paso 1: Identificando posibles casos negativos

Ahora pasemos a definir cómo calculamos el límite t. Sean el conjunto de espías S $\{\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots, \boldsymbol{s_k}\}$ y la etiqueta probabilística asignada a cada s_i $p(c_1|s_i)$, podemos utilizar la probabilidad mínima de S como límite t (p.e., $t = min\{p(c_1|s_1), p(c_1|s_2), \ldots, p(c_1|s_k)\}$) con lo que nos quedaríamos con todos los elementos de S. En un dominio sin ruido, utilizar este límite es aceptable, pero la mayoría de casos contienen outliers o ruido. Utilizar el mínimo es inviable ya que la probabilidad $p(c_1|s_i)$ de un documento s_i outlier de S podría ser 0 o mucho menor que la mayoría de elementos negativos. Como no sabemos el nivel de ruido en los datos, podemos realizar una estimación de éste. Seleccionamos un nivel de ruido l de forma que elegimos t tal que un $l\%$ de los casos tienen probabilidad menor que t. En [Liu et al., 2002] demuestran que no existe una gran diferencia entre usar valores entre 5 y 20, por lo que deciden utilizar l=15 %.

En definitiva, lo que se intenta obtener en este primer paso es el resultado expuesto en la Figura 4.4. La parte izquierda muestra la situación inicial. En el conjunto mezcla tenemos tanto casos positivos como casos negativos, pero desconocemos su etiqueta. Los espías del conjunto de positivos se añaden al conjunto mezcla. La parte derecha, en cambio, muestra el resultado que se obtiene tras aplicar esta técnica gracias a la ayuda de los espías. Podemos observar que la mayoría de los casos positivos se han quedado en el conjunto sin etiquetar y la gran parte de los negativos acaban en el conjunto de posibles

negativos N. La pureza del conjunto N es mucho mayor que la del conjunto mezcla inicial.

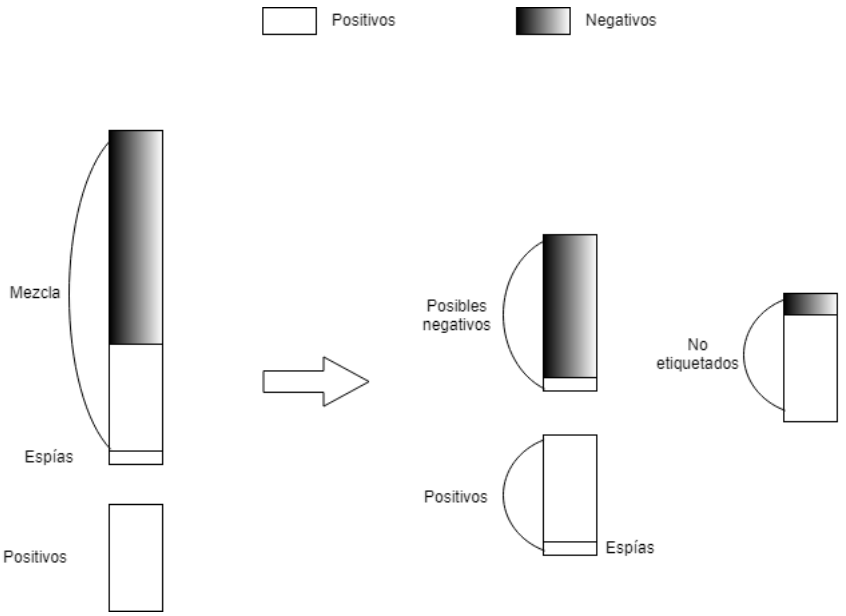

Figura 4.4: Antes y después del paso 1: Reinicialización

El segundo paso construye el clasificador final. Se utiliza de nuevo el algoritmo EM pero con el conjunto de datos P, N y U_{new}.

Spy-EM-2(P, S, N, U_{new}):
Datos: P, S, N, U_{new}
$P = P \cup S$
$\forall d_i \in P : P(C = c_1|d_i) = 1$ (la cual no cambiará en ninguna iteración del EM)
$\forall d_j \in N : P(C = c_2|d_j) = 1$ (la cual cambiará en cada iteración del EM)
$\forall d_k \in U_{new}$: No se le asigna una etiqueta inicial.
$EM(P,N,U_{new})$

Algoritmo 3: Paso 2: Identificando posibles casos negativos

Una vez converga el algoritmo obtenemos un clasificador con el que clasificaremos los conjuntos N y U_{new}.

5. CAPÍTULO

Evaluación

Antes de la evaluación vamos a presentar un poco como son los datos. Disponemos de 3 datasets: el de Airbnb, el de Wimdu y el de Homeaway. Todos ellos están compuestos por los datos extraídos gracias al *web scraper* durante 30 días. Estos datos se limpian y se agrupan por identificador. Ahora tenemos por cada alojamiento una lista de 30 o menos precios (si se ha eliminado el alojamiento de la web).

Podemos ver en la Figura 5.1 la media de los precios por localidad.

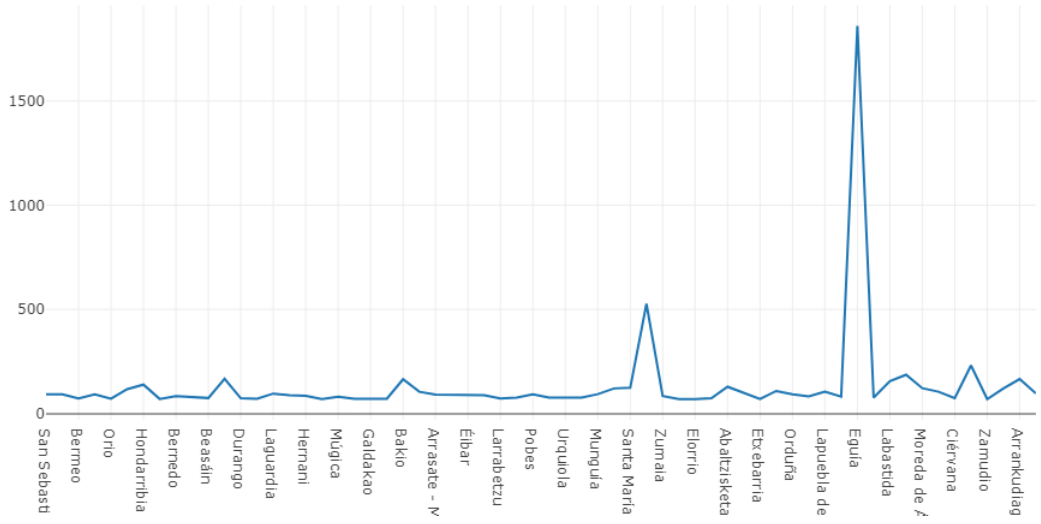

Figura 5.1: Precios medios por localidad

5.1. Evaluación del macheo

Para poder realizar la evaluación del macheo propuesto tomamos 200 alojamientos. Estos 200 alojamientos pertenecen a los datos escrapeados en un día en Wimdu. Realizamos un macheo a mano con los alojamientos de Homeaway. Para realizar este macheo manualmente consideramos que dos alojamientos son el mismo si los títulos y las descripciones se parecen y las imágenes coinciden. De esos 200 alojamientos 30 existen en la otra web y pudimos machearlos por lo que nuestra primera propuesta de machear las distintas webs quedó frustrada. Una vez tenemos un dataset de alojamientos macheados (o al menos los alojamientos existentes) utilizamos un 10 % de los datos como testeo y un 90 % de los datos para entrenar los modelos.

5.1.1. Evaluación experimental macheo

Utilizando distintas combinaciones de algoritmos y variables predictoras obtuvimos los resultados de la Tabla 5.1. La medida utilizada es el *accuracy* ya que lo que nos interesa es saber si un alojamiento se a macheado bien o no. Hay que tener en cuenta que la mayoría de los alojamientos no tienen su correspondiente en la otra web y estos se tienen que clasificar.

Variables	KNN	Radius KNN
Lat, Long	0.16	0.51
Lat, Long, Aforo, Dormitorios, Aseos	0.22	0.24
Dirección (Word Count)	0.12	0.51
Dirección (TFIDF)	0.13	0.51
Lat, Long, Aforo, Dormitorios, Aseos, Dirección (Word Count)	0.21	0.58
Lat, Long, Aforo, Dormitorios, Aseos, Dirección (TFIDF)	0.22	0.58
Lat, Long, Aforo, Dormitorios, Aseos, Dirección (TFIDF) con PCA	0.22	0.58

Tabla 5.1: Evaluación del macheo propuesto

No se obtienen resultados satisfactorios y esto se debe a varios factores. El primero de ellos es que como hemos dicho anteriormente, las webs tienen distintos alojamientos y no existen muchos alojamientos que coexistan en las distitnas webs. La mayoría de los alojamientos que coexisten en varias webs son los publicados por agencias turísticas o inmobiliarias. El segundo y el mayor de los problemas es que la mayoría de alojamientos se parecen en la mayoría de sus variables predictoras como el aforo, el número de dormito-

rios, el número de aseos..., incluso en la localización, aun siendo diferentes alojamientos. Además muchas de las localizaciones son orientativas por lo que se acaban agrupando en un mismo punto.

En definitiva, el macheo propuesto de las distintos alojamientos en distintas webs debido a las características de los datos, no es una ayuda para predecir el fraude fiscal; generaría en muchos de los casos macheos erróneos que van a aumentar el error en el modelo de predicción. Por ello, el macheo queda excluido de la detección de fraude.

5.2. Evaluación de las técnicas de detección de fraude

La evaluación de clasificadores en problemas PU es difícil debido a la ausencia de ejemplos negativos. La mayoría de las métricas como la precisión y el recall se estiman a partir de la matriz de confusión que se compone de 4 valores: true positive o verdaderos positivos (TP, número de ejemplos positivos correctamente clasificados), true negative o verdaderos negativos (TN, número de ejemplos negativos correctamente clasificados), false positive o falsos positivos (FP, número de ejemplos negativos clasificados como positivos) y false negative o falsos negativos (FN, número de ejemplos positivos clasificados como negativos).

En los problemas de PU no disponemos de ejemplos negativos por lo que no es posible estimar TN ni FP a partir de los datos. Por ello, tampoco se pueden calcular la mayoría de las métricas de evaluación más comunes. Un buen evaluador para nuestro modelo sería el valor F el cual está basado en la precisión y el recall. Para ello vamos a utilizar el *Pseudo-F* que proponen [Calvo et al., 2012] el cual es una estimación del valor F. El valor F es definido como la media armónica ponderada de la precision y del recall y su expresión general es:

$$\frac{1}{F_\alpha(r,p_r)} = \frac{1}{\alpha+1}\left(\frac{\alpha}{r}+\frac{1}{p_r}\right)$$

donde α es un factor de ponderación que nos permite enfatizar en la cantidad (el recall) o en la calidad (la precisión) de la recuperación. Cuando el valor de ponderación α es 1 (i.e. cuando consideramos el recall y la precisión igual de importantes), tenemos la definición más común de la medida F (conocida como F_1:

$$\frac{1}{F_1(r,p_r)} = \frac{1}{2}\left(\frac{1}{r}+\frac{1}{p_r}\right) \tag{5.1}$$

$$F_1(r, p_r) = \frac{2rp_r}{r + p_r} = F \tag{5.2}$$

De ahora en adelante cada vez que nos refiramos a F haremos referencia a la medida $F_1(r, p_r)$.

Dado un modelo ψ su recall es la probabilidad de que un ejemplo positivo sea clasificado como positivo por ψ y su precision, en cambio, es la probabilidad de que una instancia que se ha clasificado como positiva por ψ sea realmente positiva:

$$r = P(\psi(\boldsymbol{X}) = 1 | C = 1)$$

$$p_r = P(C = 1 | \psi(\boldsymbol{X}) = 1)$$

donde $\psi(\boldsymbol{X})$ representa la clase predicha por el clasificador ψ y C representa la clase real. Si tenemos en cuenta la regla de Bayes, tenemos que:

$$p_r = P(C = 1 | \psi(\boldsymbol{X}) = 1) = \frac{P(\psi(\boldsymbol{X}) = 1 | C = 1)p}{P(\psi(\boldsymbol{X}) = 1)} = \frac{rp}{P(\psi(\boldsymbol{X}) = 1)} \tag{5.3}$$

Mezclando las ecuaciones 5.2 y 5.3, obtenemos:

$$F = \frac{2rp}{P(\psi(\boldsymbol{X}) = 1) + p} \tag{5.4}$$

donde $P(\psi(\boldsymbol{X}) = 1)$ es la probabilidad de que el clasificador ψ clasifique la instancia como positiva (es decir, la probabilidad a priori de la clase positiva en la distribución de probabilidad definida por ψ). Como disponemos de la función de clasificación ψ, esta probabilidad puede ser estimada clasificando todas las posibles instancias y luego obteniendo el ratio de instancias clasificadas como positivas.

El recall puede ser estimado a partir de solo ejemplos positivos. Para ello, se ha usado un K-Fold Cross Calidation o una validación cruzada de repetidas k-hojas (K-Fold CV) [Rodriguez et al., 2010]. En cada repetición, los ejemplos positivos son divididos en k hojas y todas las hojas menos una se usan, junto con el conjunto de instancias no etiquetadas, para construir el clasificador PU. Este clasificador es usado para predecir la clase de las instancias en el pliegue que se ha dejado a parte y el recall es estimado como el ratio de instancias clasificadas como positivas. Todo el proceso se repite ρ veces sobre distintas particiones de los datos generados aleatoriamente y la estimación final del recall es calculado como la media de ρk estimaciones individuales.

Dada la probabilidad a priori de la clase positiva p_t[1] y un conjunto de ejemplos positivos y sin etiquetar, podemos obtener un modelo ψ_{p_t} con el que poder estimar el recall r_{p_t} y $P(\psi_{p_t}(\boldsymbol{X}) = 1)$. Por lo que el valor F puede obtenerse como:

$$F(p_t) = \frac{2r_{p_t}p}{P(\psi_{p_t}(\boldsymbol{X}) = 1) + p} \tag{5.5}$$

Para poder estimar $F(p_t)$, es necesario estimar r_{p_t}, $P(\psi_{p_t}(\boldsymbol{X}) = 1)$ y p. Como no tenemos estimador del último, se puede reemplazar por p_t. Esta estimación nos podría servir para evaluar el modelo pero, dado su alta dependencia del parámetro p_t, a un p_t mayor en el entrenamiento, mayor sería el valor de F, llevando a seleccionar el valor más alto de p_t. Para evitar este comportamiento [Calvo et al., 2012] proponen una nueva métrica llamada *Pseudo-F* (F_{ps}), definida como:

$$F_{ps}(p_t) = \frac{F(p_t)}{2p} = \frac{r_{p_t}}{P(\psi_{p_t}(\boldsymbol{X}) = 1) + p} \tag{5.6}$$

Como p es constante, $F_{ps} \propto F(p_t)$ y, por ello, $argmax_{p_t}\{F_{ps}(p_t)\} = argmax_{p_t}\{F(p_t)\}$. Es decir, F_{ps} y $F(p_t)$ son proporcionales por lo que sera la misma p_t la cual maximice ambas funciones.

De esta manera, dado un modelo ψ_{p_t}, el *Pseudo-F* es estimado como:

$$F_{ps}(p_t) = \frac{\hat{r}_{p_t}}{\hat{P}(\psi_{p_t}(\boldsymbol{X}) = 1) + p_t} \tag{5.7}$$

donde $\hat{r}_{p_t}$ es la estimación del recall y $\hat{P}(\psi_{p_t}(\boldsymbol{X}) = 1)$ es la estimación de la probabilidad de que el clasificador clasifique una instancia como positiva.

Ahora disponemos de una forma de evaluar el modelo. Aun así, ya que nuestro modelo depende de un parámetro de entrada el cual desconocemos también, vamos a utilizar esta métrica para calcular el valor de este parámetro.

La estructura general del K-Fold CV en escenarios PU es la que representa la Figura 5.2.

En este estudio como se ha explicado antes se utilizará el *Pseudo-F* para evaluar el modelo. Para calcular el *Pseudo-F* hace falta realizar una K-Fold CV para estimar el recall. Por lo tanto la evaluación junto con el tuneo del parámetro queda descrito en la Figura 5.3.

[1]Esta es la probabilidad a priori que hay que pasarle al modelo EM como hemos mencionado con antelación.

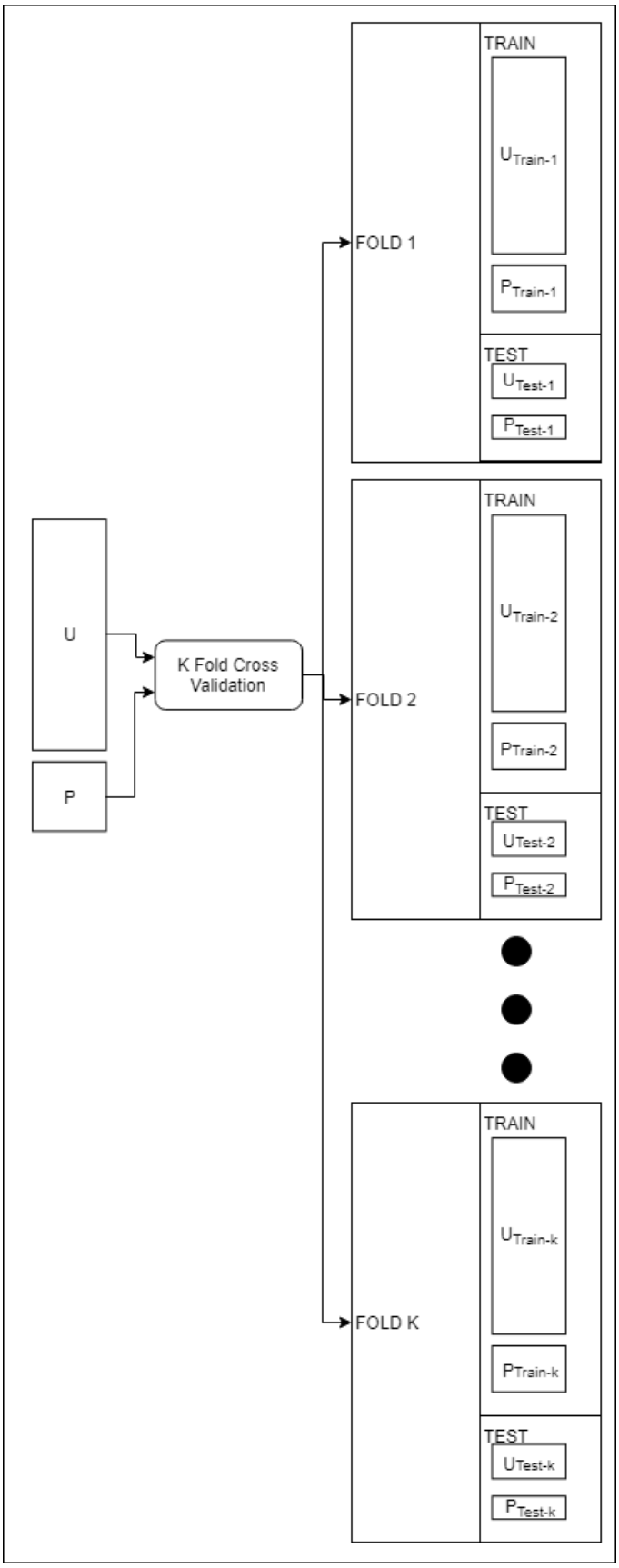

Figura 5.2: Estructura general del k-Fold Cross Validation en escenarios PU

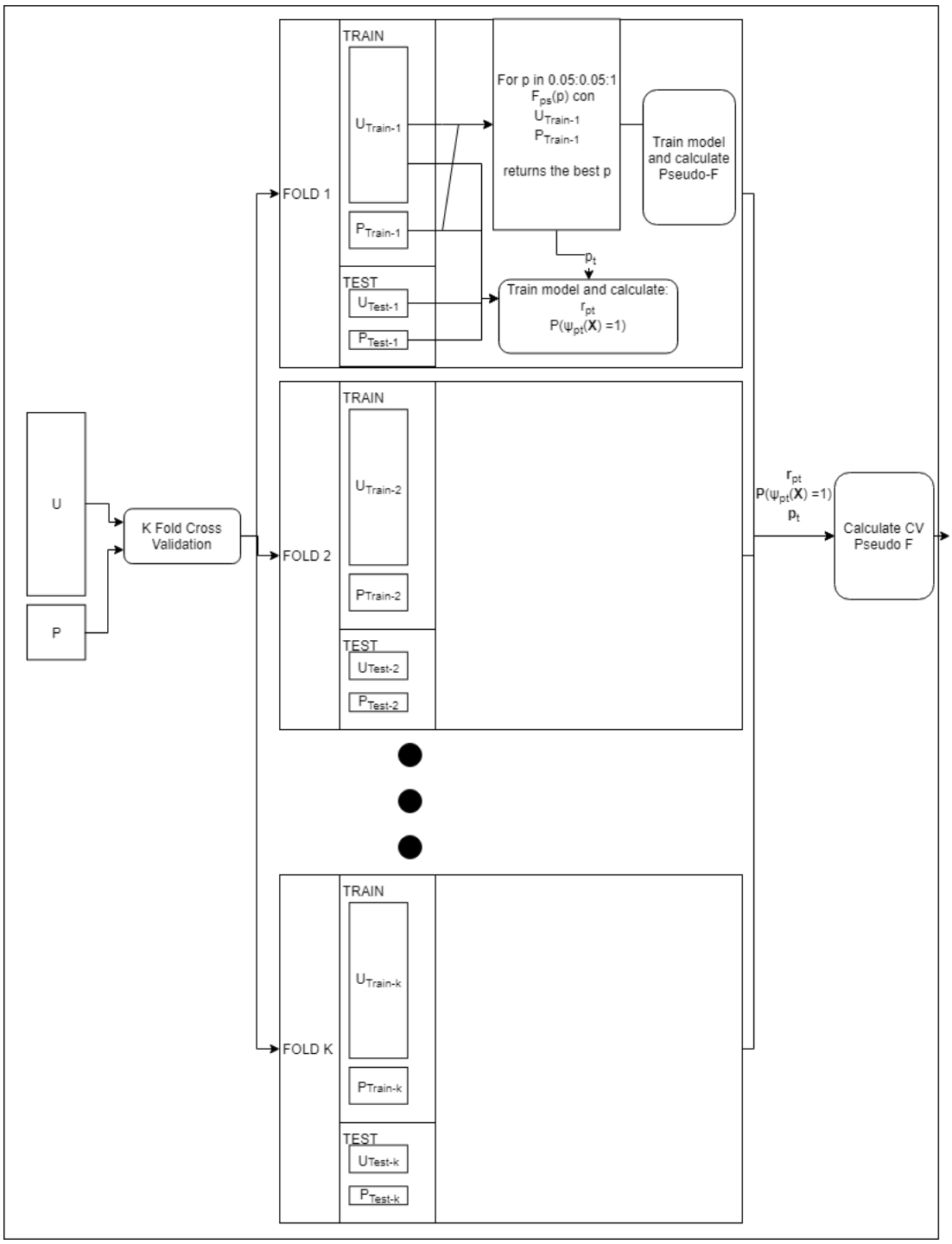

Figura 5.3: k-Fold Cross Validation *Pseudo-F* en escenarios PU

Para tunear el parámetro p (probabilidad a priori) probaremos a calcular el *Pseudo-F* con valores entre 0.05 y 1 aumentando en cada tramo en 0.05. De esta forma se elige el p con el que mayor *Pseudo-F* se consigue, el cual denotaremos como p_t.

Para calcular el *Pseudo-F* general hay que hallar los valores $\hat{r}_{p_t}$ y $\hat{P}(\psi_{p_t}(\boldsymbol{X}) = 1)$ para cada uno de las particiones. Obtendremos por cada partición unas estimadas $\hat{r}_{p_t}$, un $\hat{P}(\psi_{p_t}(\boldsymbol{X}) = 1)$ y un p_t. Vamos a denotar a cada una de las listas de valores obtenidos $\hat{\boldsymbol{r}}_{p_t}$, $\hat{\boldsymbol{P}}(\psi_{p_t}(\boldsymbol{X}) = 1)$, y $\boldsymbol{p}_t$, respectivamente. A partir de éstos podemos obtener dos valores: *micro Pseudo-F* y *macro Pseudo-F*.

Siendo $mean(\boldsymbol{x})$ la función que devuelve la media de los valores de $\boldsymbol{x}$.

Micro Pseudo-F:

$$MicroF_{ps}(p_t) = \frac{mean(\hat{\boldsymbol{r}}_{p_t})}{mean(\hat{\boldsymbol{P}}(\psi_{p_t}(\boldsymbol{X}) = 1)) + mean(\boldsymbol{p}_t)} \tag{5.8}$$

Macro Pseudo-F:

$$MacroF_{ps}(p_t) = mean\left(\frac{\hat{\boldsymbol{r}}_{p_t}}{\hat{\boldsymbol{P}}(\psi_{p_t}(\boldsymbol{X}) = 1) + \boldsymbol{p}_t}\right) \tag{5.9}$$

Una de las diferencias entre medias macro y micro es que la media macro aporta igual peso a cada una de las clases mientras que la media micro da igual peso a cada uno de los elementos a clasificar. Por eso, en las medias micro las clases grandes van a dominar a las clases pequeñas. Por ejemplo, si la clase c_1 tiene 10 ejemplos y la clase c_2 tiene 90 elementos y la precisión del modelo que vamos a evaluar es 0.6 y 0.9 respectivamente para cada clase la media micro va a ser superior que la macro ya que se va a acercar más a la precisión de la clase c_2 debido a que es más numerosa. [Van Asch, 2013]

5.2.1. Evaluación experimental de las técnicas de detección de fraude

Se han realizado experimentos sobre los tres datasets que tenemos: Airbnb, Homeaway y Wimdu. Para extender la comparativa, se ha probado con 4 datasets extras (CRX, Diabetes, Titanic WheatherAUS). Estos datasets originalmente no eran problemas de PU pero hemos eliminado etiquetas para probarlos. Para eliminar estas etiquetas hemos elegido aleatoriamente un 10 % de las etiquetas positivas y al resto les hemos quitado la etiqueta. Para los datasets de detección de fraude se han utilizado como variables predictoras la

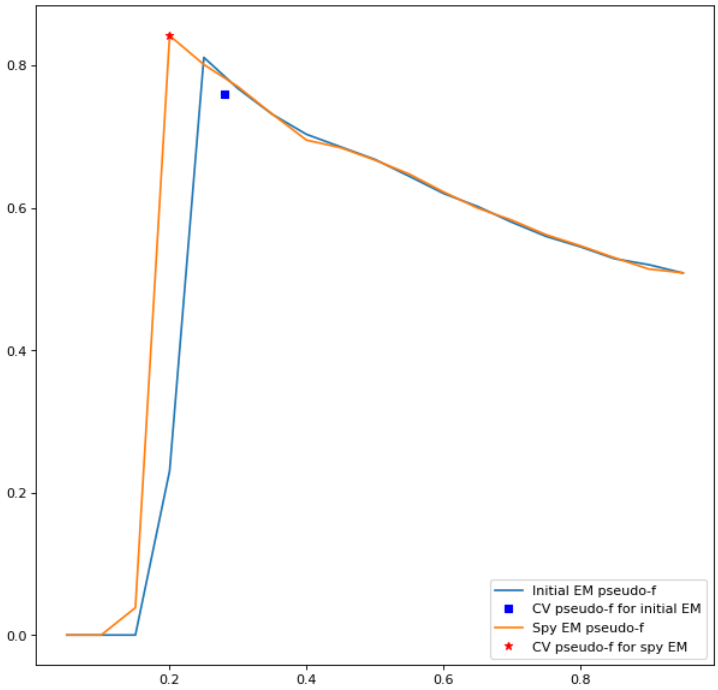

Figura 5.4: Tuneo parámetro p_t Airbnb

media, el valor mínimo, valor máximo, varianza, moda y mediana de los precios de cada alojamiento en un mes.

Los resultados obtenidos al tunear el parámetro p_t con los algoritmos Initial EM y Spy EM para cada una de las webs están representados en las Figuras 5.4, 5.5 y 5.6.

El eje x representa la variable p_t y en el eje y está representado el *Pseudo-F* obtenido para ese p_t. Como se puede observar el Spy-EM obtiene por lo general el mejor resultado con valores de p_t algo más bajos que el I-EM. Además,este valor de Spy-EM, es en todos los casos algo mayor que el I-EM. Para valores de p_t superiores, ambos algoritmos dan resultados casi idénticos y nunca por debajo del 0.5.

En las Tablas 5.2 y 5.3 se puede observar los resultados obtenidos para los algoritmos Initial EM y el Spy EM.

No existe una notable diferencia entre el Initial EM y el Spy EM pero sí que se obtienen mejores resultados utilizando el Spy EM.

Si se calcula el porcentaje de alojamientos que se clasifican como positivos (no fraudulentos) utilizando el algoritmo S-EM, para las tres páginas web ronda el 80 % (Wimdu 79 %, Airbnb 80 % y Homeaway 81 %). Esto denota que nuestro algoritmo clasificaría la gran mayoría de los casos como positivos. Ordenando los alojamientos por la probabilidad de

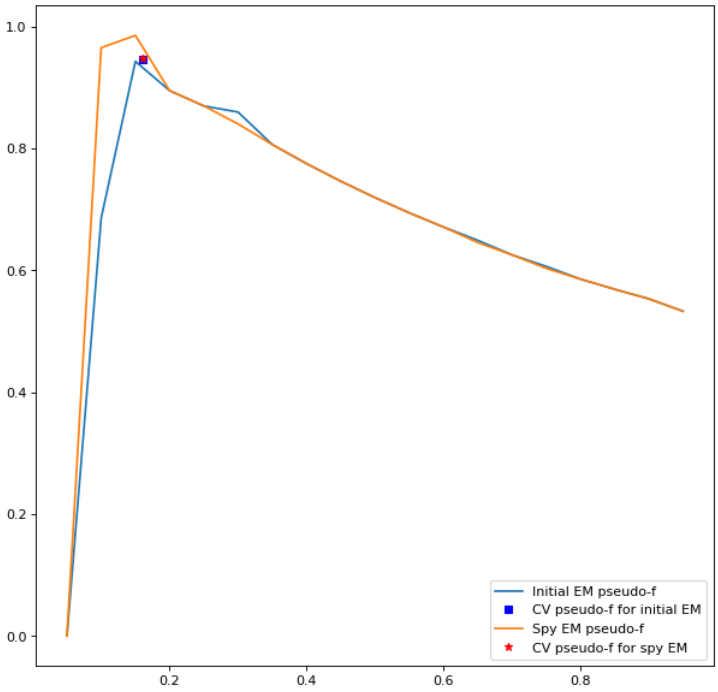

Figura 5.5: Tuneo parámetro p_t Wimdu

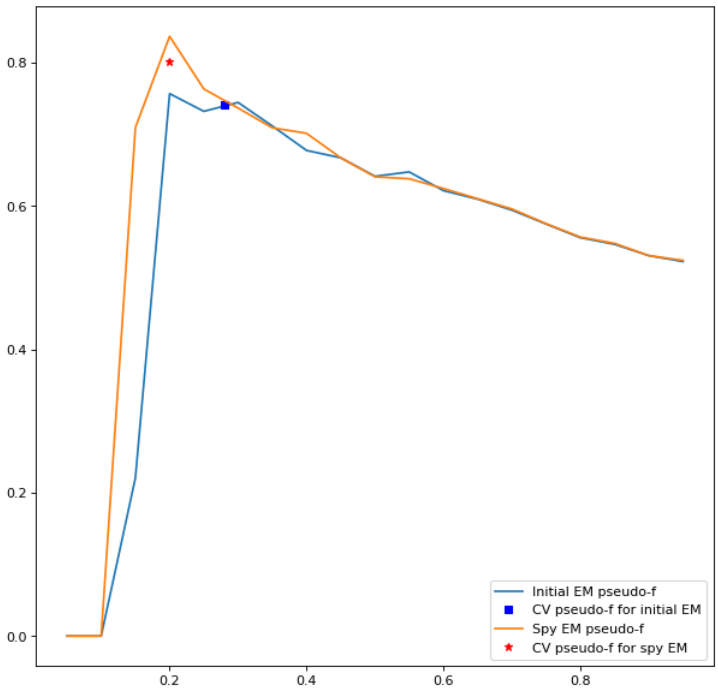

Figura 5.6: Tuneo parámetro p_t Homeaway

DATASET	MICRO F_{ps}	MACRO F_{ps}	MEAN($\boldsymbol{p}_t$)
CRX	0.61	0.61	0.28
Diabetes	0.68	0.56	0.34
Titanic	0.60	0.60	0.43
WeatherAUS	2.18	2.18	0.05
Airbnb	0.74	0.74	0.33
Wimdu	0.87	0.87	0.22
Homeaway	0.74	0.71	0.28

Tabla 5.2: Evaluación del algoritmo Initial EM

DATASET	MICRO F_{ps}	MACRO F_{ps}	MEAN($\boldsymbol{p}_t$)
CRX	0.61	0.61	0.28
Diabetes	0.68	0.56	0.34
Titanic	0.70	0.72	0.40
WeatherAUS	2.22	2.21	0.05
Airbnb	0.74	0.74	0.33
Wimdu	0.95	0.95	0.16
Homeaway	0.8	0.79	0.2

Tabla 5.3: Evaluación del algoritmo Spy EM

pertenecer a la clase positiva, permite dar una solución significativa. De esta forma nos permite dar una pista sobre si tiende más a ser un piso fraudulento o uno no fraudulento. Esta ordenación entendemos que sería de una notable utilidad en una aplicación real de estudio y chequeo de los alojamientos turísticos por parte de las entidades competentes.

Si nos fijamos en la correlación que tiene cada variable predictiva con la clase, por ejemplo en el caso del dataset de Airbnb (Figura 5.7), podemos comprobar que los pisos fraudulentos tienen a tener precios mayores. El orden de las variables predictoras es: media, mínimo, máximo, moda, mediana y etiqueta de legalidad. Se ha descartado la varianza en esta gráfica ya que no aportaba nada significativo.

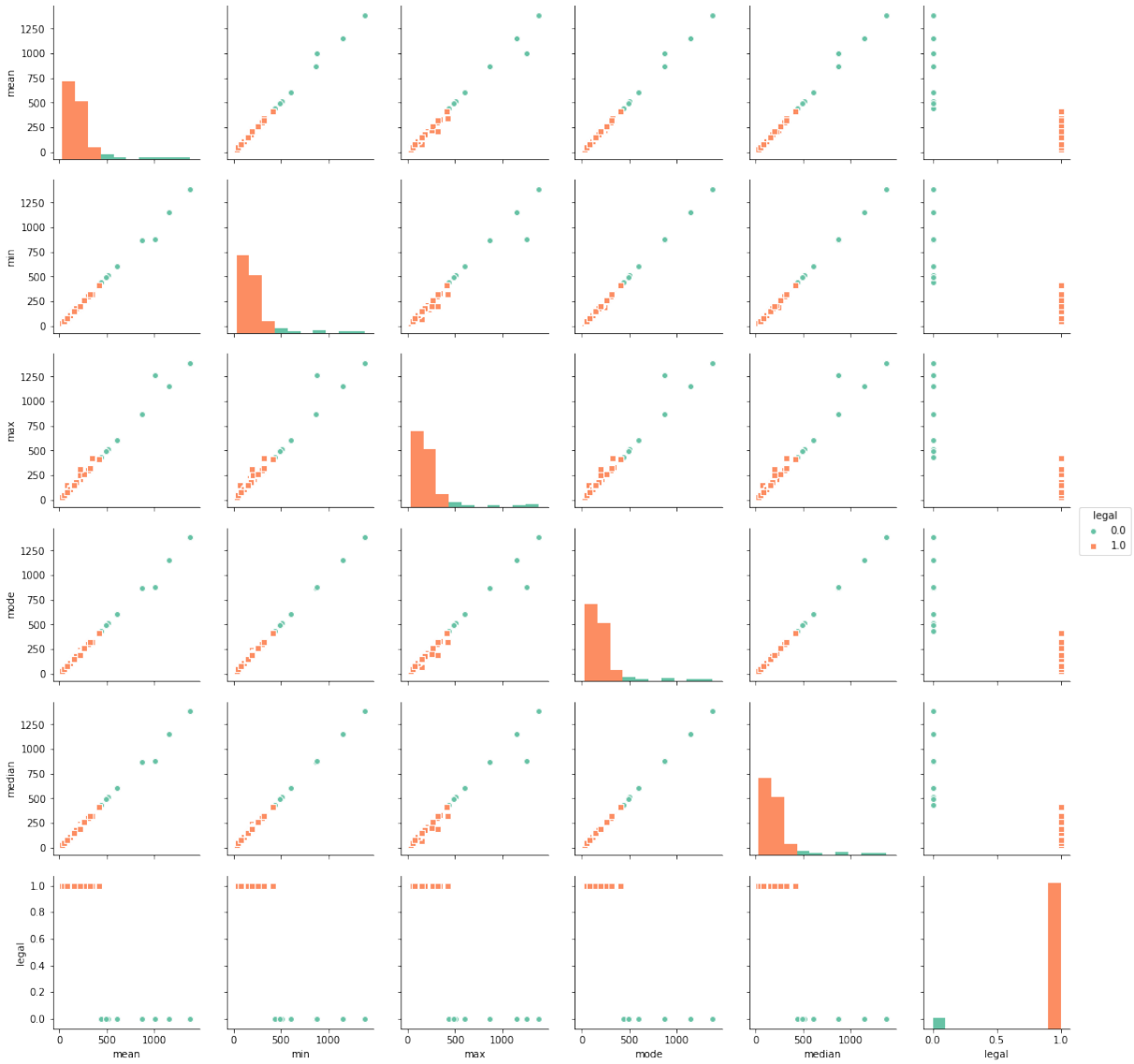

Figura 5.7: Matriz de correlación para Airbnb

6. CAPÍTULO

Conclusiones

En esta tesis hemos analizado un caso concreto, dentro de un proyecto real, cuyos datos tienen características particulares y hemos adaptado algoritmos de PU para poder dar una solución. La solución a este problema no es la óptima ya que es un caso muy complicado que conllevaría mucho más estudio pero es lo suficientemente robusta y útil como para poder dar una pista sobre que pisos pueden ser fraudulentos. La solución que hemos dado puede servir para obtener una lista de mayor probabilidad de ser fraudulentos a menor. Esto podría ayudar a la institución pública correspondiente para tener una lista con prioridad de análisis para así minimizar gastos y aumentar la probabilidad de detectar el fraude.

Gracias a este estudio podemos deducir que los pisos no fraudulentos tienden a tener precios más bajos. Esto se puede tener en consideración para futuros estudios sobre este ámbito. A priori teníamos la hipótesis de que los pisos fraudulentos tenderían a precios más bajos para no llamar la atención pero parece ser que ocurre lo contrario.

En cuanto a la evaluación del método propuesto, el uso de la medida Pseudo F es necesaria debido a la naturaleza de los datos. Esta no es una medida exacta ya que es una estimación del valor F pero nos sirve además de para tunear los parámetros, comparar ambos métodos que hemos utilizado.

En cuanto al trabajo futuro, sería interesante seguir por la línea del macheo de alojamientos. Comprobar si un alojamiento está en una única página web o en varias puede aportar información al modelo.

Otro acercamiento si se dispone de una cantidad de datos en un intervalo de tiempo mayor (1-2 años de datos) es utilizar series temporales para modelar el comportamiento de los

pisos fraudulentos. Utilizar algoritmos como el LCLC (Learning from Common Local Clusters) de [Nguyen et al., 2011] es una vía a explorar.

Por otro lado, lo más interesante sería que existiera un feedback por parte de las entidades públicas para mejorar el modelo. Si consiguieran descubrir si realmente un alojamiento es fraudulento o no y lo etiquetamos como tal el algoritmo sera más y más preciso ya que tendríamos menos datos sin etiquetar y tendríamos muestras etiquetadas como negativas.

Anexos

A. ANEXO

Ejemplo de una respuesta de la API de HomeAway


```
{
   'nextPage':'https://ws.homeaway.com/public/search?q=Basque+Country&ageSize=30&availabilityStart=2018-05
   -16&availabilityEnd=2018-05-19&page=2',
   'pageSize':30,
   'pageCount':10,
   'page':1,
   'size':275,
   'refinements':[
      {
         'key':'regions_title',
         'fieldName':'region',
         'options':[

         ]
      },
      {
         'key':'bathrooms_title',
         'fieldName':'Bathrooms',
         'options':[
            {
               'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
               tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bathrooms:1',
               'count':134,
               'title':'1+'
            },
            {
               'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
               tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bathrooms:2',
               'count':77,
               'title':'2+'
            },
            {
               'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
               tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bathrooms:3',
               'count':18,
               'title':'3+'
            },
            {
               'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
               tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bathrooms:4',
               'count':14,
               'title':'4+ bathrooms'
            }
```

```
    ]
},
{
    'key':'bedrooms_title',
    'fieldName':'Bedrooms',
    'options':[
        {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bedrooms:6',
            'count':7,
            'title':'6+'
        },
        {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bedrooms:1',
            'count':51,
            'title':'1'
        },
        {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bedrooms:2',
            'count':96,
            'title':'2'
        },
        {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bedrooms:3',
            'count':65,
            'title':'3'
        },
        {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bedrooms:4',
            'count':27,
            'title':'4'
        },
        {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bedrooms:5',
            'count':8,
            'title':'5'
        },
        {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Bedrooms:St
                udio',
            'count':21,
            'title':'Studio'
        }
    ]
},
{
    'key':'sleeps_title',
    'fieldName':'Sleeps',
    'options':[
        {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:1',
            'count':275,
            'title':'1+'
        },
        {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:10',
            'count':21,
            'title':'10+'
        },
        {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:11',
```

```
            'count':14,
            'title':'11+'
         },
         {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:12',
            'count':13,
            'title':'12+'
         },
         {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:13',
            'count':4,
            'title':'13 or more'
         },
         {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:2',
            'count':275,
            'title':'2+'
         },
         {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:3',
            'count':236,
            'title':'3+'
         },
         {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:4',
            'count':221,
            'title':'4+'
         },
         {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:5',
            'count':133,
            'title':'5+'
         },
         {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:6',
            'count':106,
            'title':'6+'
         },
         {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:7',
            'count':54,
            'title':'7+'
         },
         {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:8',
            'count':40,
            'title':'8+'
         },
         {
            'url':'https://ws.homeaway.com/public/search?q=Basque+Country&pageSize=30&page=1&availabili
            tyStart=2018-05-16&availabilityEnd=2018-05-19&refine=Sleeps:9',
            'count':24,
            'title':'9+'
         }
      ]
   }
],
'entries':[
   {
      'listingId':'8352043',
      'listingSource':'homeaway_es',
```

```
   'headline':'Apartment Zurriola Loft by FeelFree Rentals',
   'description':'The Zurriola Loft apartment is located in a building facing the sea and it has bee
   n transformed into a modern accommodation: the open space reproduces the idea of\u200b\u200ba lof
   t New York, on a small scale.\n\nWith stunning views over Zurriola Beach, this ...',
   'accommodations':'2 BR, 2.0BA, Sleeps 4',
   'minStayRange':{
      'minStayHigh':3,
      'minStayLow':3
   },
   'thumbnail':{
      'height':100,
      'imageSize':'SMALL',
      'secureUri':'https://imagesus-ssl.homeaway.com/mda01/14d774c9-176e-4865-8b20-f94ce96f98c0.1.1',
      'uri':'http://imagesus.homeaway.com/mda01/14d774c9-176e-4865-8b20-f94ce96f98c0.1.1',
      'width':134
   },
   'priceQuote':{
      'amount':772.63,
      'averageNightly':257.54,
      'currencyUnits':'USD',
      'fees':None,
      'other':None,
      'rent':772.63,
      'tax':None
   },
   'priceRanges':[
      {
         'currencyUnits':'EUR',
         'from':199.0,
         'periodType':'NIGHTLY-WEEKDAY',
         'to':199.0
      },
      {
         'currencyUnits':'EUR',
         'from':249.0,
         'periodType':'NIGHTLY-WEEKEND',
         'to':249.0
      }
   ],
   'location':{
      'lat':43.3250674,
      'lng':-1.9753214,
      'city':'San Sebastián',
      'state':'Gipuzkoa',
      'country':'ES'
   },
   'regionPath':'World Apartment #8352043',
   'reviewCount':6,
   'reviewAverage':5.0,
   'bookWithConfidence':True,
   'detailsUrl':'https://ws.homeaway.com/public/listing?id=83520
      43',
   'bathrooms':2.0,
   'bedrooms':2,
   'listingUrl':'https://www.homeaway.es/p8352043?uni_id=4409208'
},
{
   'listingId':'6906320',
   'listingSource':'homeaway_es',
   'headline':'Apartment Kantauri by FeelFree Rentals',
   'description':'If for your coming holidays in San Sebastian you"re looking for an exclusive and
      spacious apartment at an unbeatable location, then Kantauri is the perfect accommodation for
      you.\n\nIt is situated just opposite Zurriola Beach and it offers a large ...',
   'accommodations':'2 BR, 1.0BA, Sleeps 5',
   'minStayRange':{
      'minStayHigh':3,
      'minStayLow':3
   },
   'thumbnail':{
      'height':100,
```

```
      'imageSize':'SMALL',
      'secureUri':'https://imagesus-ssl.homeaway.com/mda01/4f70a
         6fc-c1cf-47aa-812f-9d290b7b2952.1.1',
      'uri':'http://imagesus.homeaway.com/mda01/4f70a6fc-c1cf-47
         aa-812f-9d290b7b2952.1.1',
      'width':134
   },
   'priceQuote':{
      'amount':772.63,
      'averageNightly':257.54,
      'currencyUnits':'USD',
      'fees':None,
      'other':None,
      'rent':772.63,
      'tax':None
   },
   'priceRanges':[
      {
         'currencyUnits':'EUR',
         'from':199.0,
         'periodType':'NIGHTLY-WEEKDAY',
         'to':199.0
      },
      {
         'currencyUnits':'EUR',
         'from':249.0,
         'periodType':'NIGHTLY-WEEKEND',
         'to':249.0
      }
   ],
   'location':{
      'lat':43.3249065,
      'lng':-1.975753,
      'city':'San Sebastián',
      'state':'Gupúzcoa',
      'country':'ES'
   },
   'regionPath':'World Apartment #6906320',
   'reviewCount':11,
   'reviewAverage':4.7272725,
   'bookWithConfidence':True,
   'detailsUrl':'https://ws.homeaway.com/public/listing?id=69063
      20',
   'bathrooms':1.0,
   'bedrooms':2,
   'listingUrl':'https://www.homeaway.es/p6906320?uni_id=4085262'
},
{
   'listingId':'8610180',
   'listingSource':'homelidays_es',
   'headline':' Next to the Zurriola Beach and the Kursaal. Optional parking. ESS00856',
   'description':'Renovated apartment in San Sebastian. Ideal for families and congressmen. Located
      next to the PALACIO DE CONGRESOS DEL KURSAAL and PLAYA DE GROS, ZURRIOLA.  Close to the OLD
      PART and the CITY CENTER (5 minutes on foot) allows immediate access on f...',
   'accommodations':'3 BR, 1.0BA, Sleeps 4',
   'minStayRange':{
      'minStayHigh':2,
      'minStayLow':2
   },
   'thumbnail':{
      'height':100,
      'imageSize':'SMALL',
      'secureUri':'https://imagesus-ssl.homeaway.com/mda01/aa2d279d-54e6-4576-ad39-e5cb4ab6d9e2.1.1',
      'uri':'http://imagesus.homeaway.com/mda01/aa2d279d-54e6-4576-ad39-e5cb4ab6d9e2.1.1',
      'width':133
   },
   'priceQuote':{
      'amount':417.96,
      'averageNightly':139.32,
      'currencyUnits':'USD',
```

```
      'fees':None,
      'other':None,
      'rent':417.96,
      'tax':None
   },
   'priceRanges':[
      {
         'currencyUnits':'EUR',
         'from':100.0,
         'periodType':'NIGHTLY-WEEKDAY',
         'to':100.0
      }
   ],
   'location':{
      'lat':43.3237275,
      'lng':-1.9771984,
      'city':'Donostia',
      'state':'PV',
      'country':'ES'
   },
   'regionPath':'World Apartment #8610180',
   'reviewCount':3,
   'reviewAverage':5.0,
   'bookWithConfidence':True,
   'detailsUrl':'https://ws.homeaway.com/public/listing?id=8610180',
   'bathrooms':1.0,
   'bedrooms':3,
   'listingUrl':'https://www.homelidays.es/alojamiento/p8610180?uni_id=4671142'
},
{
   'listingId':'1891716',
   'listingSource':'homelidays_es',
   'headline':'La Concha Bay Suite & Views',
   'description':'Located in beachfront of La Concha Bay, This elegant and unique suite is the
      perfect location for your stay in San Sebastian. \nRight close to Hotel de Londres, bars,
      restaurants, boutiques, etc. are 2 minutes away.  La Concha Beach is just in fron...',
   'accommodations':'1 BR, 1.0BA, Sleeps 2',
   'minStayRange':{
      'minStayHigh':2,
      'minStayLow':2
   },
   'thumbnail':{
      'height':100,
      'imageSize':'SMALL',
      'secureUri':'https://imagesus-ssl.homeaway.com/mda01/81d3d6de-b6f1-408b-908c-fade969d4dfa.1.1',
      'uri':'http://imagesus.homeaway.com/mda01/81d3d6de-b6f1-408b-908c-fade969d4dfa.1.1',
      'width':133
   },
   'priceQuote':{
      'amount':1003.1,
      'averageNightly':334.37,
      'currencyUnits':'USD',
      'fees':None,
      'other':None,
      'rent':1003.1,
      'tax':None
   },
   'priceRanges':[
      {
         'currencyUnits':'EUR',
         'from':280.0,
         'periodType':'NIGHTLY-WEEKDAY',
         'to':280.0
      }
   ],
   'location':{
      'lat':43.3176886,
      'lng':-1.9851683,
      'city':'Donostia',
      'state':'PV',
```

```
            'country':'ES'
         },
         'regionPath':'World Apartment #1891716',
         'reviewCount':6,
         'reviewAverage':5.0,
         'bookWithConfidence':True,
         'detailsUrl':'https://ws.homeaway.com/public/listing?id=1891716',
         'bathrooms':1.0,
         'bedrooms':1,
         'listingUrl':'https://www.homelidays.es/alojamiento/p1891716?uni_id=3604853'
      }
   ]
}
```

Bibliografía

[Batty, 2013] Batty, M. (2013). Big data, smart cities and city planning. *Dialogues in Human Geography*, 3(3):274–279.

[Bentley et al., 1977] Bentley, J. L., Stanat, D. F., and Williams, E. (1977). The complexity of finding fixed-radius near neighbors. *Information Processing Letters*, 6(6):209 – 212.

[Bhatia et al., 2010] Bhatia, N. et al. (2010). Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*.

[Calvo, 2008] Calvo, B. (2008). Positive unlabelled learning with applications in computational biology. *Department of Computer Science and ArtiEcial Intelligence. San Sebastian, Spain: University of the Basque Country*.

[Calvo et al., 2012] Calvo, B., Inza, I., Larrañaga, P., Lozano, J. A., Calvo, B., Inza, I., Lozano, J. A., and Larrañaga, P. (2012). Wrapper positive Bayesian network classifiers. *Knowl Inf Syst*, 33:631–654.

[Calvo et al., 2007a] Calvo, B., Larrañaga, P., and Lozano, J. A. (2007a). Learning bayesian classifiers from positive and unlabeled examples. *Pattern Recognition Letters*, 28(16):2375–2384.

[Calvo et al., 2007b] Calvo, B., López-Bigas, N., Furney, S. J., Larrañaga, P., and Lozano, J. A. (2007b). A partially supervised classification approach to dominant and recessive human disease gene prediction. *Computer methods and programs in biomedicine*, 85(3):229–237.

[Chen et al., 2013] Chen, J., Chen, Y., Du, X., Li, C., Lu, J., Zhao, S., and Zhou, X. (2013). Big data challenge: a data management perspective. *Frontiers of Computer Science*, 7(2):157–164.

[Chen et al., 2014] Chen, M., Mao, S., and Liu, Y. (2014). Big data: A survey. *Mobile networks and applications*, 19(2):171–209.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

[Denis, 1998] Denis, F. (1998). Pac learning from positive statistical queries. In *International Conference on Algorithmic Learning Theory*, pages 112–126. Springer.

[Denis et al., 2002] Denis, F., Gilleron, R., and Tommasi, M. (2002). Text classification from positive and unlabeled examples. In *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'02*, pages 1927–1934.

[Du Plessis et al., 2015] Du Plessis, M., Niu, G., and Sugiyama, M. (2015). Convex formulation for learning from positive and unlabeled data. In *International Conference on Machine Learning*, pages 1386–1394.

[Du Plessis et al., 2014] Du Plessis, M. C., Niu, G., and Sugiyama, M. (2014). Analysis of learning from positive and unlabeled data. In *Advances in neural information processing systems*, pages 703–711.

[Elkan and Noto, 2008] Elkan, C. and Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220. ACM.

[Friedman et al., 1997] Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, 29(2-3):131–163.

[Hernández-González et al., 2016] Hernández-González, J., Inza, I., and Lozano, J. A. (2016). Weak supervision and other non-standard classification problems: A taxonomy. *Pattern Recognition Letters*, 69:49 – 55.

[Howe et al., 2008] Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Hill, D. P., Kania, R., Schaeffer, M., St Pierre, S., et al. (2008). Big data: The future of biocuration. *Nature*, 455(7209):47.

[Jain et al., 2016] Jain, S., White, M., and Radivojac, P. (2016). Estimating the class prior and posterior from noisy positives and unlabeled data. In *Advances in Neural Information Processing Systems*, pages 2693–2701.

[Joachims, 1996] Joachims, T. (1996). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science.

[John and Langley, 1995] John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc.

[Li and Liu, 2003] Li, X. and Liu, B. (2003). Learning to classify texts using positive and unlabeled data. In *IJCAI*, volume 3, pages 587–592.

[Liu et al., 2003] Liu, B., Dai, Y., Li, X., Lee, W. S., and Yu, P. S. (2003). Building text classifiers using positive and unlabeled examples. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 179–186. IEEE.

[Liu and Lee, 2003] Liu, B. and Lee, W. S. (2003). Learning with positive and unlabeled examples using weighted logistic regression. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. ICML.

[Liu et al., 2002] Liu, B., Lee, W. S., Yu, P. S., and Li, X. (2002). Partially supervised classification of text documents. In *ICML*, volume 2, pages 387–394. Citeseer.

[Minsky, 1961] Minsky, M. (1961). Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30.

[Nguyen et al., 2011] Nguyen, M. N., Li, X.-L., and Ng, S.-K. (2011). Positive unlabeled leaning for time series classification. In *IJCAI*, volume 11, pages 1421–1426.

[Pan et al., 2012] Pan, S., Zhang, Y., and Li, X. (2012). Dynamic classifier ensemble for positive unlabeled text stream classification. *Knowledge and information systems*, 33(2):267–287.

[Reynolds et al., 2000] Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1):19 – 41.

[Richards, 2006] Richards, J. A. (2006). *Supervised Classification Techniques*, pages 193–247. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Rodriguez et al., 2010] Rodriguez, J. D., Perez, A., and Lozano, J. A. (2010). Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):569–575.

[Sakai et al., 2018] Sakai, T., Niu, G., and Sugiyama, M. (2018). Semi-supervised auc optimization based on positive-unlabeled learning. *Machine Learning*, 107(4):767–794.

[Sakai et al., 2016] Sakai, T., Plessis, M. C. D., Niu, G., and Sugiyama, M. (2016). Semi-supervised classification based on classification from positive and unlabeled data. *arXiv preprint arXiv:1605.06955*.

[Sriphaew et al., 2009] Sriphaew, K., Takamura, H., and Okumura, M. (2009). Cool blog classification from positive and unlabeled examples. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 62–73. Springer.

[Valiant, 1984] Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

[Van Asch, 2013] Van Asch, V. (2013). Macro-and micro-averaged evaluation measures [[basic draft]].

[Yu et al., 2004] Yu, H., Han, J., and Chang, K.-C. (2004). Pebl: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):70–81.

[Zhang and Lee, 2005] Zhang, D. and Lee, W. S. (2005). A simple probabilistic approach to learning from positive and unlabeled examples. In *Proceedings of the 5th Annual UK Workshop on Computational Intelligence (UKCI)*, pages 83–87.